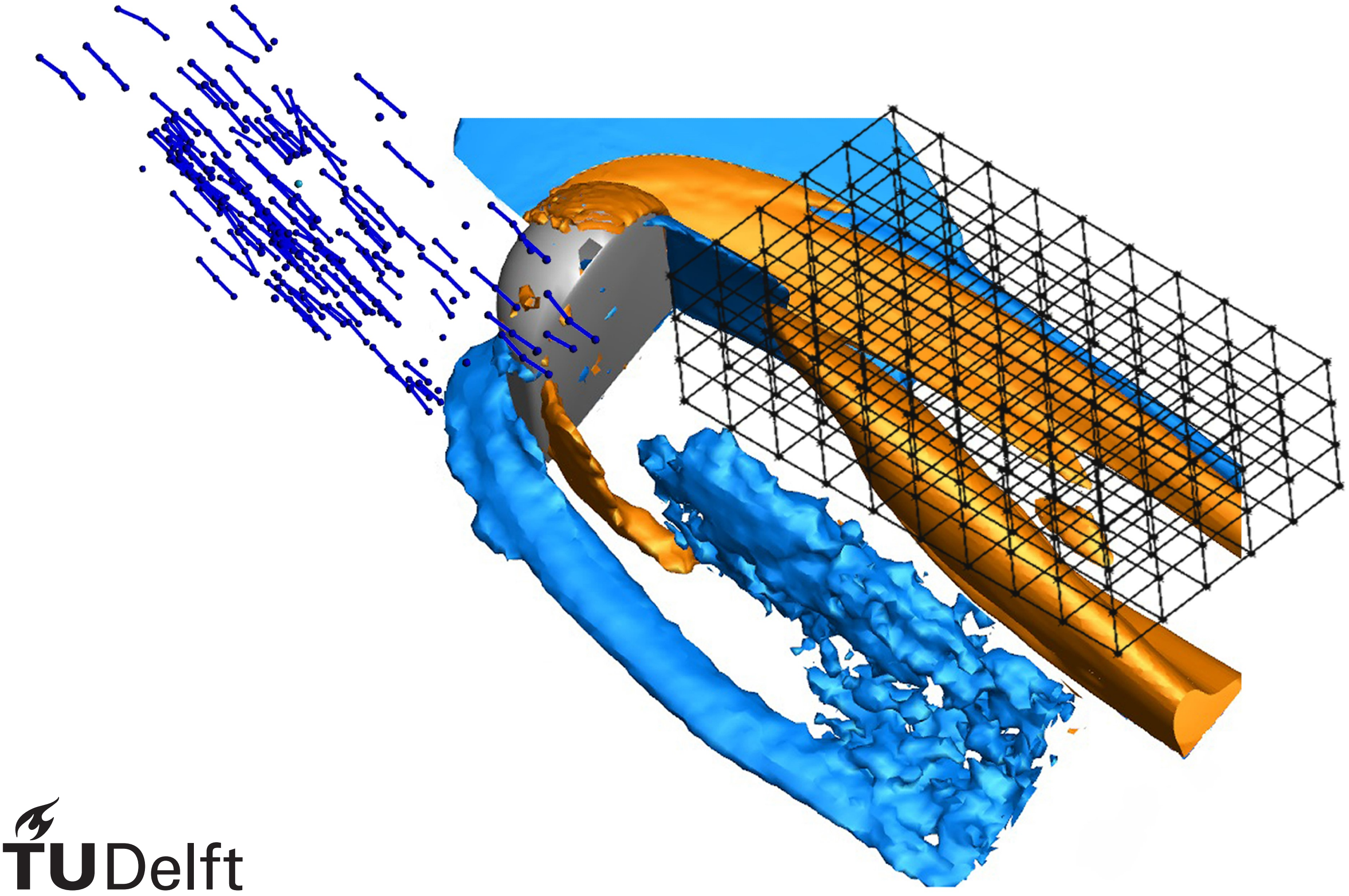
MSc Thesis

A state observer based data assimilation method between RANS and robotic PIV data

Nikhilesh Tumuluru Ramesh



TU Delft

MSc Thesis

A state observer based data assimilation method between RANS and robotic PIV data

by

Nikhilesh Tumuluru Ramesh

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday August 29, 2019 at 03:00 PM.

Student number:	4723775	
Project duration:	November 15, 2018 – July 31, 2019	
Thesis committee:	Prof. dr. ir. S. Hickel,	TU Delft, chair of committee
	Dr. A. Sciacchitano ,	TU Delft,supervisor
	Dr. D. Ragni ,	TU Delft, committee member
	Ir. E. Saredi,	TU Delft, Daily supervisor

This thesis is confidential and cannot be made public until December 31, 2021.

TUDelft

Acknowledgment

If there were one word to describe my feelings at the end of this memorable two year journey, it is *grateful*. Grateful for the opportunity to undertake this course, grateful for the relationships I have built, grateful for the timely support I received and grateful for the learnings I accumulated.

To my supervisor, Dr. Andrea Sciacchitano, I would like to express my sincere gratitude. It was a true privilege to learn from him. His constant insightful *feedback*, not unlike the crux of my thesis, provided me direction in times where I could not see the bigger picture. His attention to detail and intellect never failed to amaze me and are qualities that I will carry forward and strive to achieve.

This project would not have been possible without the support of my PhD supervisor, Edoardo Saredi. His resourcefulness,enthusiasm and willingness to help have benefited many and mostly me during the course of this project. Working together with him was a very enriching experience where I learned a lot. Thank you Edo!

At the beginning of my masters, I did not believe that it would be possible to have an active social circle due to the rigours of the course. Looking back, I can say with confidence that I would have struggled in my course without the love and support of my social circle. To the *aerodynamics haus*- Akshar, Ankith, Sidharth, Kushal, Abhinand, Nikhil, Athreya, Sampath, Jatinder, Chakshu, Shubham, Sanjeet and Maria, to *Wimbledon boys*-Karthik and Arya, to *Montgomerylaan*- Sahana, Pooja, Shrutika, Ashvij and Rajath, I will always cherish these two years. Thank you for the memories! I would be remiss not to thank all my fellow researchers in the basement who were always so cheerful and helpful. Thank you Tomas, Lucas, Jorge and Niels.

No two sentences can encapsulate the feelings I wish to express for my parents. Their unbounded love and affection has shaped me into the person I am today. I will never take for granted the sacrifices they made sending their only child to a foreign land to pursue a degree that revolves around the study of “air". To *Amma* and *Nana*, this thesis is dedicated to you.

Nikhilesh Tumuluru Ramesh
Delft, August 2019

Abstract

Experimental fluid dynamics and computational fluid dynamics have traditionally been treated as disparate fields of study. However, each field has its own unique set of advantages and disadvantages. Data assimilation is a field that can be used to leverage some of the advantages each field offers to help compensate mutual weaknesses. In this thesis, a state observer based data assimilation method is used to assimilate 3-D experimental data obtained in a wind tunnel experiment onto a steady RANS simulation. The experimental data is considered as the ground truth and is used to condition the RANS simulation. An understanding of the working of the method along with a study on the effect of different parameters of the state observer method are gathered by first applying it on the 1-D viscous Burgers equation and a 2-D CFD simulation. For the 3-D case, experimental data is obtained by performing a wind tunnel experiment using robotic PIV to map the time-averaged velocity field around a bluff body following which the data is assimilated onto a steady RANS simulation of the same body. Application of this method helps to recreate topological features and velocity fields of the flow with better accuracy than a baseline CFD simulation. Finally, the effects of the different parameters on the success of the method along with recommendations for improving the method are provided.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

CFD	Computational Fluid Dynamics
CVV	Coaxial Volumetric Velocimetry
DA	Data Assimilation
DNS	Direct Numerical Simulation
FOV	Field of View
HFSB	Helium Filled Soap Bubbles
IPR	Iterative Particle Reconstruction
LES	Large Eddy Simulation
MART	Multiplicative Algebraic Reconstruction Technique
NWP	Numerical Weather Prediction
OTF	Optical Transfer Function
PIV	Particle Image Velocimetry
PTV	Particle Tracking Velocimetry
RANS	Reynolds Averaged navier Stokes
STB	Shake The Box

Symbols

$\bar{p}$	Pressure	$\frac{kg}{ms^2}$
$\bar{u}$	Mean velocity component	$\frac{m}{s}$
δ_{99}	Boundary layer thickness	mm
$\dot{N}$	Number of bubbles per second	$bubbles/s$
$\dot{V}$	Volumetric flow rate	$\frac{m^3}{s}$
ϵ_u	Uncertainty in velocity	$\frac{m}{s}$
λ	Wavelength	nm
μ	Dynamic Viscosity	$\frac{kg}{ms}$
ν	Kinemaic viscosity	$\frac{m^2}{s}$
ω	Frequency	$\frac{1}{s}$
ρ	Density	$\frac{kg}{m^3}$
τ	Time scale	s
τ_p	Particle time response	s

τ_w	Wall shear stress	$\frac{kg}{ms^2}$
A	Area	m^2
C	Seeding Concentration	
D	Diameter	m
d	Diameter	m
E	Error term	
e	Error	
$EVol$	Feedback term	$\frac{m}{s^2}$
f	Body forcing term	$\frac{m}{s^2}$
f_A/f_B	Body forcing term	$\frac{kgm^2}{s^2}$
K	Feedback Gain	
k	Turbulent kinetic energy	$\frac{m^2}{s}$
l	Length scale	m
N	Percentage of control volumes	
p/P	Pressure	$\frac{kg}{ms^2}$
R	Reynolds stress tensor	$\frac{m^2}{s^2}$
r	weight function	
Re	Reynolds number	
t	Time	s
u/U	Velocity vector	$\frac{m}{s}$
u_τ	Skin friction velocity	$\frac{m}{s}$
X	X position	m
Y	Y position	m
y^+	Non dimensional wall unit	
Z	Z position	m
u'	Fluctuating velocity component	$\frac{m}{s}$

1
Introduction

Fluids are truly ubiquitous. From the air we breathe, the water we drink all the way up to the weather we experience, we are surrounded by fluids. There is no debating the fact that fluids play a vital role in sustaining life on earth and hence its study, fluid mechanics, is a discipline that has enamored centuries of scientific minds. From an engineering perspective however, particular attention is paid to the behaviour of fluids upon their interaction with solid objects. A brief run through the history of fluid mechanics (Calero [9]) reveal the plethora of methods and tools were used for its study. Today, the study of fluid mechanics can be classified into three broad categories- experimental, computational and analytical. While experimental and analytical methods have been the most popular, especially in the years past, rapid advances in technology has allowed computational methods to contribute significantly to the study.

Although Computational Fluid Dynamics [1] and experimental fluid dynamics adhere to the same goal, they have traditionally been treated as disparate fields. CFD, at its core is a set of approximations for the partial differential equations that describe fluid flow- the Navier Stokes equations, whereas experiments, when conducted with care can be trusted to replicate actual fluid behaviour. Each field, however, has its own unique set of advantages and disadvantages. Experimental techniques are not always feasible due to constraints in cost, equipment or facilities. The veracity of the measurements themselves can also be compromised to a certain extent by measurement noise, and the intrusiveness of the measurement techniques. Recent developments have led to the invention of techniques like Robotic Volumetric Particle Image Velocimetry (Jux [28]) [2], Helium Filled Soap Bubbles (Scarano et al.[41]) [3] as tracer particles for air flows and particle tracking algorithms like Shake the Box (Schanz et al.[43]), which make it possible to obtain large scale three dimensional velocity flow fields. However, the measurements are always corrupted by a certain level of noise, the most glaring of which includes the inability to conserve mass (DeSilva et al. [15]). Furthermore, current limitations in spatial resolution and processing algorithms make these techniques more prone to errors when calculating fields that are derived from the velocity field, for example pressure near surfaces from the Poissons equation (Patil [35]) or vorticity. CFD on the other hand, can produce fields that satisfy continuity and can provide derivatives of the velocity and pressure fields with sufficient accuracy. CFD also has the advantage of being very versatile in that it is possible to run multiple design configurations and simulations without the excessive costs incurred in manufacturing and wind tunnel testing, provided the computational costs are minimized.

CFD is an umbrella term encompassing many types of methods and practices (Versteeg and Malalasekara [55]). At the most fundamental level, any type of fluid flow can be simulated using the full set of Navier Stokes equations to represent all length and timescales in the flow. This type of simulation, also known as Direct Numerical Simulation (DNS) is extremely expensive and is generally confined to academic research due to current limitations in computer hardware. Another solution is to perform a Large Eddy Simulation (LES), which uses a filter to remove the small scales of turbulent motion and model them, while still solving the large scales. Although LES is very effective and quickly gaining popularity, it is still infeasible for many industries looking to achieve quick solutions to flow problems. The most popular method of simulating fluid flows

[1] Hereafter referred to as CFD
[2] Hereafter referred to as PIV
[3] Hereafter referred to as HFSB

involve solving a time averaged version of the Navier Stokes equations known as Reynolds Averaged Navier Stokes [4] equations. In RANS, the time averaged flow fields are produced and the effect of unsteady components on the mean flow is modeled through a term called the Reynolds stresses, which manifests itself during the derivation of the equations. Since the effect of unsteady turbulence is modeled, a lot of different models exist to tackle different flow situations and each model has parameters which are tuned to work only in those situations (Catalano and Amato [11]). Clearly, accuracy and computational cost are conflicting parameters that a person has to choose between while trying to simulate any fluid flow.

With the proliferation of aerodynamics and fluid dynamics in different industries for engineering design, the need for cheap and accurate solutions to fluid flow problems of different scales and behaviours is growing rapidly. Keeping in mind the limitations of both experimental and computational techniques, there is a growing interest to leverage the advantages that each field provides to help compensate mutual weaknesses in each other. Data Assimilation [5] is a technique that has emerged as an option to achieve this goal. DA is a mathematical field that aims at combining theory with observations. It was first introduced in weather prediction and since then, different methods of DA have been used in various disciplines including fluid mechanics. Different methods of DA that are used for fluid mechanics include variational methods, the use of Kalman filters and state observer based DA (Hayase [22]). A more comprehensive review of some of the methods of DA used for fluid mechanics are given in section 2.2.2. The state observer method, which is computationally cheaper and easier to implement than other methods is based on the principle of feedback. As a CFD solution progresses, feedback based on the deviation from empirical observation is used to correct the simulation. Furthermore, this method does not need the measurement noise to be modeled as a parameter, making the integration(assimilation) of experimental data easier. Previous uses of the state observer method for DA in fluid mechanics have however been limited. That paved the way to define the research objective of this thesis assignment:

Development and assessment of a state-observer based data assimilation method for steady Reynolds averaged Navier-Stokes simulation using a complex 3-dimensional flow around a bluff body as a test case.

In this thesis, the working principle of the state observer based DA method along with its implementation for a steady RANS solver is provided in chapter 3. Then, the method is applied on the 1-D viscous Burgers equation by assimilating a non-physical[6] forcing solution [7] . Following this is an extension of the principle to 2-D flow where a steady RANS simulation using the k-ω SST turbulence model on the flow around a half cylinder is imposed as the ground truth on a simulation of the same half cylinder using the Spallart-Allmaras turbulence model. The methodology and results are presented in chapter 4. Next, a wind tunnel experiment is performed where the complex 3-D flow around a simplified car side mirror model is captured using coaxial volumetric velocity measurements mounted on a robotic arm and the time averaged velocity field is used as the ground truth for a steady RANS simulation of the same model. The experimental setup, procedure and data reduction techniques are provided in chapter 5. The results of the experiment along with the results of implementing the method on the 3-D case using the experimental data are given in chapter 6. Finally some conclusions based on the results and recommendations for improving the method are provided in chapter 7

[4] Hereafter referred to as RANS

[5] Hereafter referred to as DA

[6] Here non-physical means that the forcing solution is different from the actual solution of the Burgers' equation given a set of initial and boundary conditions

[7] The forcing solution is also referred to as the ground truth or experimental velocity

2

The integration of computational and experimental fluid dynamics

In this chapter, relevant literature and background for the present thesis assignment are provided. The first two sections, 2.1 and 2.2 deal with the integration of computational and experimental techniques. Specifically, section 2.2 discusses briefly the history of DA and the different ways it has been used in fluid mechanics. Section 2.3 gives a brief introduction to Particle Image Velocimetry as an experimental technique and how it can be used to generate the experimental data required for the present thesis. Section 2.4 deals with the flow around a car side mirror which is the experimental test case used in the present study. Finally, the chapter ends with a discussion on the research questions and objectives in sections 2.5 and 2.6 respectively.

2.1. The use of numerical methods on experimental data

Despite CFD and experimental fluid dynamics generally being treated as separate fields, at the most fundamental level they both involve data manipulation. Hence it comes as no surprise that there are many situations where the two fields overlap and common techniques are used. CFD involves solving the differential equations that govern fluid flow with some approximations and experimental techniques usually involve gathering data from a physical experiment whether it is velocity using PIV or a hot wire, pressure using probes or temperature using thermometers. It is also well accepted that experimental data will involve noise that manifest themselves in the form of measurement errors. Furthermore, techniques like PIV or Particle Tracking Velocimetry(PTV) which are the most widely used techniques to obtain velocity due to their ability to extract whole fields while being non intrusive, derive the velocity fields based on algorithms like cross correlation, or iterative particle reconstruction which do not abide by the governing equations and thus may result in spurious velocity vectors (Westerweel [56]) and fields that do not satisfy continuity. Due to this a large number of researchers have worked to improve experimental data by applying numerical techniques to the data.

DeSilva et al. [15] define a constrained minimization problem where the objective function is defined as the discrepancy between the measured and filtered velocity fields and the constraint is the divergence free condition.They test their method on experimental results from a turbulent channel flow and find that it reduces the divergence error in tomographic PIV to nearly zero. Azijli and Dwight [5] investigate the use of Gaussian process regression to filter spurious divergence. Application of their solenoidal filter to experimental and synthetic test cases returns more accurate fields of velocity, vorticity and pressure. Numerical techniques have been used not only to improve the spatial velocity fields but also to improve temporal information from a time resolved PIV experiment. Scarano and Moore [40] develop an advection based model to increase the temporal resolution of PIV. They use Taylors hypothesis of frozen turbulence and apply supersampling or in the authors' words, *pouring Space in Time* to extract velocity information at times where the physical experiment does not have data thereby increasing the temporal resolution of the measurement. CFD in the more traditional sense, that is solving the governing equations numerically, has also been used to improve experimental data. PIV velocity data can often have gaps that are created by shadows or areas with lower seeding concentrations. Sciacchitano et al. [46] outline a method where they extract velocity fields from the regions

(a) Experimental Data.

(b) Solenoidal filter from [15] applied.

Figure 2.1: Divergence of velocity at 0.1 Hz for flow around a car side mirror.

with no data by solving the unsteady incompressible Navier-Stokes equation in those regions. They use a finite volume discretization scheme and use the velocity at the boundary of the gaps as the initial conditions.

Numerical methods have not only been applied to enhance experimental data but also to derive subsequent information. Pressure is a significant physical quantity that provides important insights into flow behaviour or force distribution around objects. Pressure is measured both directly through pressure probes, which are intrusive, and also indirectly from PIV measurements. Pressure is calculated from a velocity field by the application of the pressure Poisson equation. Van Oudheusden [54] in his review paper discusses different numerical implementation strategies for different parts of the pressure poisson equation and how they affect the accuracy of the static pressure computations in a flow field.

At this point it must be noted that even though experimental measurement techniques are known to be corrupted by some level of noise, their importance is not diminished and their necessity cannot be understated. Experiments provide us with true physical behaviour of fluid flows without any modeling. A classic example and one of the most important concepts in fluid mechanics is the law of the wall which describes fluid flow near the wall and how that affects the overall behaviour of the flow. Coles [13] gives a detailed history of the law of the wall and although the initial formulations were based on dimensional arguments, all the early work confirming them were experiments, mostly using hot wire probes. This concept is so significant that most RANS formulations that use wall functions [1] base their calculations on this universal law. Several DNS studies of either channel flows or flat plate boundary layers also validate their findings with existing experimental results. A complete survey of experimental data used to either validate or provide model constants for CFD is beyond the scope of this work, but it is well accepted that experimental data is used as the *ground truth* on which the accuracy of different numerical techniques are judged. Jameson and Martinelli [27] and Tinoco [53] discuss the different ways of reducing errors in CFD and its validation with experimental data.

2.2. The use of experimental data for numerical simulations

Clearly, there has been widespread use of numerical techniques to either improve or extract further information from experimental data. However the use of experimental data directly in numerical simulations other than for validation purposes has been surprisingly limited. The first real use of such techniques as mentioned in chapter 1 was in weather prediction. To lay the foundations for the goals of this research work, it is essential to understand how the genesis of the technique of using experimental data for numerical simulations lay in Numerical Weather Prediction [2] and its evolution and eventual propagation to other disciplines.

Lewis Fry Richardson [39] in 1922 was the first person to try and numerically forecast the weather. He used the governing equations and a finite difference discretization scheme to integrate forward in time. The domain

[1] Explained briefly in section 4.2.1

[2] Hereafter referred to as NWP

he used was Europe and referring to figure 2.2, pressure was defined at points marked by P and momentum at points marked by M. He used observations from stations named with a circled cross as the initial conditions for his simulation. His main challenge was to interpolate data from the irregular observation points to the whole grid. Unfortunately his efforts failed but led to the development of the field of weather prediction and data assimilation. NWP is essentially an *initial value problem* where observations from the atmosphere are integrated into a numerical model as the initial conditions and future states of the atmosphere are simulated.

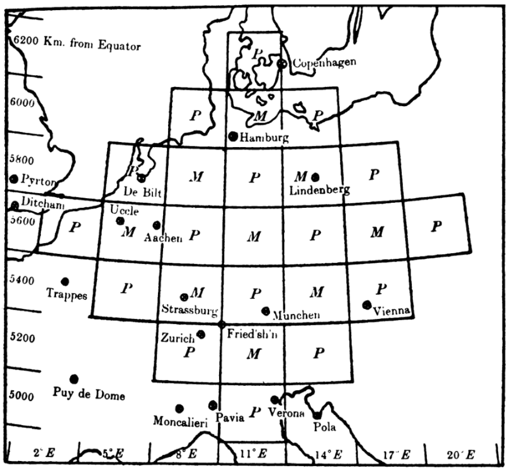


Figure 2.2: Richardson's domain. Replicated from [14].

Talagrand [52] provides a very succinct definition of data assimilation as it pertains to its meteorological origins. According to him, the word assimilation was coined in the late sixties for denoting a process in which observations distributed in time are merged together with a dynamic numerical model of the flow in order to determine as accurately as possible the state of the atmosphere. The available information essentially consists of the observations proper, and of the physical laws that govern the evolution of the flow. The latter are available in practice under the form of a numerical model. In NWP, the initial conditions are of extreme importance due to the non linear dynamic nature of the equations. Furthermore, with observations of the atmosphere being so sparse and spread out compared to the scale of the world, the whole basis of DA was to find techniques in which the observations could be integrated into the background field with as little error as possible to serve as the initial conditions for the simulations. He further describes the two main types of algorithms that exist for data assimilation.

- Sequential DA
- Variational DA

Sequential DA, as the name suggests is a process in which observations are sequentially integrated into the numerical model. The simulation progresses up to the point where observations are available and then the state vector is updated by integrating the observation using different methods outlined further in this report. Sequential DA has the property that the integrated observations affect future states of the simulation and the solution asymptotically converges to the observations.

Variational DA on the other hand, involves solving an optimization or minimization problem to reduce the errors between the observations and the simulations. According to Talagrand, variational DA aims at globally adjusting a model solution to all observations available over the assimilation period. The adjustments being simultaneous, the adjusted states at all times are influenced by all the observations over the assimilating period. This characteristic makes variational DA uniquely different from sequential DA in that it affects model solutions at all times an not just in the future. Variational methods are in general computationally more expensive than sequential methods because they involve solving an optimization problem in addition to the governing equations.

Kalnay [30] in her book describes different methods of sequential and variational DA. In sequential DA, she brings out a further distinction between two methods. Since observations from the atmosphere are made by instruments which are assumed to have a known precision, many techniques of sequential DA use probabilistic methods or statistical estimation to integrate these measurements with the background field onto the whole domain. This method of statistical estimation DA is the most popular and used for weather prediction. Many similar problems of state estimation are encountered in other fields. Talagrand provides different examples of these as well. An airplane using sensors to best estimate its position in space and time uses state estimation theory. The Kalman filter, which originated in electrical engineering is a famous tool in this regard. In plasma physics, researchers try to estimate the internal state of a system based on observations from the surface. Statistical estimation DA in NWP shares many techniques with all these fields.

The other sequential DA method that Kalnay outlines is called empirical analysis schemes where the observations from the atmosphere are used directly without probabilistic methods to influence the state vectors. Figure 2.3 shows the different forms of DA as outlined by Kalnay.

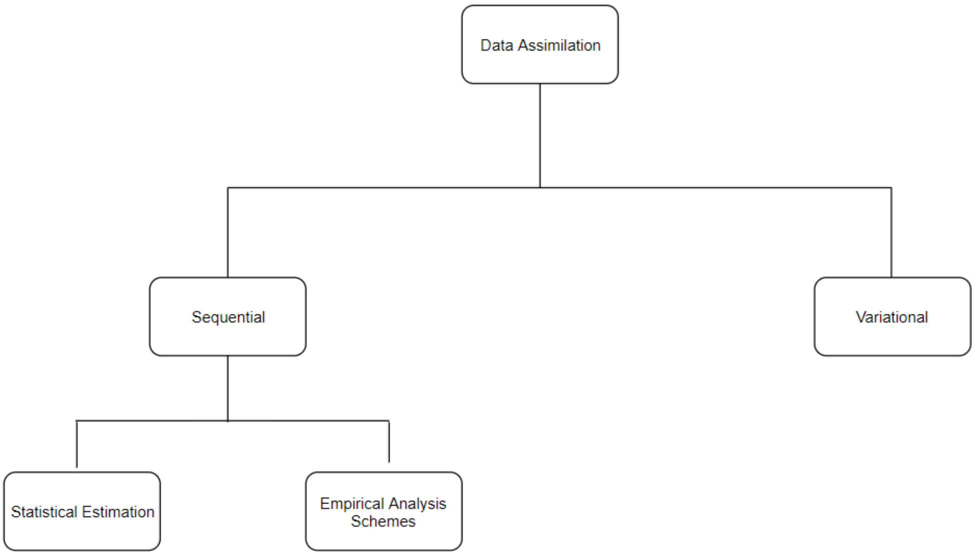


Figure 2.3: Different methods of DA.

Empirical analysis schemes are used when there are no available statistics to perform a probabilistic analysis. One of the empirical analysis methods given in Kalnay [30] is called *Nudging*.

2.2.1. Nudging

Nudging or Newtonian relaxation is a method where a term is added to the governing equation of a model that is proportional to the difference in the observations and simulations. What this term essentially does is to “nudge" the solution to the observational values. This is represented in equation 2.1 for an incompressible flow. The equation is basically the conservative form of the momentum equation.

$$\frac{\partial u}{\partial t} = -u \cdot \nabla u - \frac{1}{\rho}\nabla p + \nu \nabla^2 u + \frac{u_{obs} - u}{\tau_u} \tag{2.1}$$

Here, u is the velocity, t is time, ν is the kinematic viscosity, p is the pressure and τ_u is a time scale. The last term on the right represents the source term added in the equation that nudges the solution towards the observational velocity. Since in this equation the term has to be an acceleration to satisfy dimensional constraints, the difference in velocity is divided by a timescale. The timescale that is chosen is extremely important in terms of stability of the solution. It also has to be chosen and scaled with respect to the other terms in the equation. This model is not generally used in NWP for large domains simply because the observation data may be too sparse over a large area. However as will be shown subsequently, this method has been used for fluid mechanics and is the basis of the current research work as well.

From the brief introduction to DA and its genesis, certain points can be garnered. First, NWP is an initial value problem and the observations are sparsely distributed in both space and time. The sparsity of data is what drove the research in DA for NWP to be centered around the goal of trying to interpolate observations on to the numerical grid as the initial conditions for the simulations with as little error as possible. A second observation is that the techniques used for DA can be used and have been used for a bevy of different fields. Fluid mechanics in particular shares many similarities with NWP in both the constitutive equations and the simulation algorithms thus making the use of DA for other fluid mechanics problems an obvious choice.

2.2.2. DA in fluid mechanics

The use of DA in fluid mechanics has been steadily on the rise. The structure of the algorithms used for fluid mechanics has remained the same as shown in figure 2.3. Hayase [22] provides an excellent review of the different uses of DA for fluid mechanics. The use of different methods of DA for flow related problems are shown in figure 2.4. The Kalman filter and variational methods have been the most popular and state observers, a technique which is explained in section 2.2.2, have not been explored as much.

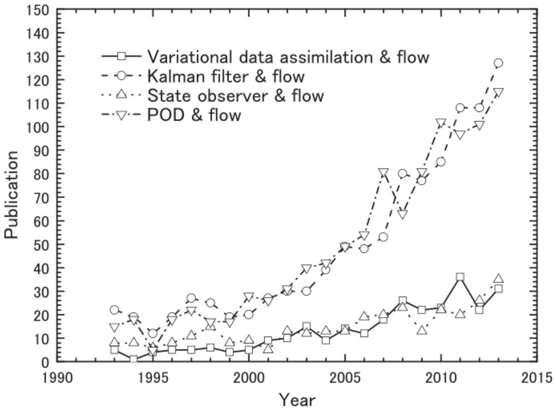


Figure 2.4: Uses of DA for flow related problems in publications. Reproduced from [22].

DA based on statistical estimation theory

Statistical estimation theory is a field that has been used by a lot of disciplines. It is based on the theory of least-square estimation, maximum likelihood, bayesian techniques and other probabilistic tools. The Kalman filter, which was introduced in electrical engineering is a popular estimation tool. It is an algorithm that best produces the estimated state of a dynamical system using measurements that are known to be noisy and a physical model of the system. It has been used extensively for many engineering problems and has also been used for fluid mechanics. It has especially been used extensively for flow control related problems. In feedback driven flow control, it is necessary to estimate the state of a system and then provide a feedback signal to an actuator to modify the flow accordingly. A lot of work has been done on flow control using the Kalman filter including controlling wall bounded flows. Since flow control involves physical measurement of a flow parameter and using that to generate a feedback, the Kalman filter is ideal to estimate the flow state using the limited noisy measurements. Hoepffner et al.[25] and Chevalier et al. [12] discuss the use of the Kalman filter along with a linearized version of the Navier-Stokes equations for wall bounded laminar and turbulent flows respectively.

Takao Suzuki and his group have developed a hybrid algorithm that combines PTV measurements and DNS simulations. They have done this using three different algorithms with increasing levels of fidelity. These include a POD-Galerkin approach [50] , what they describe as a “conventional" approach [51] and the highest fidelity algorithm which uses a reduced order Kalman filter approach [49]. The conventional algorithm uses a state observer based method and will be discussed further in section 2.2.2. Basically, they conduct a time resolved PTV experiment on a planar jet at Re 2000 and integrate the measurements obtained from the experiment onto a DNS simulation of the same flow. They match the acquisition rate of the experiment to match a multiple of the time step of the CFD simulation. In their method, they apply DA on an unsteady flow and hence the Reynolds number of the flow becomes an important parameter. If the Reynolds number increases, the computational time step decreases and it becomes harder for the PTV to maintain that resolution. To extend the applicability to noisy time- resolved PIV data, the proposed algorithm optimizes the data input temporally and spatially by introducing a reduced-order Kalman filter.

Typically, statistical estimation theory is used in situations where a large state vector is to be reconstructed from limited and noisy measurements. This is why it is the most popular in NWP and in flow control. Furthermore, all the methods must model measurement error as a parameter in the models.

DA based on Variational methods

The variational method is a computationally intensive method requiring the solution of an optimization problem along with the numerical simulation. However while sequential methods like statistical estimation methods and the state observer reduce the error asymptotically, variational methods provides solution that agree with the measurement data throughout the simulation. Foures et al. [21] define a data assimilation method based on a variational formulation and Lagrange multipliers to enforce the Navier Stokes equations. It is applied for reconstructing the mean flow around an infinite cylinder at Re 150. The base solution is taken by time averaging instantaneous flow fields from a DNS simulation of the same flow. The method is implemented by applying a forcing term in the steady RANS equations. For their study they choose the Reynolds stress as the control parameter and arrive at an optimal forcing where the error between the simulation and measurement is minimized. The steady RANS equation is shown in equation 2.2. When deriving the RANS equation, the source term comes out as the Reynolds stresses. The Reynolds stresses constitute the closure problem for turbulent flows. Since they do not explicitly depend on the mean flow, it can be considered as an independent forcing term and the authors have chosen that term as the control parameter.

$$\begin{gathered}\bar{u}\cdot\nabla\bar{u}+\nabla\bar{p}-Re^{-1}\nabla^2\bar{u}=f\\ f=-\nabla\cdot R\\ R_{ij}=\overline{{u_i}'{u_j}'}\end{gathered}\tag{2.2}$$

An advantage this method has is that the algorithm is defined in such a way that even partial measurements, such as encountered in actual experiments can be used to recreate with low error the full state vector. The method has also be tested with noise and was found to be robust even with a signal to noise ratio of 100 %.

DA based on the state observer

The state observer is an element of control theory. It is based on the principle of feedback. Based on the difference between measurement and computation, a signal is sent to the model through a feedback law. The principle of operation is very similar to nudging as discussed in section 2.2.1. The state observer offer an advantage over other methods of DA due to their simplicity and flexibility in the design of the feedback gain. The Kalman filter and variational methods have been popular in both NWP and DA used for fluid mechanics mainly due to the fact that observations were scarce and sometimes not evenly distributed in time or space. However, with the advent of large scale PIV mesurements as will be discussed in section 2.3, the use of a state observer becomes much more intriguing. The following section discusses the different uses of the state observer method for fluid flows in different literature.

Hayase and Hayashi [23] use a state observer to replicate the physical flow in a turbulent flow through a duct of square cross section. A SIMPLER algorithm is used as the mathematical model for the real flow. The dynamical behaviour of the flow is governed by the incompressible Navier-Stokes equations. The flow domain is shown in figure 2.5. For the “measured physical solution" a plane is taken downstream of the inlet and an output is taken as the x component of velocity at 10x10 grid points at the plane for a simulation on the same grid without any forcing. The error e is calculated as the difference between the physical solution and the hybrid solution. For the input condition, the pressure boundary condition is modified through a simple proportional feedback control law as shown in equation 2.3. The pressure correction is added to the boundary elements which correspond to the grid nodes where the error is measured. It must be noted here that the authors did not use a source term in the Navier Stokes equations as the feedback was given directly to the boundary condition of the flow.

$$\begin{gathered}\delta p_{jk}=K_P e_{jk}\\ \Delta p_{jk}=\Delta p+\delta p_{jk},(j,k=1,2,3..10)\end{gathered}\tag{2.3}$$

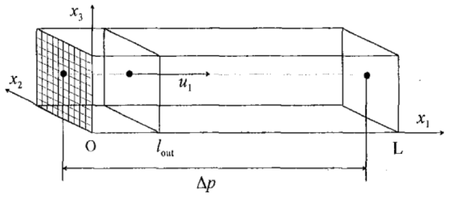


Figure 2.5: Domain used . Reproduced from [23].

Here, Δp_{jk} is the pressure correction at the inlet boundary condition. K_P is the proportional feedback gain. The authors take a plane 7.5 % of the domain from the inlet boundary. Figure 2.6 shows the average estimation error over all the points for different gains. The error is seen to decrease with the use of the observer

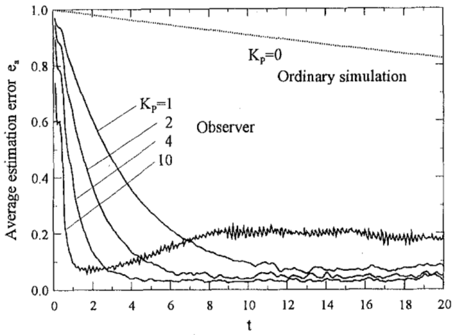


Figure 2.6: Average estimation error vs time . Reproduced from [23].

and the authors claim reduced error in estimation of the velocity by a factor of 0.6 in the whole domain or a factor of 0.3 behind the output measurement plane in comparison with the ordinary flow simulation without feedback.

Imagawa and Hayase [26] follow up the previous paper with a more detailed study on the effect of the number of forcing points and the value of the feedback constant. In the previous study, an adhoc feedback law was designed by taking a random plane and using velocity in that plane to calculate the error and using that to update the pressure boundary condition at the inlet. In the current study, they change two features of their model. First, they add a source term in the Navier Stokes equation and model it on the velocity difference at each control volume similar to what is shown in equation 2.1 with the difference being that the inverse of the timescale is interpreted as the feedback gain to the system. Second, they calculate the velocity difference at all grid points of the model and then perform three case studies.

- The first case study is to apply feedback at all the control volumes using all three velocity components.
- The second case is to force at all the control volumes but only using partial velocity components.
- The final case is to force at only a few control volumes but using all the velocity components.

Figure 2.7 shows the norm of the estimation error vs time for case 1 where all the points and all the velocity components are forced. The error drops drastically for K_p between 1-16 but slows down upto 40. For higher gains, the model is unstable. Figure 2.8 shows the norm of the estimation error for cases with only partial velocity components forced. Here, u_1 corresponds to streamwise direction, u_2 is the spanwise component and u_3 is the vertical component of velocity. From the graphs, the importance of forcing the streamwise direction is highlighted. For the case with only partial points forced, they used a single point at the center of the domain as the forced control volume. Their analysis showed that the error reduced downstream of the point but the time taken to reach a steady state solution was much longer. They also found that increasing the feedback gain at some point destabilizes the feedback control leading to diverging solutions.
Nisugi et al. [33] apply the technique implemented by Hayase and Hayashi[23] for a real flow where the data is taken from a physical experiment. They call their technique a hybrid wind tunnel which integrates

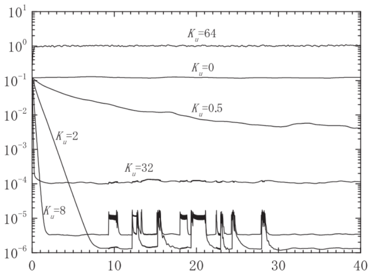
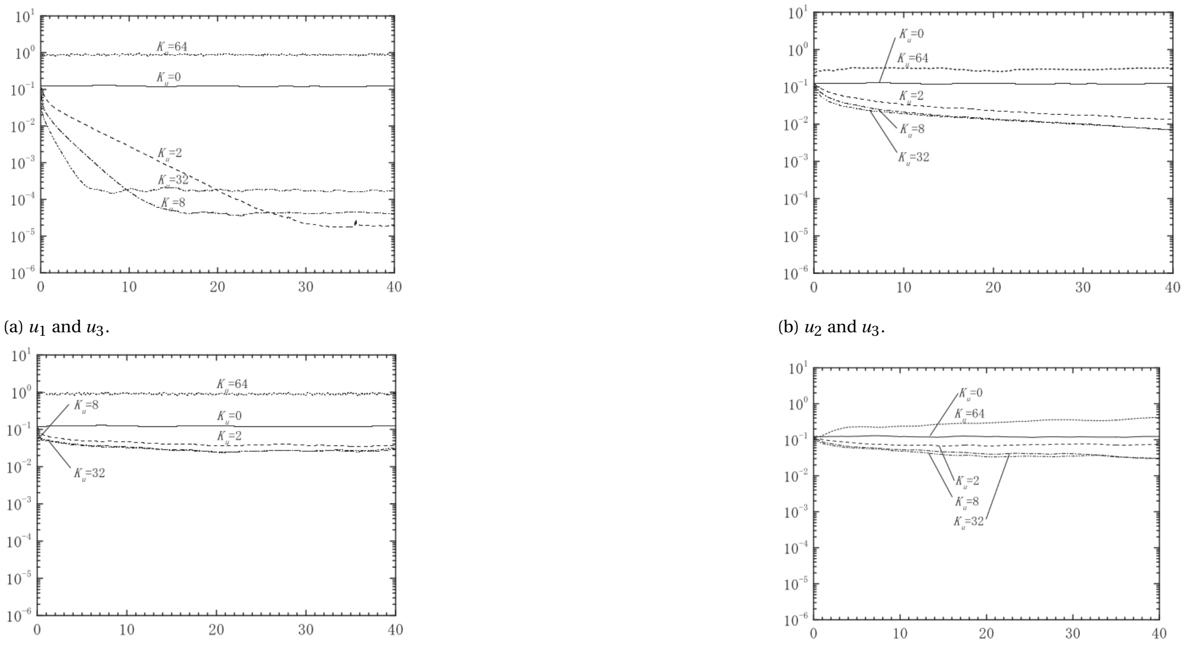


Figure 2.7: Error norm vs time for case 1. Reproduced from [26].



(a) u_1 and u_3.

(b) u_2 and u_3.

(c) u_1.

(d) u_2.

Figure 2.8: Error norm vs time for the different components of velocity (case 2). Reproduced from [26].

experimental measurement with a wind tunnel with a corresponding numerical simulation. They perform the analysis for the fundamental flow of the Karman vortex street in the wake of a square cylinder at an Re_D of 1200. For this model, the authors choose to provide two signals. One feedback signal to compensate the error the pressure on the side walls of the cylinder and the feed-forward signal to adjust the upstream velocity boundary condition. The pressure term forcing is applied by adding a body forcing term in the Navier Stokes equations to the concerned control volume where the error is calculated. The forcing is calculated as shown in equation 2.4 The authors perform a physical experiment in the wind tunnel where they measure pressure in the points shown in figure 2.9 by using pressure taps . They also measure the x component of velocity with a Laser Doppler Anemometer(LDA) system at 413 points which are depicted as the grid points in figure 2.9. Figure 2.10 shows the hybrid wind tunnel system along with the control volumes A and B where the forcing is applied. S represents the stagnation point.

$$\begin{aligned} f_A &= -KA_C * (P_{AS} - {P_{AS}}^*) \\ f_B &= -KA_C * (P_{BS} - {P_{BS}}^*) \end{aligned} \tag{2.4}$$

Here, P_{AS} represents the difference in pressure between the control volume A and the stagnation point. The star indicates the experimental value while the terms without the star represent the value from the hybrid model. The main aim of the authors is to replicate the shedding characteristics of the wind tunnel experiment in the numerical simulation. For the given Re shedding was observed at 2.75 Hz. As compared with the ordinary simulation, the hybrid wind tunnel substantially improves the accuracy and the efficiency in

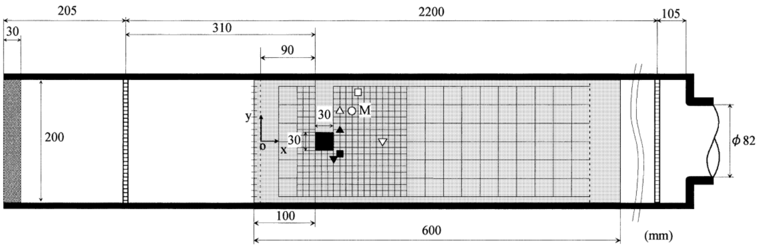


Figure 2.9: Wind tunnel experiment in [33].

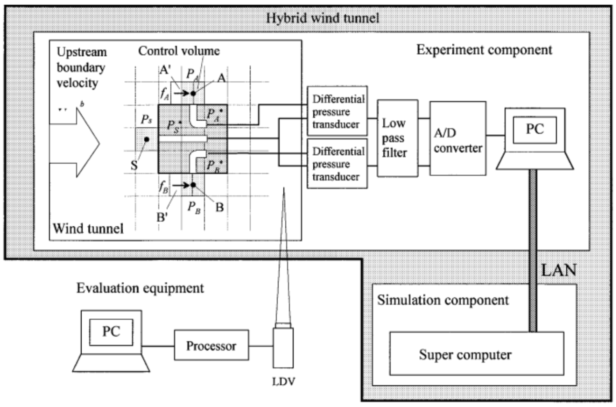


Figure 2.10: Hybrid model setup in [33].

the analysis of the relevant flow. Especially, the oscillation of the flow with the hybrid wind tunnel exactly synchronizes with that of the experiment, while that with the ordinary simulation never behave like that. In comparison with the experiment, the hybrid wind tunnel provides more detailed information of the flow than the experiment does. The analysis performed in this study is for an assumed 2-dimensional flow.

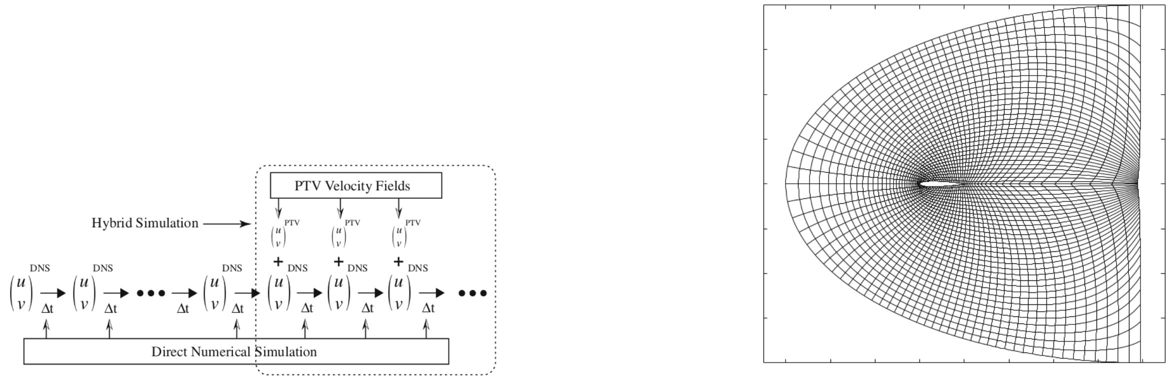
With measurement techniques like PIV/PTV emerging to map whole flow fields, their use in DA has become imminent for some time. Suzuki et al. [51] develop a technique that uses velocity fields from a PTV experiment as the ground truth. The hybrid simulation they run is to integrate PTV velocity fields onto a DNS simulation by matching the acquisition frequency of the PTV with the time step of the DNS. The investigated problem is a flow past a NACA 0012 airfoil at an angle of attack of 15 degrees and a Re of 1300.The schematic of their method and the numerical domain are shown in figure 2.11 The velocity fields obtained from the PTV experiment are first pre-processed to ensure that they satisfy continuity. This is done by minimizing a cost function that involves the rms error between measured and hybrid velocity fields and the continuity constraint multiplied by a lagrange multiplier.

The hybrid velocity field is calculated as shown in equation 2.5.

$$\begin{aligned} u_{hybrid} &= r * u_{DNS} + (1-r) * (u_{PTV}) \\ v_{hybrid} &= r * v_{DNS} + (1-r) * (v_{PTV}) \end{aligned} \tag{2.5}$$

Here, r is the weight function and is assigned a value between 0 and 1. It is constant in space and time. For this study it was assigned a value of 0.75 so that the weight of the PTV field is 25 percent. Since the PTV data is available only in the field of view of the camera, and the domain of the CFD simulation is typically larger than that of the PTV domain, Suzuki uses a patch function $f(x,y)$ which takes a value of unity of 1 in the middle of the domain where the PTV data is available and 0 outside of it. Equation 2.5 is modified as shown in equation 2.6.

The authors find that with the application of the hybrid method, the flow field gradually converges to the PTV field. The original DNS simulation predicts large scale vortex shedding, while the PTV data does not.



(a) Schematic of hybrid method.

(b) 2D domain used for both the DNS and the hybrid simulations.

Figure 2.11: Left: Schematic showing the working of their method, Right: Domain on which analysis was conducted. Reproduced from [51].

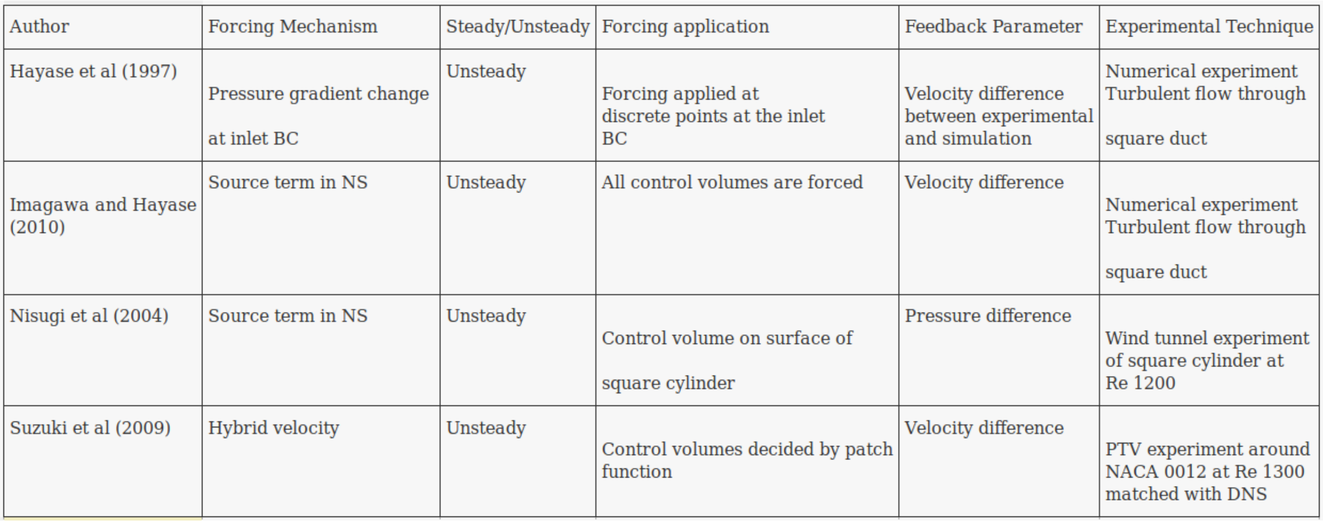
the hybrid simulations show that fluctuations of the shear layer are significantly suppressed and vortices are generated farther downstream. Thus the hybrid simulation is able to reconstruct the PTV flow field.

$$\begin{aligned} u_{hybrid} &= (1-(1-r)f) * u_{DNS} + (1-r)f * (u_{PTV}) \\ v_{hybrid} &= (1-(1-r)f) * v_{DNS} + (1-r)f * (v_{PTV}) \end{aligned} \tag{2.6}$$

To compare the different techniques and methods that have been used for state observer based DA, refer to figure 2.12. It can be seen that the use of the state observer in DA for fluid mechanics is still in its nascent stage. There are a few points that can be noted here

- The studies performed till now have either used CFD as the ground truth or in cases where experiments are used only a 2-D flow is considered.
- It is also clear that there is no single accepted way to adopt a forcing technique. Forcing at different points of the domain using different forcing functions have all produced fairly successful and varied results.
- All the studies have performed DA on unsteady flows.

There is a large scope for taking forward this technique to make it useful for industry purposes. In the industry, where computationally cheap and accurate solutions of the flow are desired and the aerodynamics typically involve complex three dimensional features, it becomes imperative that any technique that is designed for use in the industry must be able to extend the method to 3 dimensional flows at low computational costs. The lowest computational cost for CFD at the moment are RANS solvers. Thus, the logical step to take is to study the feasibility and extension of this technique to 3 dimensional flows and the integration of 3 dimensional velocity fields derived from experiments into RANS simulations.

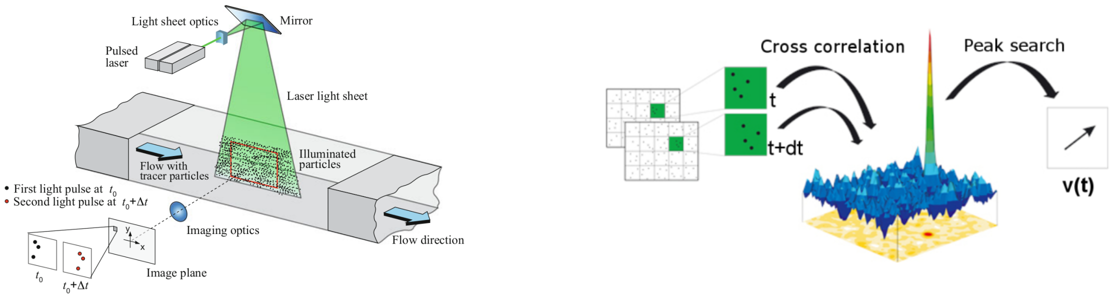


Author	Forcing Mechanism	Steady/Unsteady	Forcing application	Feedback Parameter	Experimental Technique
Hayase et al (1997)	Pressure gradient change at inlet BC	Unsteady	Forcing applied at discrete points at the inlet BC	Velocity difference between experimental and simulation	Numerical experiment Turbulent flow through square duct
Imagawa and Hayase (2010)	Source term in NS	Unsteady	All control volumes are forced	Velocity difference	Numerical experiment Turbulent flow through square duct
Nisugi et al (2004)	Source term in NS	Unsteady	Control volume on surface of square cylinder	Pressure difference	Wind tunnel experiment of square cylinder at Re 1200
Suzuki et al (2009)	Hybrid velocity	Unsteady	Control volumes decided by patch function	Velocity difference	PTV experiment around NACA 0012 at Re 1300 matched with DNS

Figure 2.12: Different methods of state observer based DA used in fluid mechanics.

2.3. Particle Image Velocimetry

Quantitative analysis of fluid flow in theory is not difficult. It is not hard to imagine a leaf floating on a river and trying to calculate the velocity of the water by estimating the distance the leaf travelled in a set time. However, accurate analysis of fluid velocity at a large number of points simultaneously is a technique that has only been developed in the last few decades. Particle Image Velocimetry or PIV is a non-intrusive, quantitative, optical technique of flow visualization. Basically, a fluid flow is seeded with tracer particles that adequately follow the flow, a source of illumination to highlight the tracers, and an optical system to capture the motion. The idea is that a camera captures in quick succession two pictures of the flow seeded with particles and by analysing the displacement of the particles by use of a statistical method called cross correlation, the velocity of the flow field can be deduced(Displacement divided by the time between two images Δt). Originally, PIV as introduced by Adrian [2] presented a planar technique where a laser sheet illuminated a plane on which the camera was focused(Field of View [3]). This technique is referred to as 2C-2D(2 Components 2 Dimensions). Figure 2.13 depicts the principles involved in planar PIV and the post processing of the images to acquire the velocity fields. For a comprehensive read on PIV techniques the reader is encouraged to refer to Raffel et al. [37].



(a) Experimental Setup of PIV experiment. Reproduced from [37].

(b) Cross correlation to produce velocity vectors. Reproduced from [59].

Figure 2.13: Basics of PIV.

The main components in a typical PIV experiment as described in Raffel et al. [37] include:

- Tracer Particles or Seeding
- Illumination
- Recording

[3] Hereafter referred to as FOV

- Calibration
- Evaluation
- Post-Processing

Clearly, PIV depends on a lot of external factors that have little to do with fluid mechanics and the development of the field has echoed this fact. The development of PIV has hinged on the advancements in techniques in other fields like digital imaging and image processing algorithms. Despite 2C-2D PIV being instrumental to a number of research works and understanding of fluid behaviour, there was a need to extend the technique to three dimensional flows. Since most flows encompass large scale three dimensional behaviour (3C-3D), PIV could only be viable for the industry if this requirement could be met (Jux([28])). The first notable breakthrough to volumetric PIV was provided by Elsinga et al. [19] called tomographic PIV which was able to reliably produce 3C-3D information of the flow, but there were numerous intermediate steps between the first introduction to planar PIV and tomographic PIV, a good review of which can be found in [3].

2.3.1. Tomographic PIV and its limitations

Due to the inherent property of cameras projecting the images they take onto a single plane, it becomes hard to reconstruct a 3D volume from 2D images. The most intuitive way to do this is to reconstruct particles based on triangulation by intersection of lines of sight from different cameras. However this method suffers from the production of “ghost particles" as the triangulation error is increased thus making it unsuitable for large particle concentrations. In this form of particle reconstruction, the particle information is stored in the form of position and intensity only. There is another way to perform particle reconstruction where instead of individual particles, the volume to be reconstructed is discretized into “voxels". The voxel is then used to represent light intensity. One of the methods using the voxel approach is the Multiplicative Algebraic Reconstruction Technique(MART) as introduced by Herman and lent [24]. MART uses an algebraic iterative technique where the 3D volume is discretized into “voxels" and the intensity of the particles distributed in the voxels are related to the intensity of the 2D projections using a linear relationship. MART has been proven as a successful technique for reconstructing particles even at image densities up to 0.05 particles per pixel(ppp). MART was one of the most widely used method in the initial stages of tomographic PIV. In a typical tomographic PIV experiment, three or more camera are focused on an illuminated volume. Following the capture of the multiple images, the particle distribution is calculated and finally the extension of the 2D cross correlation technique to 3D enables the calculation of the velocity vectors in the volume. A schematic of the process is shown in figure 2.14 Tomographic PIV however, despite being a major breakthrough in quantitative flow visualization, still presented many limitations. First, the setup required for an experiment is very involved and limits optical access sometimes. Second, the MART algorithm, especially if used for time resolved experiments can be prohibitive. Another limitation is the small volumes that can be measured due to limited pulse energy from the lasers and the scattering efficiency of the tracer particles (Scarano et al. [41]).

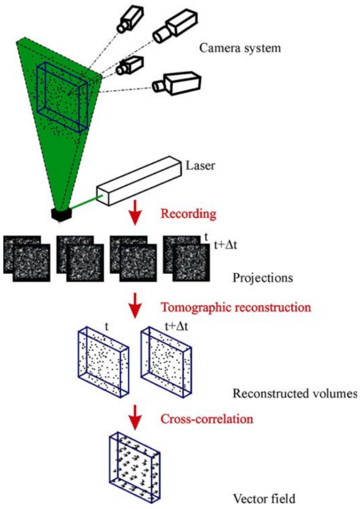


Figure 2.14: Schematic of tomographic PIV. Reproduced from [19].

2.3.2. PIV moving to larger volumes

Tomographic PIV as it was initially used was extremely important in the research community, but due to the aforementioned limitations, was still not feasible for industries looking to perform large scale measurements. Up till the publication of Scarano et al.[41], the maximum volume that was measured was reported to be 16x22x8 cm^3. As mentioned in the limitations in section 2.3, the developments in PIV were due to advancements in fields other than fluid mechanics. The increase in measurement volume too was spearheaded by two changes in the process and these are discussed in brief detail below.

Tracer particles

For a tracer particle to be useful for a PIV experiment, it must satisfy two main criterion. First, it should be able to adequately follow the flow. In technical terms, a parameter called slip velocity, whose formula is shown in equation 2.7(where d_p is the particle diameter,ρ_p is the particle density,ρ_f is the fluid density, U is the fluid velocity, V is the particle velocity and μ is the dynamic viscosity of the fluid), is used to judge whether a particle can follow a flow. The formula is derived by conducting a force balance on a particle in a fluid. From inspection of the formula, it is clear that if the particle density matches that of the fluid, or in other words if the particle is *neutrally buoyant*, the slip velocity is zero and the particle can adequately follow the flow. This makes sense intuitively as a particle with the same inertial property as the fluid (density) can be considered as a packet of the fluid itself. However, most particles are not neutrally buoyant if air flows are considered and their density is generally orders of magnitude higher than the fluid ($\rho_p >> \rho_f$). Solution of equation 2.7 which is a differential equation provides a parameter called the time response (τ_p) of the particle as given in equation 2.8. For the particle to adequately follow the flow, the time response should be smaller than the smallest time scale of the flow. Clearly, as the particle diameter increases, the time response also increases, mandating the particle size to be kept as small as possible.

$$U_{slip} = U - V = -\frac{{d_p}^2}{18}\frac{\rho_p - \rho_f}{\mu}\frac{\partial U}{\partial t} \tag{2.7}$$

$$\tau_p = {d_p}^2\frac{\rho_p - \rho_f}{18\mu} \tag{2.8}$$

The second parameter used to judge the feasibility of a tracer particle is its scattering properties. For a tracer to be captured by the cameras, it should be able to effectively scatter the limited light from the laser in the direction of the cameras and this in turn affects the final image intensity and contrast with the surroundings of a particle in an image. Intuitively, it makes sense that a larger particle would be able to scatter more light, a fact that is in direct opposition to the tracer fidelity requirement as shown in section 2.3.2. Melling [32] provides a comprehensive study on the scattering properties of tracers. One of the major challenges in PIV was to balance the scattering requirements with the loss in tracing fidelity by increasing the diameter of the particle. For PIV to move on to larger volumes, it was apparent that a large particle diameter with as close to neutral buoyancy as possible would be the best fit. This led to the selection of Helium Filled Soap Bubbles(HFSB) as the best choice for moving towards large scale volumetric measurements in air.

HFSB particles as the name suggests are particles of soap that are filled with a mixture of air and helium in proportions that make them neutrally buoyant. These particles have diameters between 200 and 400 μm which is hundreds of times as large as the conventional fog particles used in PIV and have a time response of around 10 μs (Scarano et al. [41]) making them quite suitable for quantitative studies in air flows. Due to their large size, they provide ample scattering even at low light intensities , thus allowing the laser light to be spread over a larger area thereby increasing the measurement volume.

Particle tracking algorithms

With HFSB providing the impetus to increase the measurement volumes in experiments, computational power required by algorithms like MART still proved too prohibitive to extend the range of measurable volumes. As previously explained, MART uses a voxel based approach to reconstruct the position of particles in 3D space. The other approach is the reconstruction of individual particle positions through triangulation of the lines of sight from different cameras which fall under the category of particle based approaches to reconstruction. Even though particle based tracking approaches were present since the beginnings of PIV, their main disadvantage was that the reliability of tracking was compromised with increasing particle image concentrations and this limited particle concentrations to a measly 0.001 ppp (Wieneke ([58])). However, this changed with the introduction of Iterative Particle Reconstruction (IPR) by Wieneke [58], an iterative particle tracking algorithm that allows particle image concentrations of up to 0.05 ppp, which is what is obtainable with classical tomographic PIV using MART and cross correlation algorithms.

In IPR, the particle positions obtained from triangulation are reprojected back on to the 2D images using a predefined Optical Transfer Function (OTF) (Schanz et al. [42]). The projected images are then subtracted from the original images to produce residual images which are then used to find corrected particle positions. Once the particle positions are corrected, those with intensity below a selected threshold are deleted from the volume. New particles can be found from the residual images and added. This whole process can be repeated iteratively for better accuracy. Figure 2.15 shows the flowchart of the process. The key feature of IPR is its ability to sustain high image concentrations while still being accurate, making the reconstruction of particle position computationally much cheaper.

Shake the box [4] as introduced by Schanz et al. [43] is a Lagrangian particle tracking algorithm for obtaining time resolved 3D PTV data. The algorithm takes IPR one step further by using temporal information to predict future states of the flow. STB is capable of handling image density concentrations of the same order as tomographic PIV. The working principle is briefly explained as follows. The computation of Lagrangian tracking with STB involves three phases of the algorithm: Initialisation, convergence and converged phase. In the initialization step, no temporal information is available and particles are reconstructed using algorithms similar to IPR. This is done for about 4 time steps in order to facilitate the creation of a particle “track". This track is used to extrapolate the position and predict the particle in the next time step. The predicted images are then compared to the original and a process referred to as “shaking" is performed to minimize the residuals. This is the convergence step. The residual image after shaking contains untracked particles which are likely the new particles that have entered the domain. These are again tracked using the same process. The benefit of the prediction and shaking procedure is that the information density in the residual images is significantly lower as in the raw particle images. A schematic of the STB process is shown in figure 2.16.

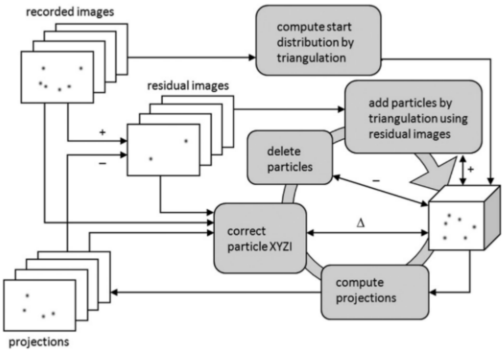


Figure 2.15: Schematic of IPR. Reproduced from [58].

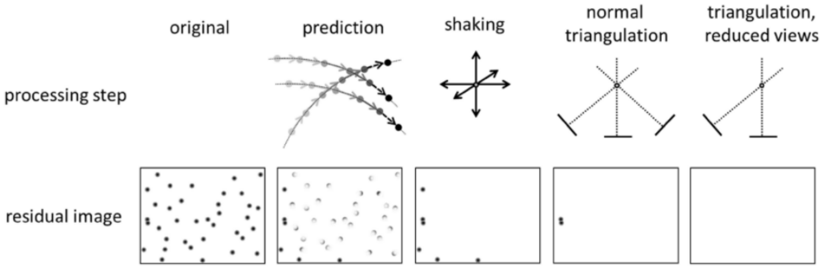


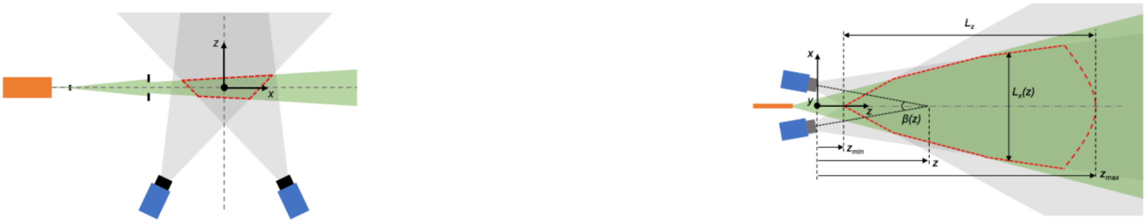
Figure 2.16: Schematic of the STB process for a single time-step in the convergence state. Reproduced from [43].

[4] Hereafter referred to as STB

2.3.3. Robotic Volumetric PIV

While the introduction of HFSB and new algorithms like STB allowed volumetric PIV to expand to new limits, there were still problems of optical access and complexity of setup involved. This prohibited large scale use of tomographic PIV in industries where planar and stereo PIV still constituted a majority of used the experimental techniques (Jux [28]). The use of a Coaxial Volumetric Velocimetry [5] probe mounted on a robotic arm as introduced by Schneiders et al. [44] opened up the possibility of obtaining *time-averaged* velocity fields in a domain as large as can be reached by the robot by ensemble averaging different overlapping sub volumes.

The CVV system consists of four cameras mounted together with low tomographic apertures and large focal lengths along with an optical fibre which provides the illumination from along the cameras' viewing direction in an expanding beam. This configuration allows the system of cameras and laser to be moved simultaneously without the need for continuous recalibration as their positions are fixed with respect to each other. This configuration is quite different to the standard tomographic system where the viewing directions are generally perpendicular to the laser axis. Figure 2.17 shows the difference between both systems. As a result of this system, it is possible to mount the setup on a robotic arm thereby ensuring an increased measurement volume by dividing the entire domain into smaller subvolumes and then later stitching the subvolumes together as shown in figure 2.18 Furthermore, the robotic system can negate problems of optical access. A proof of the versatility of the system is provided in Jux et al. [29] where the time averaged flowfield around a full scale cyclist is mapped.



(a) Conventional tomographic setup.

(b) CVV setup.

Figure 2.17: Difference between a conventional tomo setup and the CVV. The cameras are represented in blue, the laser in orange and the red dotted line represents the measurement volume. Reproduced from [44].

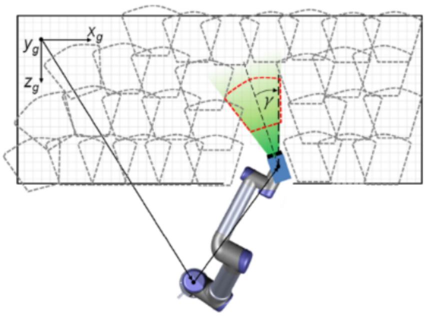


Figure 2.18: Full volume divided into subvolumes. Reproduced from [29].

The CVV system however has its own disadvantages. Due to the low tomographic aperture, errors in capturing the exact position in the depth direction are increased. Particles which are originally spherical in shape are now viewed as elongated diamonds. The smaller the aperture angle, the more pronounced this effect.

[5] Hereafter referred to as CVV

2.4. Flow around car side mirror

The flow around a car side mirror is chosen as the experimental test case due to the complex three dimensional flow that develops in its wake. De Villiers [16] in his PhD thesis describes the flow around a car side mirror model. The car mirror is mounted on a flat plate as shown in figure 2.19 and simulated using a Large Eddy Simulation at a Reynolds number based on chord diameter of $5X10^5$. The author notes that at those flow conditions and a time averaged sense, there is a massive separation and recirculation region directly behind the mirror and a reattachment point of approximately 2.25 diameters downstream of the model. Figure 2.20 shows the velocity contours and the streamlines of the flow around the object. Due to the mirror being placed on a flat plate, a distinct horseshoe vortex is formed around the model visualized in figure 2.21 as the second invariant of the velocity gradient $Q_{\nabla \bar{U}} = 5X10^5$.

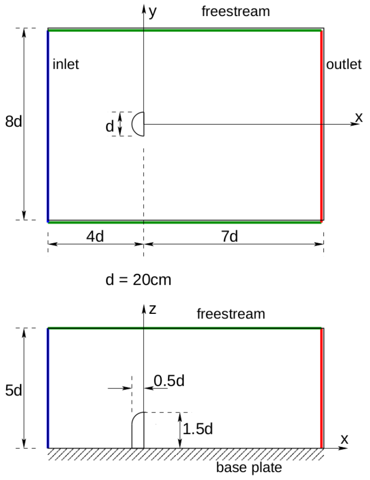
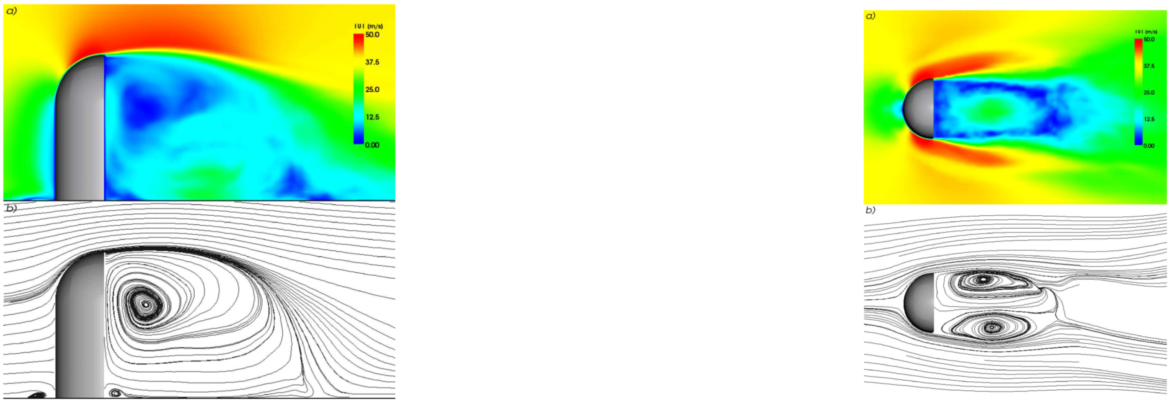


Figure 2.19: Top and side view of computational domain around car side mirror used in [16].



(a) Side View.

(b) Top View.

Figure 2.20: Time averaged velocity contours and streamlines of flow around car side mirror. Reproduced from [16].

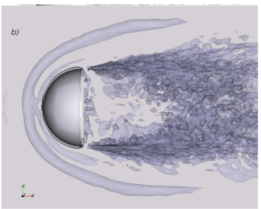


Figure 2.21: Horseshoe vortex wrapped around the model.

2.5. Research Questions

From the literature study it is clear that the state observer based DA technique for fluid mechanics is still in its nascent stage. Furthermore, the availability of techniques like robotic volumetric PIV enable capturing the complete time averaged flowfield around objects with relative ease, making their integration into the state observer technique feasible. The research questions thus are defined to help integrate volumetric PIV and the DA technique and also to provide a structure to the process. The main research question can be stated as follows:

Can a state observer based data assimilation method for steady RANS based on robotic volumetric PIV data be used to improve computational efficiency and decrease errors in CFD data for 3-D flow around a bluff body?

The main research question can be broken down into different questions that will be answered throughout the course of this thesis. Two main divisions of sub questions can be generated from the main question. These are related to the *assimilation* of experimental data and the *performance* of the technique.

2.5.1. Assimilation

1. How will the data from PTV be included in a RANS simulation? More specifically, how will the feedback term be mathematically constructed?

2. What are the different ways to assimilate data into a CFD simulation? More specifically, at what points will the forcing/feedback terms be applied and how strong will they be?

3. Can velocity fields obtained directly from PTV without the use of a solenoidal filter be integrated into a CFD simulation without causing divergence?

2.5.2. Performance

1. Can DA faithfully reproduce the main flow topological features around the car side mirror model?

2. Can DA decrease the error in CFD simulations?

    - Does the error (defined as the rms of error for velocity at all points between CFD and experimental data) decrease upon the application of DA?
    - What is the minimum achievable error in terms of percentage, with 0% implying CFD data completely replicates PTV data?

3. How does applying DA affect the convergence characteristics of a CFD simulation?

2.6. Research Objectives

The main research objective as given in chapter 1 is :

Development and assessment of a state-observer based data assimilation method for steady Reynolds averaged Navier-Stokes simulation using a complex 3-dimensional flow around a bluff body as a test case.

A successful implementation of this objective could prove as a launching pad to new techniques where data assimilation improves the quality of data that can be achieved independently from either CFD or a physical experiment. To meet the main objective, it is broken down in to sub objectives as follows:

1. Detailed literature study on the state observer based DA technique and PIV.

2. Design and perform experiment of flow around car side mirror model using robotic volumetric PIV.

3. Design the construction of the state observer based DA technique and its implementation in OpenFOAM

4. Test the technique on the 1D viscous Burgers equation as well as a 2D CFD case.

5. Implement DA for the flow around the side mirror

3
Principles of the state observer technique

In this chapter, the fundamental equation describing the operation of the state observer used in this thesis is provided in section 3.2. Further, different implementation strategies of the forcing term along with its use in the CFD solver are discussed in section 3.3. Section 3.6 briefly discusses turbulence modeling in RANS. This chapter also defines key terminology that will continue to be used throughout the course of this thesis.

3.1. Basics of the state observer

The state observer is an element of control theory (Ellis[18]). In the basic form, it is used when only partial measurements of an actual physical system are available and the function of the observer is to reconstruct the full state of the system through a model by incorporating the measurements in a way in which they guide the model to the physical solution. Consider a physical system that is represented by the equation 3.1 where x is the state variable, n is the discrete time , u is the input to the system and y is the outputs of the system. This equation simply states that the future states of a system depend on its present state and the input conditions. Here A,B and C represent system coefficients.

$$\begin{aligned} x_{n+1} &= Ax_n + Bu_n \\ y &= Cx_n \end{aligned} \tag{3.1}$$

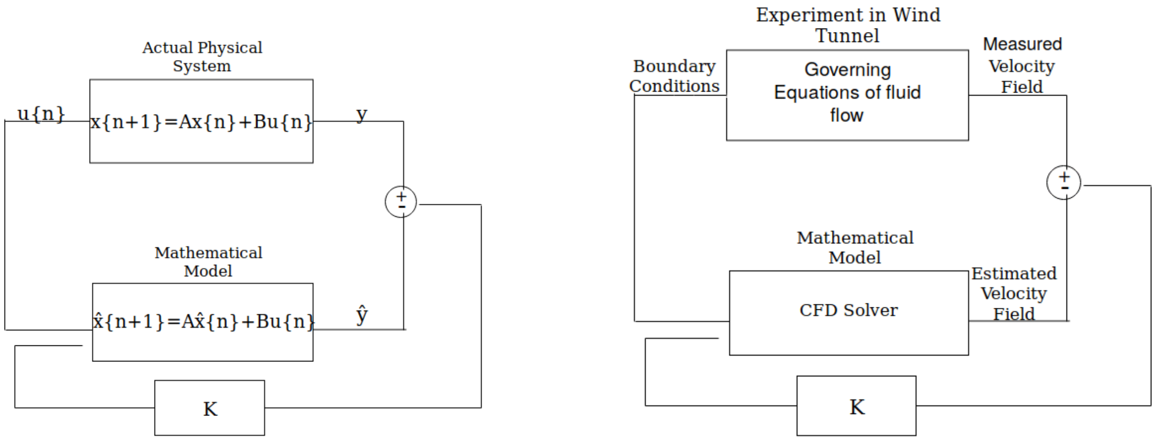
However, mathematical models of a system generally contain approximations and the equations of the system can be modified to accommodate for the estimated variables. The equations representing a model of the system are shown in equation 3.2 where the hat over the variables means that they are the estimated values from the model and not the true physical values.

$$\begin{aligned} \hat{x}_{n+1} &= A\hat{x}_n + Bu_n \\ \hat{y} &= C\hat{x}_n \end{aligned} \tag{3.2}$$

The idea behind a state observer is that by *observing* the outputs of a physical system, they can be incorporated into the model in order to guide the model to the physical state of the system. Typically, the outputs generated by the model $\hat{y}$ are different from the measured outputs y and this difference is what is incorporated into the model. Mathematically this is expressed through equation 3.3 and can be seen visually in figure 3.1a. The L in equation 3.3 is the distribution matrix for incorporating the measurements(which are typically sparse) into the model. This matrix also usually includes a gain which controls the rate of convergence of the system.

$$\hat{x}_{n+1} = A\hat{x}_n + Bu_n + L(\hat{y} - y) \tag{3.3}$$

Specifically for application of the technique in this thesis, there are some points to be noted. First, the objective is to perform the DA on a steady RANS solver, meaning that the physical measurement is to be taken from an experiment and the measurements are averaged in time. However, most steady state CFD solvers perform their calculation in iterations or steps, meaning the mathematical construction is similar to the general formulation as shown in equation 3.3. In this work, the mathematical model is represented by the Navier Stokes equations for fluid flow, the input conditions can be interpreted as the boundary conditions and the output from a physical PIV experiment is the velocity field at discrete locations. The flowchart roughly depicting the process used for the present work is given in figure 3.1b



(a) General state observer.

(b) Flowchart for present work.

Figure 3.1: Basic principle of state observer.

3.2. Mathematical formulation

The constitutive equations for fluid flow are the Navier Stokes equations. In the present study as well, the Navier Stokes equations are used as the mathematical model, albeit with a few assumptions. Firstly, due to the low speeds [1], the flow is considered incompressible. Also, no thermal effects are considered meaning the energy equation is left out of the model. Finally, all the time derivatives are neglected under the assumption of steady flow. The Navier Stokes equations in general are considered to include both the momentum and the mass conservation equations. However, the dynamics of the equation are contained in the momentum equation and the mass conservation serves as a mathematical constraint that is to be satisfied. Thus, when constructing the feedback term, it is placed in the momentum equation. The feedback term is modeled using the experimental velocity (also referred to as the ground truth) u^*. The final form of the mathematical model is given in equation 3.4 which is the vector form of the steady incompressible momentum equation with an extra feedback term. The feedback term is highlighted with a box.[2]

$$u\cdot\nabla u = -\nabla\frac{p}{\rho} + \nu\nabla^2 u + \boxed{K\frac{(u^*-u)(|u^*-u|)}{l}} \tag{3.4}$$

Here, u is the velocity , p is the static pressure, ρ is the density,l is a length scale and K is the feedback gain. The feedback term in the equation represents an acceleration/deceleration. If the term is positive, the fluid accelerates and vice versa. The reasons for choosing this term and its implications are discussed further. It must be noted that in this thesis, all available velocity components are used for forcing and the effect of forcing only partial components is not studied.

3.2.1. Choice of feedback term

In theory, the feedback term can be modeled using two parameters in an incompressible fluid flow, namely the pressure or the velocity. Previous studies have used both the terms as seen in figure 2.12, but no conclusive evidence has been provided as to why one may be more beneficial than the other. Velocity is the primary variable obtained from a PIV experiment and even though it is possible to extract pressure, it is derived through a further set of approximations thereby increasingly a layer of uncertainty. Thus velocity obtained from the experiment is considered as the *ground truth* and also used in the feedback term.

The construction of the feedback term consists of three terms, namely the error term in the numerator,the gain K and the length scale l.

Error term

Multiple variations of the feedback term can be constructed, as long as it includes the observations and guides the model to the observations and also satisfies dimensional constraints(the feedback term should be an

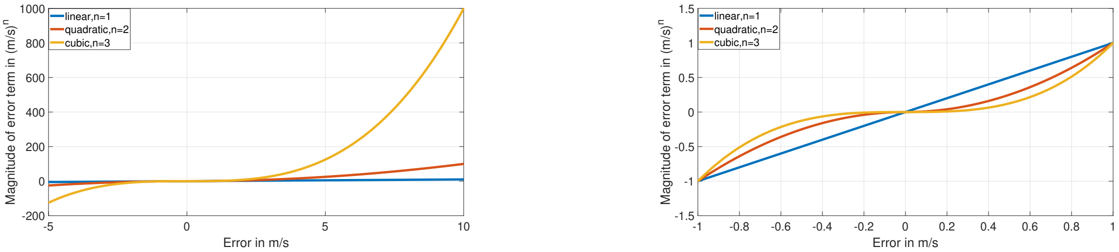
[1] The experiment is conducted at 12 m/s

[2] Here, feedback is used in the context of solving this equation in iterations in CFD, where this term then acts like a feedback term for subsequent iterations

acceleration with the units m/s^2). The error term can be modeled linearly, quadratically or even of higher orders as long as it is able to switch signs when the error switches sign. The dimensionality is maintained by using an appropriate scaling term, here referred to as the length scale(although it is not restricted to only length). In general, it can be represented as:

$$|(u^* - u)^{n-1}|(u^* - u)$$

Here, n is the order.The choice of the error term significantly affects the dynamics of the system. Consider a simple example to show the impact of each choice. Since the final experiment is a flow around a car side mirror assume the freestream velocity is 10 m/s, and due to the complex nature of the flow the dispersion of velocity can be assumed to go anywhere between -5 m/s in the wake to slightly greater than the freestream value. Now, in a CFD simulation, the solution can be initialized by assuming a constant velocity in the entire domain and the velocity at each point is updated at successive iterations. The difference in velocity at each control volume or the error as defined by the feedback term can also vary between -5 to 10 (assuming a 0 initial velocity). Figure 3.2 shows the impact of the order of the polynomial on the error term. For velocities outside (-1 1), the cubic term “explodes". The quadratic term is still much larger than the linear term implying that any regions where there is a large error has a stronger feedback than a linear term.



(a) Full range.

(b) Zoomed in.

Figure 3.2: Impact of the error term.

Feedback gain

The feedback gain can be implemented in various ways. The Kalman filter is an example where the feedback gain is made to vary both spatially and temporally based on the solution (Suzuki [49]). However, for the present study only a constant feedback gain both spatially and temporally has been employed. The value of the gain is however varied and its impact is quantified.

Length scale

The length scale is another parameter that can be subjected to arbitrary definition. The length scale is used to satisfy dimensional constraints in a case where the quadratic error term is considered. For higher orders, the length scale can be modified as a reference velocity times a reference length $u^{n-1}l$. If a linear error term is used, then a time scale is required. The length scale can be chosen based on local scaling like the $\frac{volume}{area}$ of a cell or on a global scale. The former would mean the forcing term again changes based on the local size of the cell making it harder to isolate the actual contributing mechanism to the feedback control. Thus, a global constant length scale of 1 is chosen to simplify the analysis.

3.3. Implementation

The objective of the present work is to run the state observer method for a steady RANS simulation and thus a choice had to be made on the solver. There are many CFD solvers available to choose from but for this thesis it was decided to use OpenFOAM v1706 (Open-source Field Operation And Manipulation)[3]. OpenFOAM is a collection of tools for computational continuum mechanics. It is not only limited to CFD but has capabilities to extend to problems such as but not limited to combustion, solid mechanics, electromagnetics and even finance. It is written in C++, and its code is publicly available on GitHub, under the GNU General Public License v3.0. OpenFOAM despite having a steep learning curve compared to other black box commercial CFD

[3] https://www.openfoam.com/

software, offers the flexibility to essentially create your own solver along with complete control on the solution methods. Furthermore, due to the multiple levels of abstraction offered, the user does not have to bother with the inner workings of each solver, and instead simply can build the high level commands for the solver. It is for this reason that OpenFOAM was chosen as the most appropriate software for the purpose of this thesis.

Running an OpenFOAM simulation puts the user directly or indirectly in touch with different parts of the software. These include the *executable applications* (solvers, preprocessing and post processing utilities), the *configuration system* (dictionary files) , the *boundary conditions* and the *numerical operations* (discretization schemes). A detailed review of the different parts of the software and programming options is beyond the scope of the present work and the reader is encouraged to refer to Maric et al. [31]. The executable applications further consist of solver, test and utilities. For this thesis, the solver used is the steady incompressible Semi-Implicit Method for Pressure Linked Equations (SIMPLE) solver. SIMPLE is an algorithm used for solving the coupled pressure and momentum equations in an iterative way and was introduced by Patankar [34].

A solver consists of a code that contains the main() function and possibly some subsidiary header files required by that particular solver. One of the header files is used to create the fields needed for the solver. For the present work, the code has been modified as follows. First, three new vector field variables are created:

- U^* which contains the ground truth velocity values (u^*) interpolated onto the mesh(CFD grid).
- E which is the error term $(u^* - u)|(u^* - u)|$.
- $EVol$ which is the final dimensionally correct feedback term that is added to the equations. $K\frac{(u^*-u)(|u^*-u|)}{l}$

$EVol$ is calculated using E and U^* as will be shown in section 3.4 and is inserted into the final discretized equation as a forcing term. The code containing the modifications is provided in Appendix A.

3.4. Forcing strategies

OpenFOAM uses a finite volume approach of discretization in order to solve the Navier Stokes equations. In the finite volume approach, the entire fluid domain is divided into *control volumes* and the equations are integrated over each control volume. One of the research questions (section 2.5) of this assignment is to evaluate what the best forcing strategy is. In other words, how will the forcing or feedback term be applied. Since the domain is split into control volumes, this provides the flexibility of choosing the control volumes where forcing is applied. To obtain a clear picture of the effect of forcing at different control volumes, two main parameters are varied and studied. These are:

- The total number of control volumes that are forced in comparison to the total mesh size (N)
- Feedback gain (K)

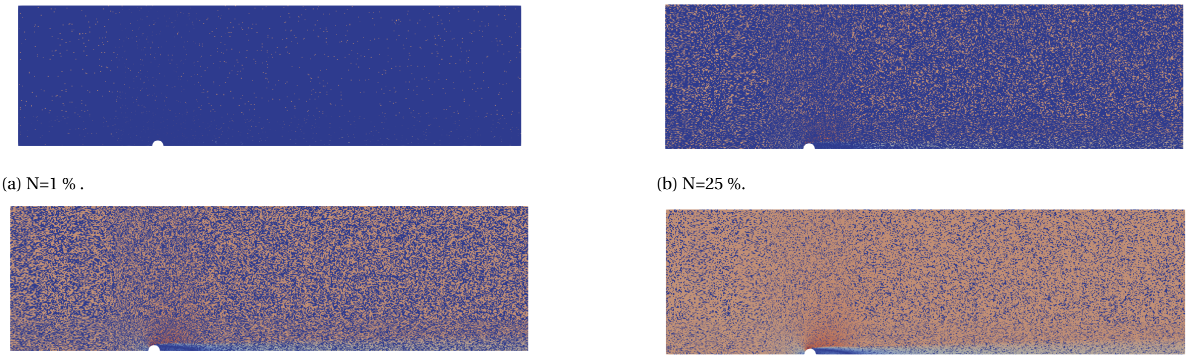
Number of control volumes (N)

The selection of control volumes to apply forcing has to be done with care due to the many permutations that exist. Instinctively, it would make sense to apply forcing wherever there is information available (ground truth velocity). However, studying the effect of forcing at a limited number of control volumes offers the opportunity to understand if forcing at only a few points can steer the simulation towards the ground truth even in control volumes where forcing is not applied. This is very important as it can possibly answer the question of what is the minimum number of control volumes or percentage of control volumes that need to be forced for the solution to approach the ground truth at all points faithfully? From an experimental point of view, it could mean that acquiring data in a limited domain or only at a few locations is sufficient. Throughout the remainder of this thesis, this parameter will be denoted by N, where N is the percentage of points in the domain that are forced.

Feedback gain (K)

The strength of the feedback term as discussed in section 3.2.1 depends on the error term, the length scale and the feedback gain. Of all the three terms, the feedback gain is the term that is chosen as the control parameter and is varied to study its effect. The value of the feedback gain is stored in a dictionary file as a variable that can be changed easily.

To this end, two main strategies of forcing are applied. First, forcing is applied to a fixed percentage of points where information is available. The points are chosen randomly to ensure there is no bias based on the location of forcing (in a wake, upstream/ downstream of an object). Figure 3.3 shows different percentage of forcing points applied on the 2D half cylinder case. The blue regions show where forcing is *not* applied. As the percentage of forcing increases, the ground truth solution of the flow across the half cylinder comes into sharper focus. For this, first, the field u^* is created based on the percentage of forcing points required in MATLAB. To achieve this, the ground truth is interpolated using the MATLAB function *griddata*(using linear interpolation) only on the randomly selected points (centers of the control volumes) and the rest of the points are assigned a 0 value. In the solver code, the error term E is calculated at all points and a conditional statement is written where only the cells with a non zero u^* are assigned an $EVol$ and the remaining are assigned 0. Finally, this field is inserted into the momentum equation. The second forcing strategy is to apply a *volume forcing* where all the forcing points are together and not sparsely distributed. This is shown in figure 3.4.


(a) N=1 % .

(b) N=25 %.

(c) N=50 % .

(d) N=75 %

Figure 3.3: Random forcing of points where N is the percentage of forced points from available ground truth data.

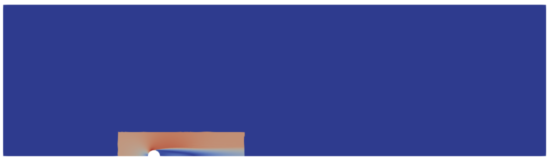


Figure 3.4: Volume forcing around half cylinder.

3.5. Physical significance of distributed forcing

Forcing at randomly distributed points or even volume forcing at only specific locations is counter-intuitive to the notion of a forcing term. Generally, a forcing term like gravity or an electric field affects the whole domain, thereby justifying their names as *body* forcing terms. In that sense forcing a fluid at only specific locations does not seem to draw parallels with a real world fluid flow. However, taking inspiration from the origins of DA in weather prediction, it is possible to draw some parallels for this technique to a real world flow at a global scale. At the global scale, atmospheric flow is one continuous complex flow. Due to the enormous scale of the earth, phenomena like localized heating or cooling cause differential pressure gradients at those locations leading to localized “winds". Different regions experience different forcing mechanisms but because of continuity, what happens in one region directly or indirectly affects the state of another region of the world. In other words, each region cannot exist independently of the other regions. A similar analogy to this is the *butterfly effect* which is a term that is derived from the metaphorical example of the details of a tornado (the exact time of formation, the exact path taken) being influenced by minor perturbations such as the flapping of the wings of a distant butterfly several weeks earlier. Although the butterfly effect was used to describe the sensitivity of the atmospheric equations to initial conditions and is used as a classical example in chaos theory, it draws similar analogies as to how one event can cascade into another both in time and space. A similar argument can be constructed for the present work, where even though the flow is invariant in time, the forcing applied at discrete locations not only affects those regions and their neighbours, but to a certain extent, influences the flow in the entire domain through continuity. This is also a reason for forcing at only a few regions to see if the forcing is strong enough to drive the flow in the non-forced regions toward the ground truth.

3.6. Turbulence modeling

The governing equations of fluid flow, the Navier Stokes equations are a set of non-linear, coupled partial differential equations. The conservation form of the equations for incompressible flows are:

$$\nabla \cdot u = 0 \tag{3.5}$$

$$\frac{\partial u}{\partial t} + \nabla \cdot (uu) + \frac{1}{\rho}\nabla p - \nabla \cdot (\nu \nabla u) = 0 \tag{3.6}$$

Though DNS solves these equations without modeling, it is generally too computationally expensive for purposes other than academic research, especially when dealing with high Reynolds numbers. Turbulence introduces a wide range of time and length scales and the computational power required to resolve all these scales with $Re^{2.7}$ (Reynolds [38]) in terms of the grid requirement. Thus for engineering interest, simpler solutions of the flow equation have been developed of which one is the RANS formulation. RANS equations are the Navier Stokes equations subjected to time averaging. In the derivation of RANS equations, the velocity is decomposed into a mean($\bar{u}$) and a fluctuating component(u').

$$u(x,t) = \bar{u}(x,t) + u'(x,t) \tag{3.7}$$

Here, x is the position in space and t is time. Substitution of this into equations 3.6 and 3.5 and then applying time averaging[4] produces the RANS equations.

$$\nabla \cdot \bar{u} = 0 \tag{3.8}$$

$$\frac{\partial \bar{u}}{\partial t} + \nabla \cdot (\bar{u}\bar{u}) + \frac{1}{\rho}\nabla \bar{p} - \nabla \cdot (\nu \nabla \bar{u}) = -\nabla \cdot \overline{u'u'} \tag{3.9}$$

The RANS equation, although solving for the mean flow, introduces an extra variable called the Reynolds stress term, which is the term on the right in equation 3.9. This term represents the effect of the turbulence on the mean flow and introduces what is called as the closure problem. To successfully solve the RANS equations, it is necessary to model the Reynolds stresses and various methods for this have been proposed. It has previously been mentioned that in this thesis two turbulence models are used: the $k-\omega\ SST$ and the Spallart-Allmaras models. Both these models use the Boussinesq hypothesis where the Reynolds stresses are related to the mean velocity gradients through an *eddy viscosity.* These models are also thus known as eddy viscosity models. The $k-\omega\ SST$ model solves two additional equations for the turbulent kinetic energy k and the specific turbulence dissipation rate ω to solve for the eddy viscosity field. Thus it is also part of the family of two-equation turbulence models. The Spallart-Allmaras turbulence model on the other hand is a one-equation model and solves modeled transport equation for the kinematic eddy turbulent viscosity ν_t. Due to the different equations solved, model constants and formulations, these two turbulence models can give different results in different situations (Bardina et al.[6]).

This chapter introduced the methodology and implementation of the state observer. The next chapter deals with the application of this method to a 1-D and a 2-D case along with the results.

[4] Time averaging is defined as $\bar{X} = lim_{t-\infty} \frac{1}{T}\int_{t_0}^{t_0+T} xdt$

4 Numerical assessment

Comparing the research questions (section 2.5) and the previous work done in state observer based DA (figure 2.12), it may seem as though the present work is “skipping" a few steps. All the previous work has been done for unsteady flows and this work attempts to work with time averaged flows and a steady CFD solver. Furthermore, the previous work has dealt with predominantly 2D flows or even in cases where 3D flows have been used, the ground truth has been constructed from previous CFD results. In cases where experimental data has been used, it has been subjected to further numerical treatment like the application of solenoidal filters. One of the objectives of this work is to directly apply experimental data without solenoidal filtering for a complex 3 dimensional flow around a simplified car side mirror. Before going to 3-D, the first step of this thesis is to prove that the method works for 1-D and 2-D cases before going into the final case. This chapter provides the numerical assessment of the technique by considering the 1-D viscous Burgers' equation for the 1-D assessment in section 4.1 then considering the steady 2-D flow around a half cylinder model using the RANS equations for the 2-D assessment in section 4.2.

In the 1-D case, a “non-physical" solution , which basically means a solution which is not the final solution of the Burgers' equation(but is still continuous) given a certain initial condition, is applied as the ground truth u^*. The equation is not steady, but is used to gauge how enforcing a non-physical solution, specifically, applying forcing at different points(N) and using different feedback gains(K), affects the final solution. The next step is to apply the technique to a 2-D steady case and perform the same analysis. For this case, the steady state flow around a half cylinder is used. To generate the ground truth solution, the flow around the half cylinder is first solved using the $k-\omega\ SST$ turbulence model and this solution is enforced onto a simulation using the $Spallart-Allmaras$ turbulence model.

4.1. 1D case: The viscous Burgers' equation

The Burgers' equation is a non-linear convection diffusion equation and involves a balance between time evolution, viscous diffusion and non-linearity. It was studied by Burgers [8] to gain insights into the behaviour of turbulence described by the interaction of the effects of convection and diffusion. The equation is given in 1-D as follows:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}-\nu\frac{\partial^2 u}{\partial x^2}=0 \tag{4.1}$$

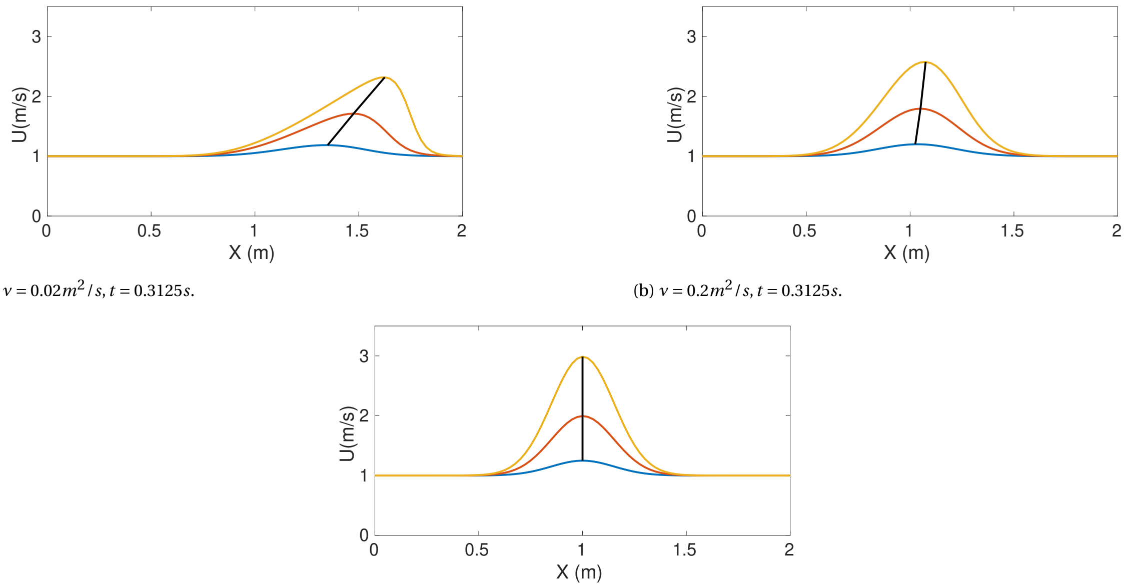
To appreciate the behaviour of the equations, a simple example is shown. Consider equation 4.1 solved in the domain $x\in[0,2]$ with the initial conditions represented by:

$$u=1+a*e^{\frac{-(x-b)^2}{2c^2}} \tag{4.2}$$

with the three variables a,b and c defined as:

- Case 1: a=0.25,b=1,c=0.15 (shown in blue)
- Case 2: a=1,b=1,c=0.15 (shown in red)
- Case 3: a=2,b=1,c=0.15 (shown in yellow)

The initial conditions are depicted in figure 4.1c. The equation is solved with Dirichlet boundary conditions at either end of the domain with u=1 m/s. Two different values of viscosity are considered- $\nu = 0.02m^2/s$ is chosen for the first trial and $\nu = 0.2m^2/s$ for the second trial. Figures 4.1a and 4.1b show the evolution of the solution after t=0.3125 s for both the trials respectively. The effect of both convection and diffusion is seen clearly in both cases. Since the initial profiles are positive velocities, all the “humps" travel to the right. An interesting effect occurs due to the non-linearity of the equations. From figure 4.1a, it can be seen that the front side of the propagating wave steepens compared to the back side. This can sometimes lead to a phenomena known as *wave breaking* (Pomeau[36]) where the top portion of the wave moves beyond the regions below it and beyond this point a continuous function cannot be maintained anymore. When the viscosity is high, as in figure 4.1b, diffusion provides a “smoothing" and this phenomena can be avoided. In the subsequent analysis, high viscosity has been used to ensure no discontinuities occur in the flow and the state observer is applied.



(a) $\nu = 0.02m^2/s, t = 0.3125s$.

(b) $\nu = 0.2m^2/s, t = 0.3125s$.

(c) Initial conditions,t=0.

Figure 4.1: Behaviour of Burger’s equation.

4.1.1. Application of state observer

The state observer for the Burgers’ equation is applied using the formulation shown in equation 4.3. This equation is solved numerically. Time is discretized using an explicit forward finite difference scheme, the non linear convection term with a backward difference in space and the diffusion term with a central difference scheme. The discretized version of the equation is shown in equation 4.4. Here, n denotes the time step and i denotes the spatial coordinate. The details of the simulations are provided in table 4.1. To understand the effect of forcing, it is applied at all the points in the domain (N=100%). The quadratic error term is chosen for analysis.

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2} + k(u-u^*)|u-u^*| \tag{4.3}$$

$$\frac{u_i^{n+1}-u_i^n}{\Delta t} = -(u_i^n)\frac{u_{i+1}^n-u_i^n}{\Delta x} + \nu\frac{u_{i+1}^n-2u_i^n+u_{i-1}^n}{\Delta x^2} + k(u_i^n-u_i^{*n})|(u_i^n-u_i^{*n})| \tag{4.4}$$

No. of points	81
Domain length	2 m
Time step	4e-4 s
Viscosity	0.3 m^2/s
No. of time steps	25000
No. of points forced	81
Feedback gain	2
Boundary Conditions	Dirichlet

Table 4.1: Details of simulation.

Ground Truth u^*

For equation 4.1, using one of the initial conditions (case 1) and in the same domain, the solution will diffuse and eventually (after infinite time) the whole domain will attain a constant steady state solution. The ground truth function that is enforced is given by equation 4.5 and is shown with the initial condition as given in the aforementioned case 1 in figure 4.2.

$$u^* = 1 + \sin \pi \frac{x}{2} \tag{4.5}$$

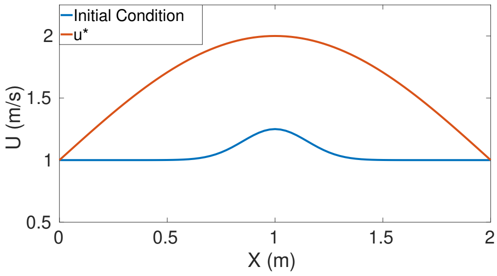
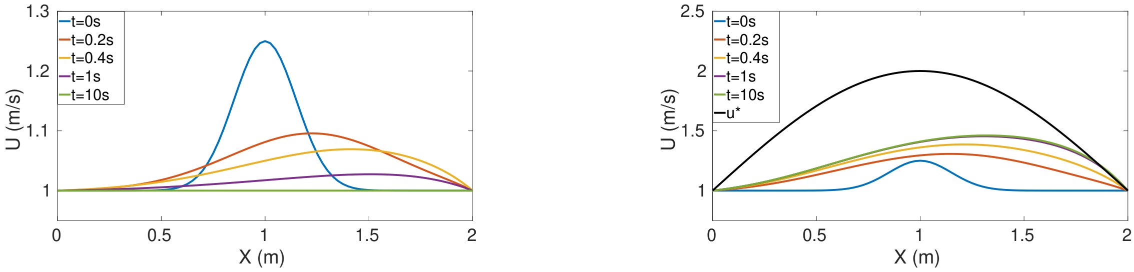


Figure 4.2: Initial Condition and non-physical forced solution.

Results

The result of the simulations without and with forcing are shown in figure 4.3a and 4.3b. When no forcing is applied, the initial solution convects and eventually diffuses leading to a steady state solution that is constant throughout the domain and equal to the velocities at the boundaries (1 m/s in this case). When forcing using the ground truth as in equation 4.5 is applied, the behaviour of the solution is significantly different. As seen in figure 4.3b, the initial "hump" appears to grow while advecting and finally obtains a steady state solution that tends to have approached the forced solution. The comparison between the final iteration of the forced solution and the forcing function is shown in figure 4.4.



(a) Without forcing.

(b) With forcing.

Figure 4.3: Time evolution of the initial solution.

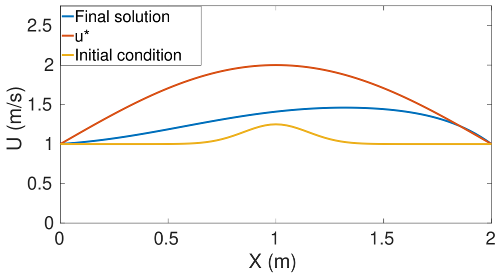


Figure 4.4: Initial and final solutions compared to the ground truth (u^*).

Role of forcing term

Referring to equation 4.3, it can be seen that the time evolution of the velocity is a balance between three terms, namely the convection, diffusion and the forcing term. The sum of these individual components contribute to the progression of the solution. Therefore analysis of these terms can provide some insights into its behaviour.

Consider two points A and B whose position is shown in figure 4.5. For both these points, the individual terms of the Burgers' equation have been plotted for every iteration for both the forced and the unforced case. The results are shown in figure 4.6. When the solution is not forced, the values of the individual terms at both locations eventually tends to 0. For point A, which is behind the hump of the initial solution, the convection term is negative since the wave moves to the right but the viscous term is positive and more dominant initially meaning that the total, which is basically the final acceleration of the point, is initially positive. However, this quickly changes as the convection term then dominates the viscous term and they both then slowly tend to 0 as the solution diffuses out of the domain. For point B, which is on the right side of the hump, both the viscous and the convection terms are initially positive and after the hump passes over it, the viscous term quickly changes sign and finally both terms taper off. Despite the initial differences between points A and B that arise purely due to their position in space, in both cases, the final values of the individual terms tends to zero.

When forcing is applied, the behaviour of the solution changes. For point A, which is initially far from the ground truth, the forcing term is large and dominates the other two terms. This trend continues throughout the simulation and at one point, the sum of all three terms approaches 0, indicating a balance. For point A, this happens when the viscous and convection term are negative, meaning that both the terms try to produce a deceleration in point A, but due to the point still being quite far away from the forcing solution, the forcing term produces an acceleration. For point B, a similar trend is observed where the forcing term dominates the other two terms, but due to this point being on the right side of the hump, the convection term is positive throughout and the forcing term has a smaller magnitude that at point A.

Clearly, applying forcing changes the dynamics of the system. There are some points worth noting regarding the behaviour of the solution when forcing is applied:

- The forcing term tries to “pull" the solution towards the forcing function and if other terms have the opposite effect, there is a point when equilibrium between all the terms will be reached and the solution attains a steady state.
- This further implies that if the forcing is “non-physical" then the final steady state cannot completely reach the forcing solution because of the tendency of the other terms to move the solution in a different direction.
- The magnitude of forcing can be altered by changing the feedback gain K and this will further dictate the relative magnitude of the forcing term with respect to the other terms.

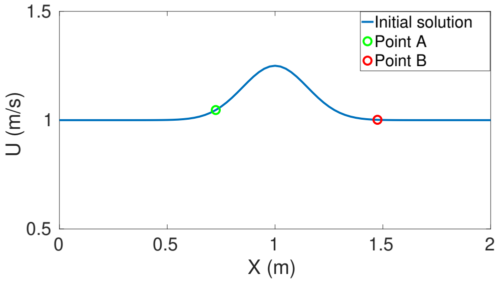
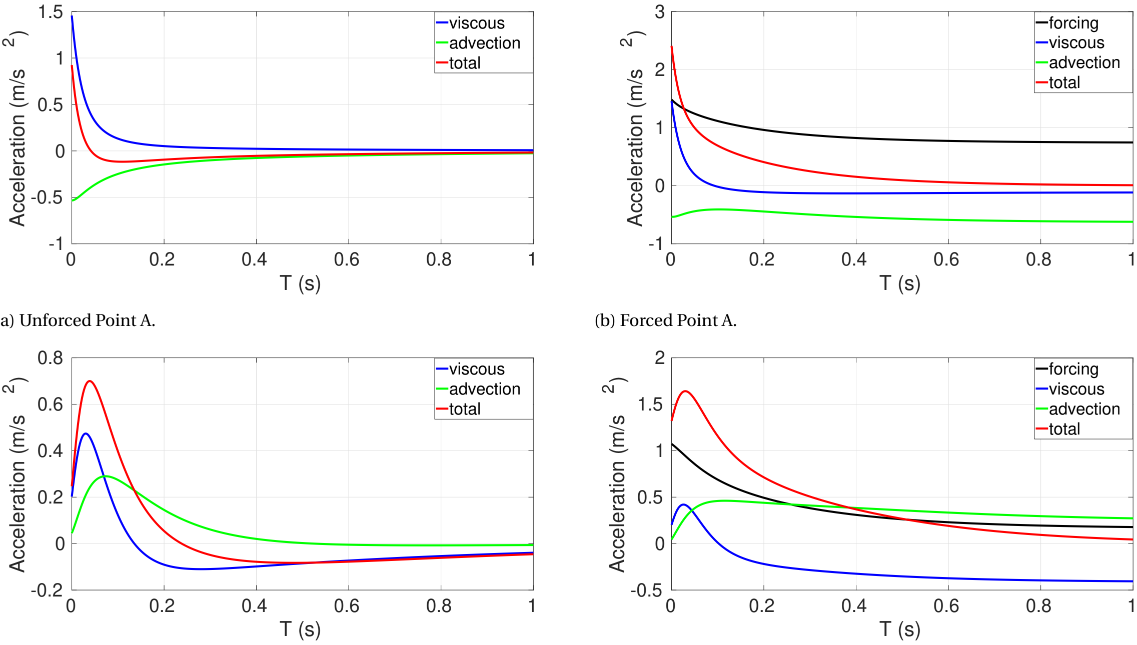


Figure 4.5: Location of points A and B.



(a) Unforced Point A.

(b) Forced Point A.

(c) Unforced Point B.

(d) Forced Point B.

Figure 4.6: Location of points A and B and behaviour of solution at those points.

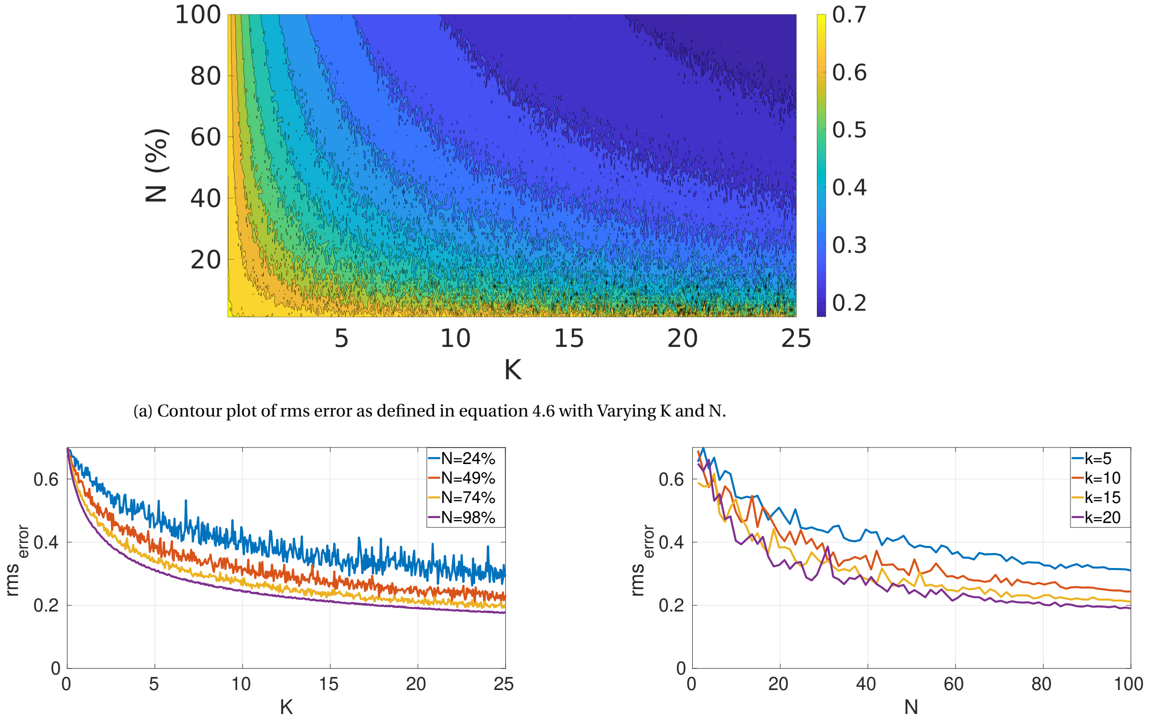
4.1.2. Random forcing for the Burgers equation

The previous section dealt with forcing at all the points of the domain. However an objective of this thesis is to quantify the effect of distributed forcing. To help meet this objective, the same simulation is run but this time with different number of forcing points. N denotes the percentage of points that are forced. At the same time, the gain K is also varied and the effect of both these parameters is quantified through the rms error e_{rms} which is defined as in equation 4.6. This definition ensures that the value of error remains dimensionless. The rms error is calculated after the final iteration for all simulations (u_{final}).

$$e_{rms} = rms\left(\frac{u_{final} - u^*}{u_{ref}}\right) \; with \quad u_{ref} = 1m/s \tag{4.6}$$

The forcing strategy is applied as follows: Since the domain is divided into 81 points, the number of points forced is increased at every run from 1 to 81. For each run, the value of K is changed from 0.1 to 25 in 500 steps and in each of those steps, the points to be forced are selected randomly. This ensures that the effect of forcing at the randomly chosen points is repeatable and is not biased based on the spatial position of the points.

The results of the simulation are shown in figure 4.7. From the contour plot it can be seen that as both the feedback gain and the number of forcing points increase, the rms error decreases. Figures 4.7b and 4.7c shows how each parameter individually affects the rms error. From the figures, it can be seen that even when the full volume is forced and the gain keeps increasing, the rms error asymptotically approaches a fixed value which is higher than 0, implying that there will still be error in the final steady state solution even for high gains. An interesting observation is that for the random forcing, due to the asymptotic nature of both the variables, K and N, the increase in either does not correspond to a linear reduction in error. For example, for a gain of 15, doubling the number of forcing points from 24 % to 48 % results in an rms error reduction of only around 15 % whereas increasing the number of forced points from 0 to 24 % reduces the rms error by almost 43 %.



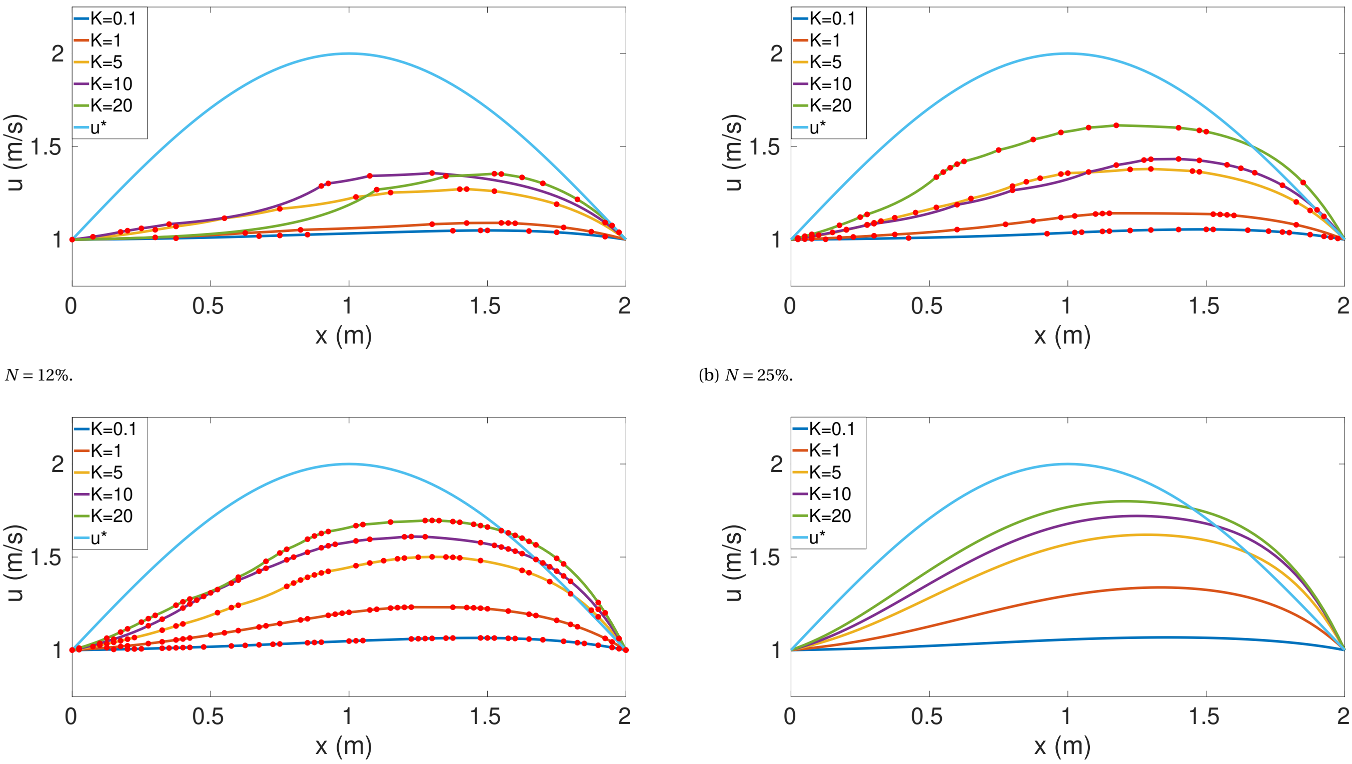
(a) Contour plot of rms error as defined in equation 4.6 with Varying K and N.

(b) rms error vs N.

(c) rms error vs K.

Figure 4.7: RMS error for varying values of N and K.

Figure 4.8 shows a few examples of the effect of forcing on the final solution. The red dots show the points where the solution was forced (not shown for N=100 %). Since the points were chosen randomly, they vary between different runs. An interesting observation from the figures is how the function becomes locally non-smooth wherever forcing is applied. Since an explicit scheme is used in time for this analysis, the forcing affects only the cell where it is enforced at any particular iteration. If the forcing is not too strong, the jump in the velocity between iterations is small enough for the overall function to still remain continuous although the smoothness is compromised. The effect of forcing at a point is then spread to neighbouring points in the subsequent iteration through the effect of convection and diffusion and can be visualized easily by noting that spatial gradients take information from their neighbourhood points. If the forcing is too strong, then the function becomes discontinuous and stability issues arise. Furthermore, when N is low, the spatial position of the points significantly affects the final solution. This trend is also observed in fig 4.7, where for low values of N, increasing the K does not necessarily reduce the error.



(a) $N = 12\%$.

(b) $N = 25\%$.

(c) $N = 50\%$.

(d) $N = 100\%$.

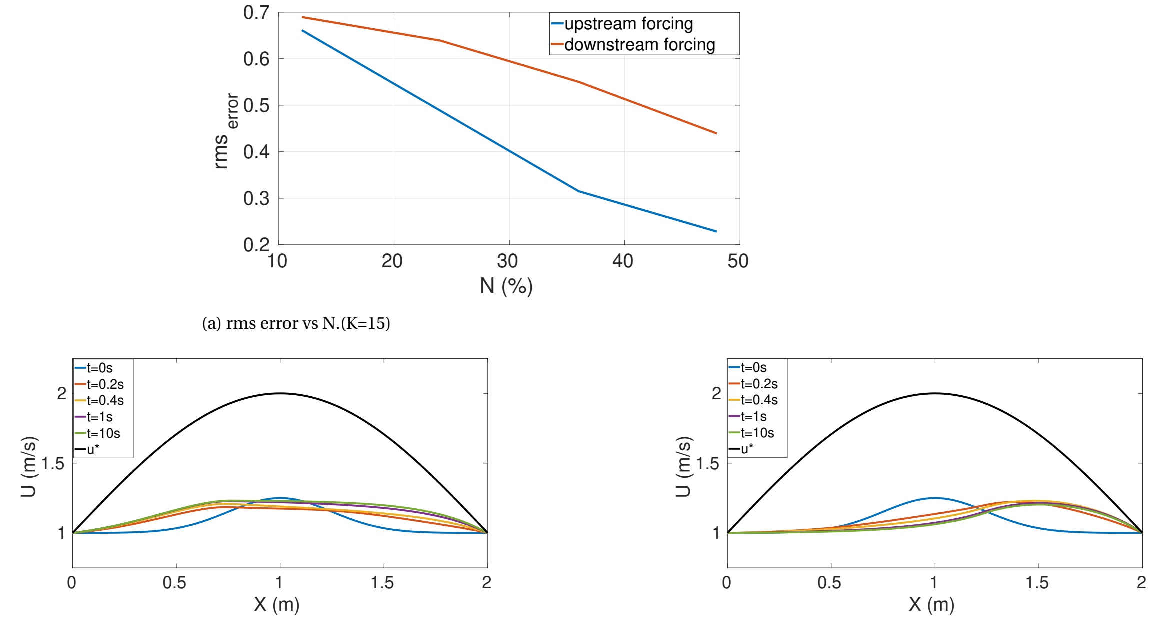
Figure 4.8: Solutions of DA applied to Burgers' equation. Red dots represent the locations of forcing.

4.1.3. Volume forcing for the Burgers' equation

Volume forcing is essentially one of the possible realizations of random forcing but with the added advantage of knowing that in the case of volume forcing the final solution can be influenced by the spatial locations of the forcing points. The exercise of choosing a location for volume forcing can also be random, however the volumes can be chosen with some scientific intuition. Due to the presence of the convection and the diffusion term, it is interesting to understand how these terms help propagate the influence of forcing at a few locations to other parts of the domain. Therefore, in the present study, the domain is divided into two regions: *upstream* and *downstream*, which represent the first half and the second half of the domain respectively. The forcing strategy is as follows: For the upstream case, the forcing points are increased in steps of 10, always starting from the first point and for the downstream case, the points are forced starting from the last point in steps of 10. The rms error of the total solution is then compared for the two cases. For the sake of simplicity, the gain K is maintained at a constant value of 15 for all the simulations. The result(rms error) of the simulations are shown in figure 4.9a.Upstream forcing for all cases produced smaller errors than downstream forcing and this can primarily be attributed to convection and can clearly be seen in figures 4.9b and 4.9c where 30 points (N=36 %) were forced for each case and the evolution of the solution is shown.

4.1.4. Convergence

Convergence for the case of the Burgers' equation is defined as the iteration number where the rms value of error as defined in equation 4.6 changes by less than 0.1 % for the subsequent iteration. The results are shown in figure 4.10. The trend shown is very similar to that seen in figure 4.7, however for low values of K and N, the convergence is erratic.

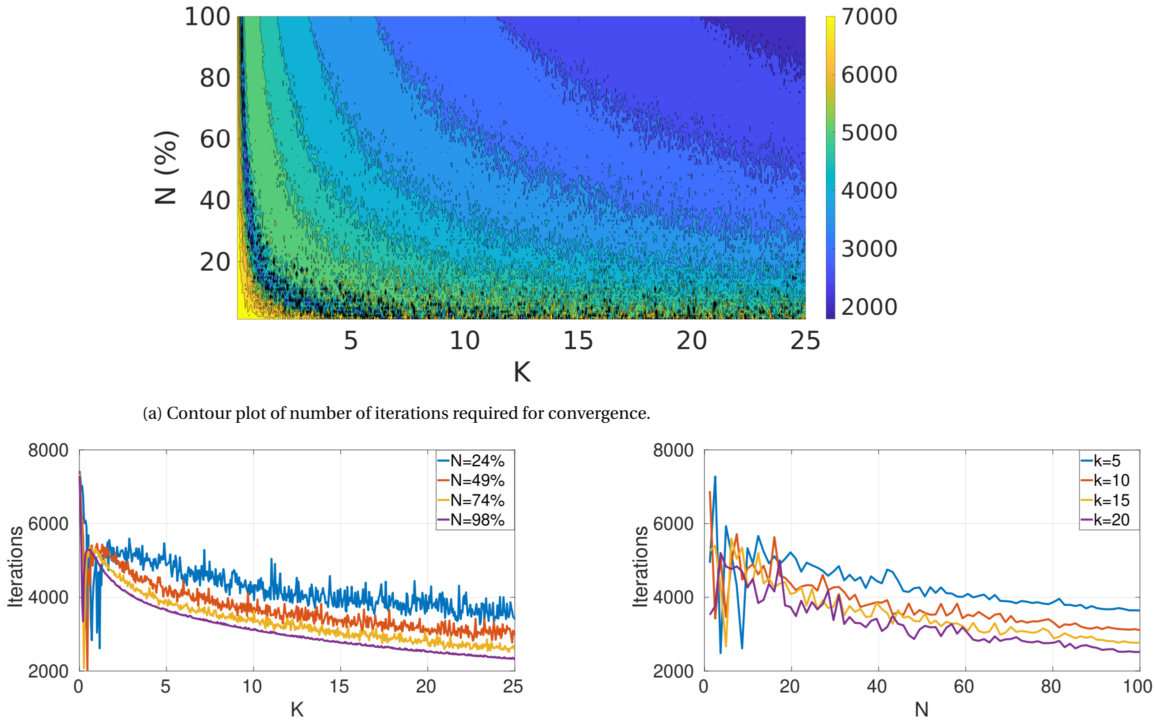


(a) rms error vs N.(K=15)

(b) Upstream forcing.(N=36 %,K=15)

(c) Downstream forcing.(N=36 %,K=15)

Figure 4.9: Difference between upstream and downstream forcing.



(a) Contour plot of number of iterations required for convergence.

(b) Iterations vs N.

(c) Iterations vs K.

Figure 4.10: Convergence plots for Burgers' equation with DA. Shown only for random forcing.

4.2. 2D case: Steady flow around a half cylinder

For application of the state observer to a 2-D flow, the steady state flow around a half cylinder is chosen. In this study, the half cylinder implies that a full cylinder is cut in half in a direction parallel to the flow, essentially giving rise to a bluff body. This geometry is chosen as opposed to a full cylinder to avoid unsteady effects such as vortex shedding. Even though the proposed technique should in theory work even when applied to cases where the flow may exhibit unsteady characteristics, this geometry is chosen to simplify the analysis. Furthermore, a half cylinder still exhibits some complex dynamics like separation and reattachment with a long wake. As mentioned earlier, all the simulations are performed using OpenFOAM v1706. In the present study, the *ground truth* is acquired from a simulation of the half cylinder using the $k-\omega SST$ model and the state observer is applied to the Spallart-Allmaras model to try and replicate the former. When solved with both these models, the solution showed quite noticeable differences in the flowfield thus providing grounds for application of the state observer. The flow is solved at a Reynolds number based on cylinder diameter of 1000.

4.2.1. Computational setup

The computational domain is illustrated in figure 4.11. Sufficient distance is provided downstream of the cylinder to prevent boundary effects from the outlet. The domain consists of 7 boundary patches: inlet, outlet, top, ground , cylinder wall and sides. Even though the solution is performed in 2-D, OpenFOAM always works with 3-D meshes by adding one cell in the third dimension, which cannot be seen in the illustration. The outlet is placed at around 30D downstream of the cylinder, and both the top and the inlet are placed around 15D away from the centre of the cylinder.

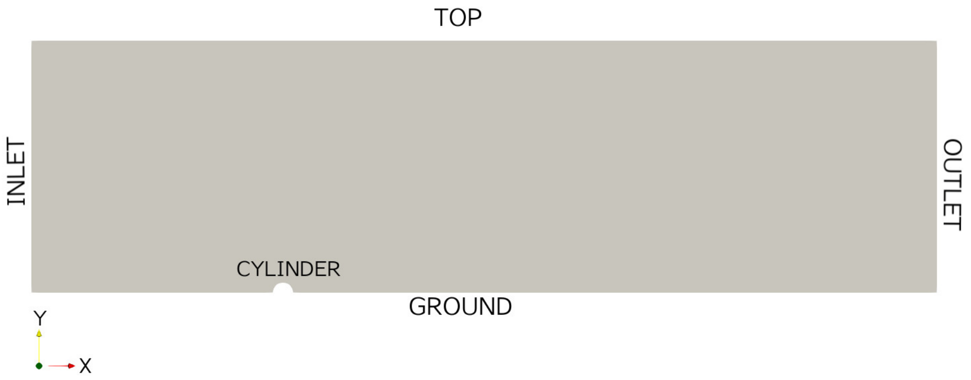


Figure 4.11: Computational domain for half cylinder.

Boundary and initial conditions

Boundary conditions are a very important part of a CFD simulation as they can influence the dynamics of the partial differential equations. Therefore specifying appropriate boundary conditions is of extreme importance. The boundary conditions used in both simulations are shown in table 4.2. The Neumann boundary condition has been used primarily at the outlet while the inlet has been assigned Dirichlet boundary conditions. The units of pressure are m^2/s^2 because due to the incompressibility assumption, pressure is divided by the density. The symmetry boundary condition acts like a wall with slip and the empty boundary condition implies that the equations are not solved in that direction. Extra boundary conditions arise for specification of turbulent properties. For all the properties, the formulas are taken from Stringer et al.[48] and Derksen [17] and are shown below:

Variable	Units	Inlet	Outlet	Top	Ground	Side	Cylinder	Initial Field
p	m^2/s^2	0	0	symmetry	symmetry	empty	$\partial p/\partial n = 0$	0
u	m/s	(1 0 0)	$\partial u/\partial n = 0$	symmetry	symmetry	empty	no-slip	(1 0 0)
ω	$1/s$	9	$\partial \omega/\partial n = 0$	symmetry	symmetry	empty	Wall function	9
k	m^2/s	0.00135	gradient=0	symmetry	symmetry	empty	1E-10	0.00135
ν_t	m^2/s	15e-5	$\partial \nu_t/\partial n = 0$	symmetry	symmetry	empty	0	15e-5
$EVol$	m/s^2	(0 0 0)	(0 0 0)	symmetry	symmetry	empty	(0 0 0)	(0 0 0)

Table 4.2: Boundary conditions for the simulations.

$$k = \frac{3}{2}(uI)^2 \tag{4.7}$$

$$\omega = \frac{\rho k}{\mu}(\frac{\mu_t}{\mu})^{-1} \tag{4.8}$$

$$k_{wall} = 0 \tag{4.9}$$

$$\omega_{wall} = \frac{6\mu}{\beta_1 y_{wall}^2} \tag{4.10}$$

Here, $\beta = 0.075$, u is the freestream velocity, I is the turbulent intensity, $\frac{\mu_t}{\mu}$ is the turbulent viscosity ratio. The turbulent intensity is chosen as 3% which corresponds to a medium-turbulence case, and the turbulent viscosity ratio has set to 15. For the turbulent kinetic energy at the wall, to avoid problems with the solver it is set to $1E-10$ instead of 0. Wall functions are used when the mesh is not completely resolved up to the wall. (More details are provided in section 4.2.1). However, in this case, the mesh is resolved up to the wall and low Re formulations for the wall functions which are available in OpenFOAM are used. Here, low Re corresponds to the flow near the wall and not the general flow.

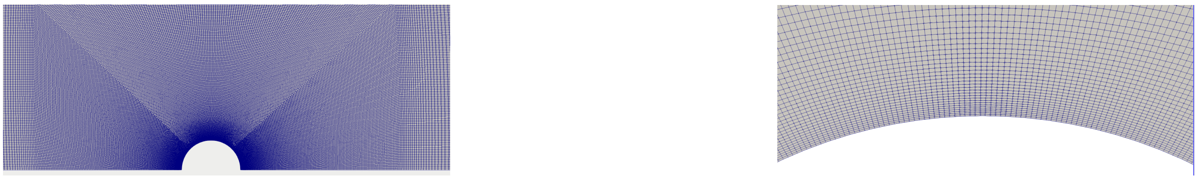
Discretization

The mesh is generated using ANSYS ICEM CFD v 16.0. Meshing is performed by dividing the domain into blocks on which a structured grid was defined. Meshing affects all subsequent steps of a CFD process and thus must be performed with care. For engineering interest, it is important to simulate what happens close to walls of objects and areas directly downstream of them. The interface between a solid object and a fluid is of utmost importance as it dictates the momentum exchange and the flow phenomena which in turn affects the flow downstream. The size of the cells near the boundary is usually dictated by the non dimensional number y^+, which is given by:

$$y^+ = \frac{y_{wall} u_\tau}{\mu} \tag{4.11}$$

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{4.12}$$

Here u_τ is the skin friction velocity. A universal boundary layer profile called the “law of the wall" dictates the flow behaviour near the wall. This profile is divided into different regions namely, the viscous sublayer, the buffer layer, the log layer and the outer layer(Bradshaw and Huang [7]). The viscous sublayer is the region where $y^+ < 5$. The buffer layer is the region $< 5y+ < 30$. To completely simulate boundary layer flow and dynamics, it is essential to resolve the boundary layer well into the viscous sublayer. However, this can prove to be too expensive in terms of computational cost due to the number of cells required to achieve this. Instead, *wall functions* are used to approximate the velocity field close to the wall and the first cell should be placed outside the buffer layer, typically in the log layer. For the half cylinder, since only a 2-D solution is required and also to avoid approximation wherever possible, the mesh is resolved with a $y^+ \approx 1$. The total number of cells in the domain is around 200000. The mesh is made fine in the wake region and all the transitions are made smooth. The maximum aspect ratio is 3.16, average non-orthogonality is 10.14 degrees and maximum skewness is 0.71. The final mesh is shown in figure 4.12.



(a) Domain divided into different blocks.

(b) Near wall mesh

Figure 4.12: Mesh for half cylinder.

Discretization schemes

The discretization schemes used for converting the partial differential equations to algebraic equations are shown in table 4.3. More information about each scheme can be found at the OpenFOAM [1] website

[1] https://cfd.direct/openfoam/user-guide/v6-fvschemes/

Term	**Scheme**
Gradient	Gauss linear
	u-Bounded Gauss linear
Divergence	k-Bounded Gauss linear limited
	ω- Bounded Gauss linear limited
Laplacian	Gauss linear corrected
Interpolation	linear
Surface normal gradients	corrected

Table 4.3: Discretization schemes.

Solver settings

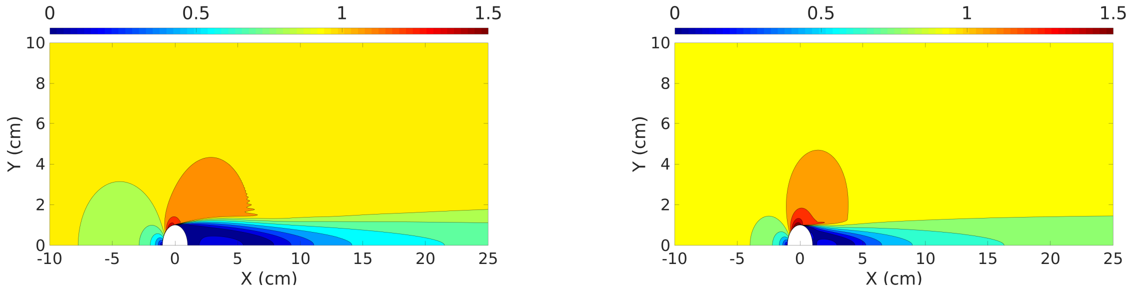
After discretization, the equations, which are now in a system of linear algebraic equations, have to be solved. These are usually solved using iterative methods. The objective of solving in iterations is to reduce the equation residuals (which in simple terms is the error in the result). The smaller the residual is, the more accurate the solution. Before solving an equation for a particular field, the initial residual is evaluated based on the current values of the field. After each solver iteration the residual is re-evaluated. There are three ways in which a solver can be stopped:

- The residual falls below a specified solver tolerance.
- The ratio of current to initial residuals falls below a relative tolerance.
- The number of iterations exceeds a set maximum number.

For the present study, all the tolerances have been specified as 1E-10, along with relaxation factors for velocity of 0.9 and 0.8 for pressure. The maximum number of iterations allowed for each solver is 5000, except for the baseline cases to generate the ground truth which were solved till convergence (tolerance of 1E-7). Since such a low tolerance is set, for all the DA simulations, the iteration limit is reached. This is done to facilitate analysis of the convergence characteristics.

4.2.2. Baseline results

To provide a means of comparing the effect of DA, simulations using the two aforementioned turbulence models are performed and their differences noted. The contours of velocity magnitude provided by each simulation is shown in figure 4.13 where the half cylinder is shown in white. From the figure, it can be seen that the two cases produce different flowfields despite the fact that they were simulated on the same mesh using same boundary and initial conditions (except for the turbulence properties). Furthermore, the $k-\omega\ SST$ simulation shows some oscillations in the velocity near the cylinder and above the wake. Although the exact reason for this could not be pinpointed, it could be a manifestation of the interaction of the turbulence model with the mesh since it did not appear with the Spallart-Allmaras model. Furthermore, the objective of applying DA was to understand if a CFD simulation can be driven to a result that is different than what it would lead to if not forced, and not too much emphasis is placed on the physical feasibility of the forcing solution. Thus, it was decided to use the solution provided by the $k-\omega\ SST$ as the ground truth. Comparing the two velocity fields, it can be seen that the most prominent difference between the solutions is the larger wake region in the $k-\omega\ SST$ model. Figure 4.14 shows the difference in the wake profiles of streamwise velocity between the two simulations at 3D downstream of the half cylinder. These differences in the velocity field are large enough to justify the application of DA and check if the resulting solution can approach the ground truth velocity.



(a) $k-\omega$ SST.

(b) Spallart-Allmaras

Figure 4.13: Contours of velocity magnitude (m/s) for the two turbulence models.

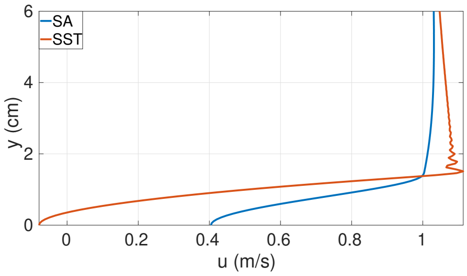


Figure 4.14: Streamwise velocity profile comparison in the wake 3D downstream of half cylinder.

4.2.3. Random forcing for half cylinder

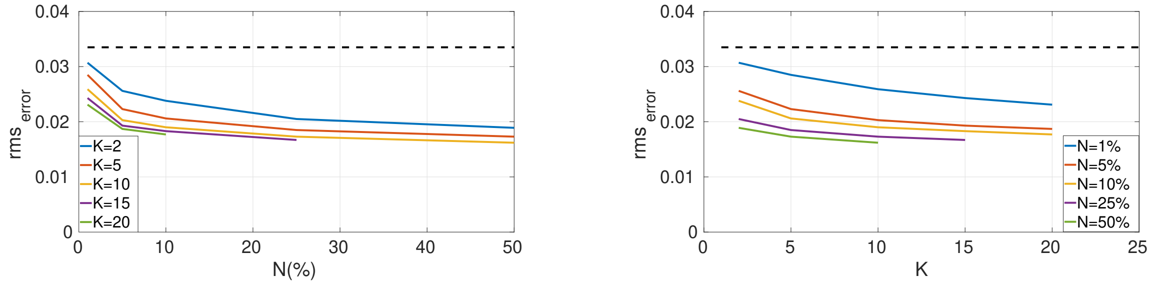
Random forcing for the half cylinder is applied using the same strategy as shown in figure 3.3. As in the case of the Burgers' equation, the two parameters N and K are varied. However, due to constraints in time, only a few values of K and N were selected. Table 4.4 shows the different runs that are conducted and whether the runs converged or not. As the percentage of points increased along with the feedback gain, some simulations started to diverge. Divergence can be caused by a lot of factors but for the DA, it is assumed that the stability of the solution was compromised due to the high magnitude of forcing.

To evaluate the effectiveness of the DA, the same measure for error as defined in equation 4.6 is used but with all the components of velocity. To ensure that the results are not biased by the spatial positions of the random points, runs with lower percentage (1,5 %) are repeated 4 times and their average is considered. Figure 4.15 shows the rms error values for the different runs. It must be noted here that the magnitude of the rms error is small when compared to that observed in the Burgers' equation analysis and this is due to the fact that in the majority of the domain, the differences in velocities are negligible and only in the wake regions is where the velocities show appreciable differences. The black dotted lines in figure 4.15 represent the rms error for an unforced simulated. That is, the rms error in velocity between the unforced Spallart-Allmaras and the $k-\omega$ SST simulations.

The trend for the half cylinder resembles the trend for the analysis with Burgers' equation. The error again asymptotically approaches a value higher than 0. Another point worth noting is that even for a small percentage of points and a low feedback gain, the error reduces compared to the baseline case. Figure 4.16 shows the contours of velocity magnitude of the flow around the half cylinder with DA compared to an unforced

$N\|K$	2	5	10	15	20
1%	✓	✓	✓	✓	✓
5%	✓	✓	✓	✓	✓
10%	✓	✓	✓	✓	✓
25%	✓	✓	✓	✓	✗
50%	✓	✓	✓	✗	✗

Table 4.4: Different runs for half cylinder DA. Tick mark indicates that the simulation converged.

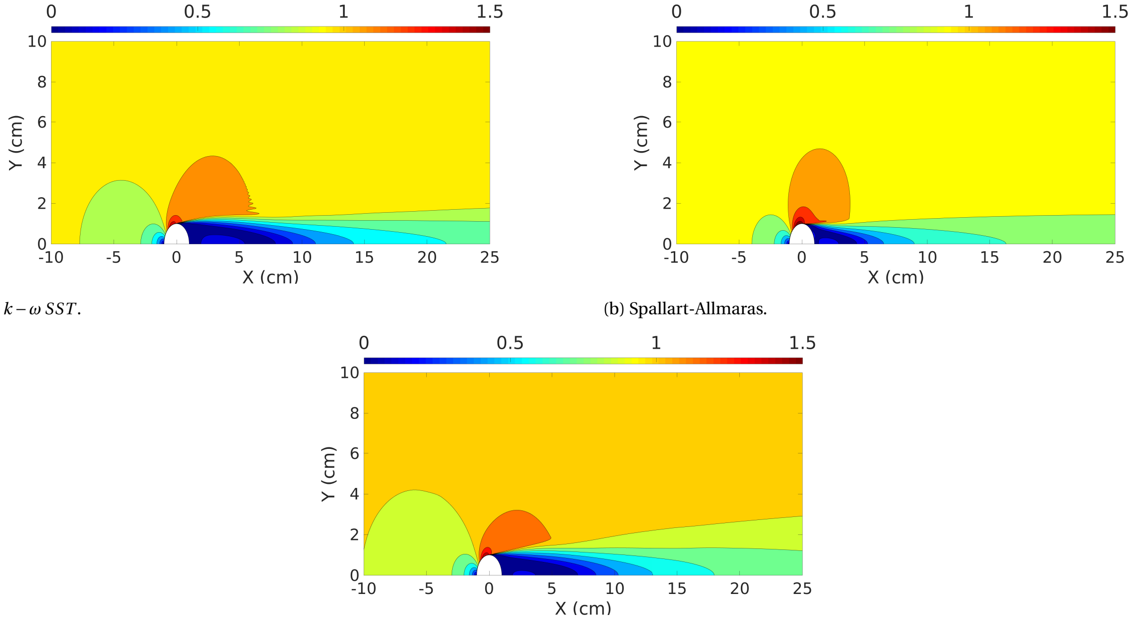


(a) rms error vs N for different K.

(b) rms error vs K for different N.

Figure 4.15: rms error for different runs with DA.

solution and the ground truth. The results are shown for the case where N=50% and K=10. Figure 4.17 shows the streamwise velocity profile comparison for both the unforced simulations and the DA assisted simulation. Clearly, the velocity approaches the ground truth velocity, and also does not share the same problems of oscillation as seen in the ground truth velocity. A point to note for the forcing in the 2D case as compared to the Burgers' equation is that in this case, there are no smoothness problems in the velocity field that occur due to forcing. This is due to the fact that in addition to solving the momentum equation which contains the



(a) $k-\omega$ SST.

(b) Spallart-Allmaras.

(c) DA (N=50%,K=10).

Figure 4.16: Contours of velocity magnitude (m/s) for original simulations and DA assisted simulation.

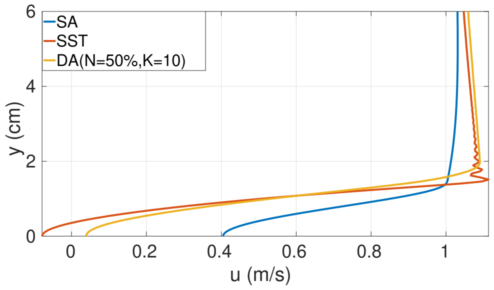
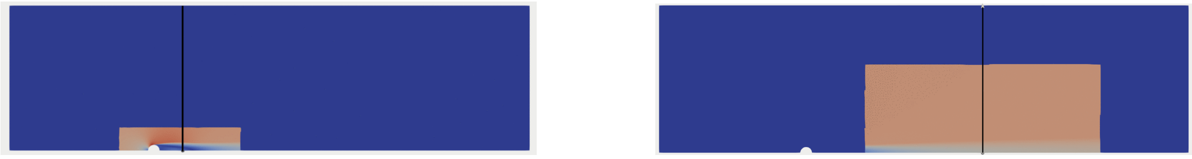


Figure 4.17: Streamwise velocity profile comparison in the wake 3D downstream of half cylinder.

forcing term, the continuity equation has to be satisfied and this is achieved by solving a pressure equation (Patankar [34]). This places a constraint on the velocity field obtained from the momentum equation and thus within the same iteration, the velocity field is recalculated in order to satisfy continuity and smoothness is maintained. Also the freestream velocity for the case with DA is slightly higher than both the baseline cases with a maximum difference of around 2% of the freestream velocity, although no particular reason for this behavior could be determined.

4.2.4. Volume forcing for half cylinder

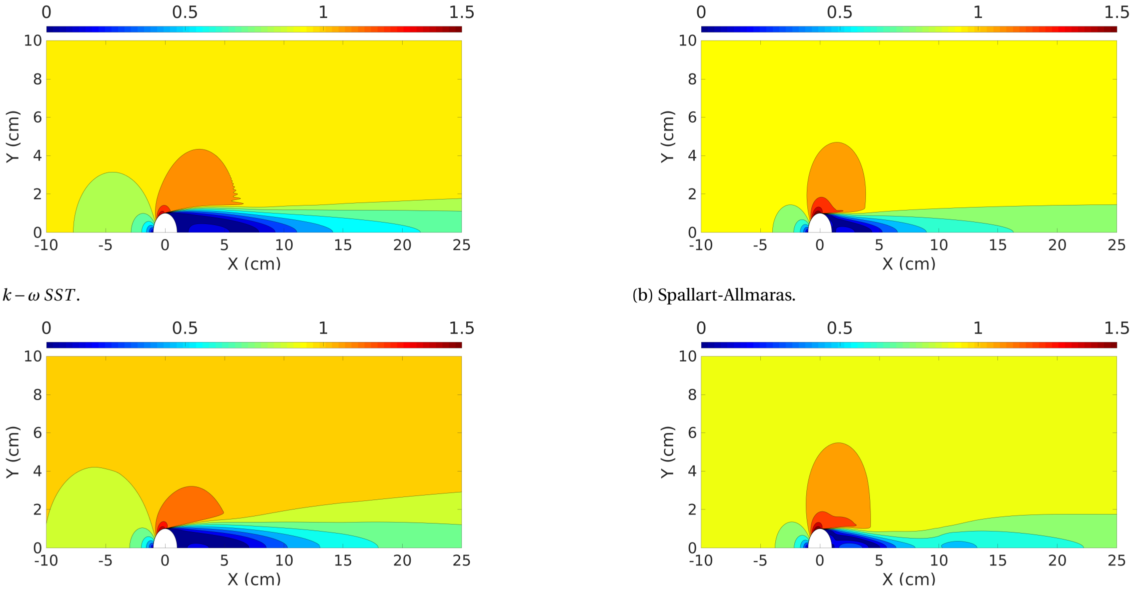
For volume forcing, similar to the Burgers' analysis, two regions are chosen: upstream and downstream. Upstream in this case does not refer to the region directly upstream of the cylinder but includes the flow around the cylinder and the wake. Figure 4.18 shows the two domain used for volume forcing. Both the domains are chosen such that the total cell count approximately corresponded to N=25 % and because the mesh is much finer near the cylinder, the downstream region appears much larger in volume. Both simulations are run with $K = 5$ to ensure convergence. Figure 4.19 shows the contours of velocity magnitude of the two volume cases compared to the original unforced simulations. Downstream forcing again is worse than upstream forcing as was observed in the Burgers' equation analysis. Also, the same slight increase in freestream velocity is again observed in the case of upstream forcing. Figure 4.20 shows the difference between forcing upstream and downstream by comparing velocity profiles at locations 1 and 2 which are marked in figure 4.18. Location 1 is around 2D downstream while location 2 is around 15D downstream of the half cylinder. Figure 4.20a shows the profiles at location 1 where upstream forcing is applied and figure 4.20b shows the profiles at location 2 which coincides with where the downstream forcing is applied. It is clear that in case of upstream forcing, the region that is forced closely follows the ground truth and also follows a similar trend downstream, but in case of downstream forcing, the profiles are farther away from the ground truth both in locations where it is forced and not forced.



(a) Upstream forcing with location 1 marked as the black line.

(b) Downstream forcing with location 2 marked as the black line.

Figure 4.18: Forcing domains for volume forcing



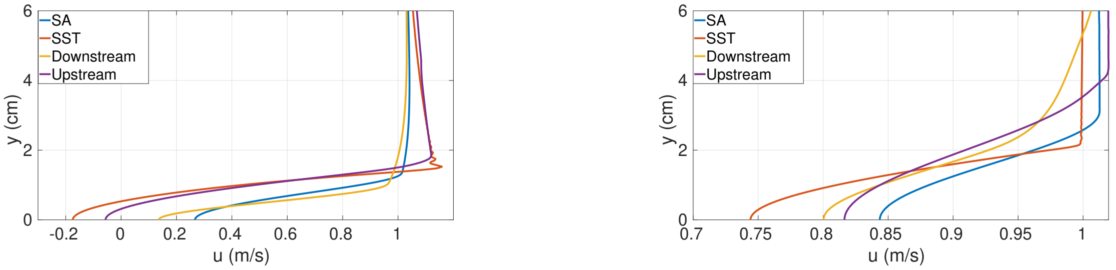
(a) $k-\omega\, SST$.

(b) Spallart-Allmaras.

(c) Upstream forcing.

(d) Downstream forcing

Figure 4.19: Contours of velocity magnitude (m/s) for original simulations and DA assisted simulation.



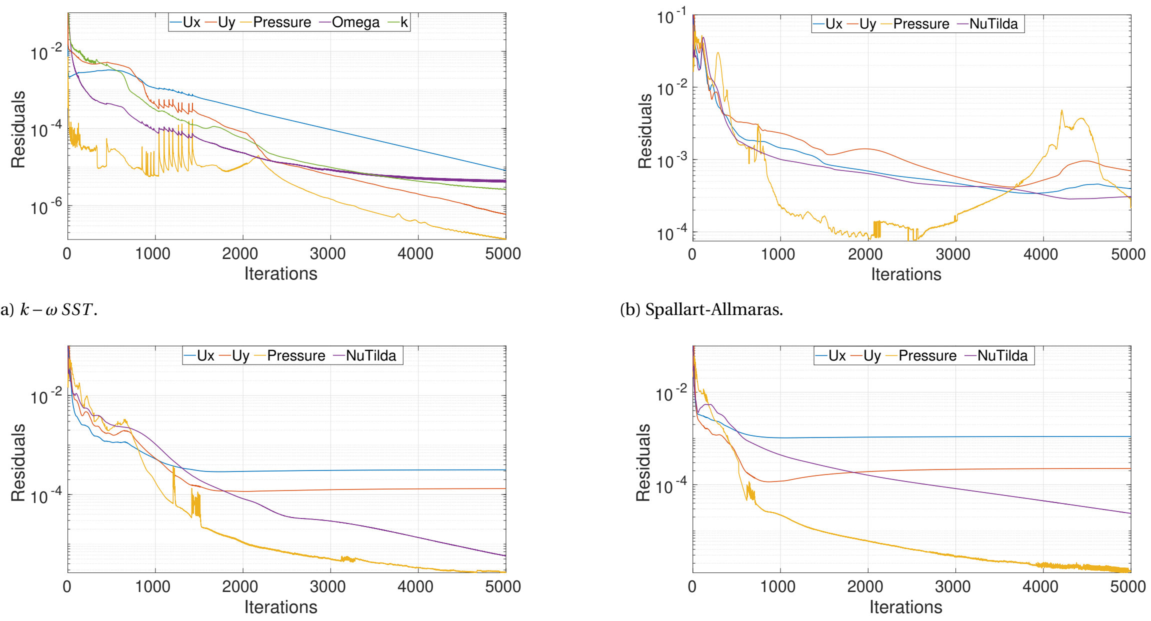
(a) Velocity profile comparison at location 1.

(b) Velocity profile comparison at location 2.

Figure 4.20: Difference in wake velocity profiles for upstream and downstream forcing.

4.2.5. Convergence

Figure 4.21. shows the convergence characteristics for the DA compared to the baseline SA and the ground truth simulation. The results are shown for the first 5000 iterations. The DA simulations show stable convergence behaviour after the initial convergence. This stabilization occurs faster if the forcing is high as can be seen in figure 4.21d where the velocity residuals stabilize after around 1000 iterations. The values of the residuals remains in the same order as the baseline simulation.



(a) $k-\omega$ SST.

(b) Spallart-Allmaras.

(c) K=2,N=25 %.

(d) K=10,N=50 %

Figure 4.21: Contours of velocity magnitude (m/s) for original simulations and DA assisted simulation.

4.3. Conclusions

Both the 1-D and the 2-D analysis presented very similar behaviours after the application of DA. In summary:

- The reduction in error (measure of how the solution approaches the ground truth) asymptotically reaches a value greater than zero for both parameters K and N.
- Forcing applied downstream, due to convection of the solution, cannot sufficiently affect upstream regions whereas the converse is true.
- Forcing using the current method causes the solution to reach an “equilibrium" that lies between the forcing solution and the original simulation without forcing.
- Excessive forcing, both in N and K can lead to stability issues.

- The rate of convergence increases upon application of DA.

The present chapter dealt with the application of the state observer method in 1-D and 2-D. Before moving on to a 3-D assessment using experimental data, the details of the experimental setup and data reduction are covered in the next chapter.

5

Experimental setup and data reduction

Part of the novelty of the present work is the use of 3-D velocity field information for use in DA. The 3-D velocity field information is procured by performing an experimental campaign of the flow around a car side mirror model. This chapter presents the experimental set up and methodology used for obtaining the velocity field around the model using robotic volumetric PIV. The first part of the chapter deals with the experimental setup including the wind tunnel specifications, details of the model, specifications of the robotic PIV system and the test conditions (section 5.1 and 5.2). The next part of the chapter, section 5.3 deals with the calibration and the test methodologies and the final section (5.4) provides a description of the data reduction techniques.

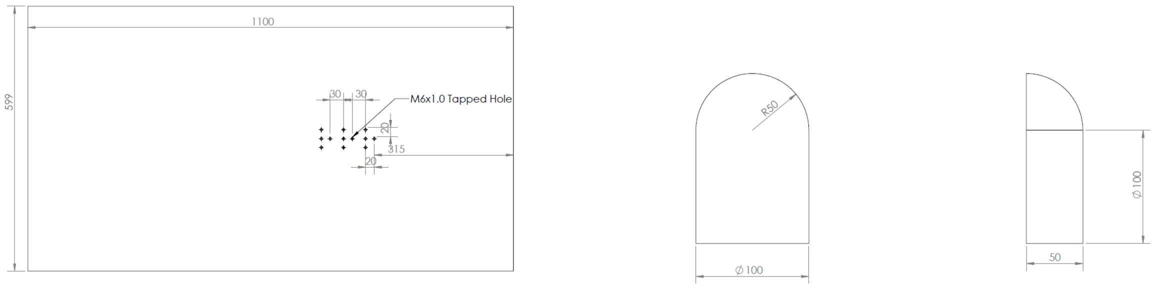
5.1. Wind tunnel, test section and model

The experiments are conducted in the W-Tunnel of the High Speed Laboratory at TU Delft. The W-Tunnel is a low speed open-circuit wind tunnel with nozzle exit areas that can be changed between 40 cm X 40 cm, 50 cm X 50 cm and 60 cm X 60 cm. The air flow is driven by a centrifugal fan powered my a motor, whose rotation rate is adjustable and can result in a maximum velocity of 35 m/s for the 40 cm X 40 cm cross section for which the minimum achievable turbulence intensity is reported to be 0.5 % ([1]). For the present study, the 60 cm X 60 cm nozzle exit section is used for which the maximum achievable velocity is approximately 15 m/s. The velocity is manually adjusted by observing and setting the dynamic pressure obtained from a pitot-static probe placed near the nozzle exit to the desired value. The final nozzle exit used results in a contraction ratio of 4.

The test model is a simplified car side mirror geometry that is fixed onto a flat wooden plate. The model consists of a quarter sphere of diameter 100 mm on top of a half cylinder. The engineering drawings of the plate and the model are shown in figure 5.1. The model has 4 M6 holes drilled into it that attach onto the middle set of holes drilled into the plate, which sets the flat side of the side mirror model 375 mm from the leading edge of the plate. The plate is placed such that the leading edge is 120 mm inside the nozzle exit and placed 100 mm above the floor of the nozzle exit to create a fresh boundary layer and avoid the wind tunnel boundary layer. The model is in an open-jet and causes a blockage of approximately 4.2 %. The flatness of the plate is ensured by using an anglemeter and ensuring that no part of the plate is inclined at more than 0.1 ° to the base reference (ground). To further ensure that there is no leading edge separation, a tuft is probed along the length of the leading edge and is seen to follow the flow. A tripping device is placed 50 mm from the leading edge to promote a fully developed turbulent boundary layer near the model. The boundary layer thickness δ_{99} is later calculated from PIV results to be 18 mm, 300 mm downstream of the leading edge when the model is not present. Both the car side mirror and the flat plate are painted black to avoid reflections while performing PIV. The final setup is shown in figure 5.2. Experiments are conducted using a single speed of 12 m/s.

5.2. PIV setup

The robotic PIV system used for the present study comprises of two main components, namely the seeding system, and the robotic PIV system which further comprises of the robot , the cameras and the illumination. The following sections will discuss each component briefly.



(a) Flat plate dimensions in *mm*.

(b) Side mirror model dimensions in *mm*.

Figure 5.1: Flat plate and model dimensions.

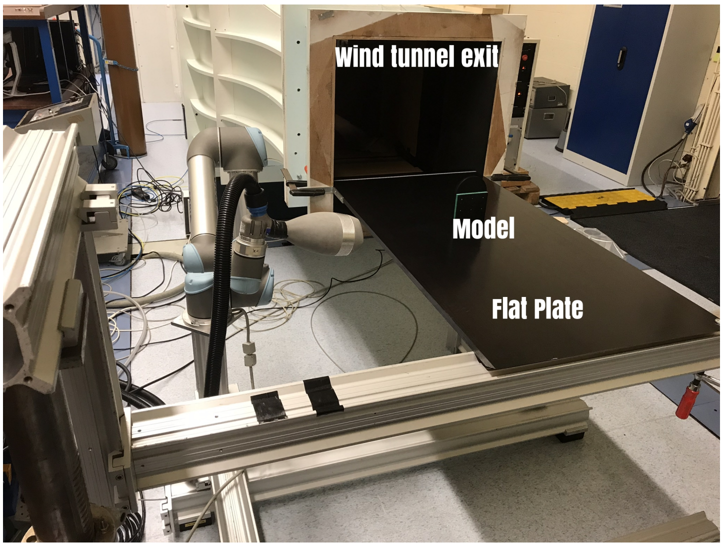


Figure 5.2: Setup of the experiment showing the model located at the wind tunnel exit, and the robotic PIV system.

5.2.1. Seeding system

Helium Filled Soap Bubbles (HFSB) are used as the tracer particle for the present study. These are bubbles that are produced by mixing air, helium and soap in such a proportion that the bubbles become neutrally buoyant . They are generated by small nozzles and the three constituents are supplied from a Fluid Supply Unit [1] by *LaVision*. The FSU controls the supply of the three constituents by means of pressure in the supply lines. For the present study all three constituents were maintained at 2 bar pressure. A total of 200 nozzles are placed on an aerodynamic rake. The rake consists of ten parallel wings placed 50 mm apart and 1 m in total height. Each wing contains 20 nozzles. The rake is placed in the settling chamber and is shown in figure 5.3. Due the contraction, the final seeded streamtube is around 50 cm X 25 cm in cross section area.

The seeding concentration of tracer particles is an important parameter in PIV and is estimated following Caridi et al.9[10].

$$C_{HFSB} = \frac{\dot{N}}{\dot{V}} = \frac{\dot{N}}{AU_0} \tag{5.1}$$

Here, $\dot{N}$ is the number of bubbles produced per second, $\dot{V}$ is the volumetric flow rate, A is the seeded area and U_0 is the velocity in the settling chamber. The bubbles are estimated to be generated at a rate of 30,000 /s per nozzle (Faleiros et al.[20]). Assuming only 60 % of the nozzles to be working, the production rate is $\dot{N} = 3.6 \cdot 10^6$ bubbles/s, giving a seeding concentration of $9.6 \cdot 10^6$ bubbles/m^3 or 9.6 bubbles/cm^3.

[1] Abbreviated as FSU

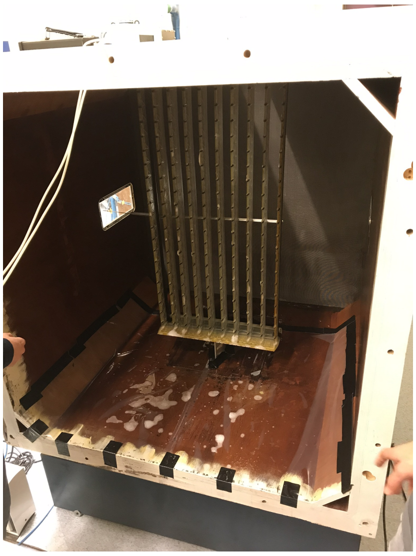


Figure 5.3: Seeding rake.

5.2.2. Robotic PIV system

The robotic PIV system as introduced by Schneiders et al.[44] consists of a Coaxial Volumetric Velocity [2] probe which houses four CMOS cameras along with an optical fibre for illumination, all of which are mounted on a robot.

CVV probe

The velocity probe used is the *LaVision MiniShaker Aero* which consists of an oval shaped housing for reducing aerodynamic interference (Jux [28]). Along with the cameras and the optical fibre, it contains a fixture for attaching onto the robot arm. The dimensions of the system (wxhxd) are 132x106x276 mm^3. The different parts of the system are shown in figure 5.4 and table 5.1 shows the specifications of the system used for the experiments.

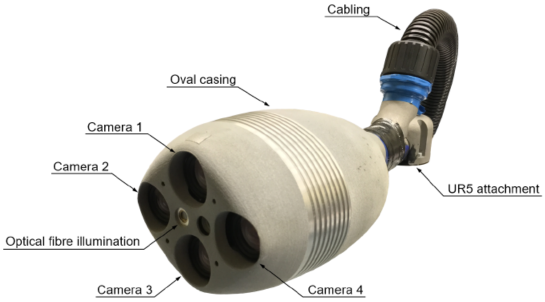


Figure 5.4: *LaVision MiniShaker Aero* CVV probe.

Illumination

The laser light is provided by a *Quantronix Darwin Duo Nd:YLF* high speed laser. The laser contains two cavities and is capable of producing 25 mJ of energy when operating at 1 kHz. The frequency can be adjusted between 0.2 and 10 kHz and the light is emitted at λ= 527 nm , which is green colored light on the visible spectrum. The light is routed to the CVV probe by means of an optical fibre which is connected to the laser head. The light from the probe expands and takes the shape of a cone as shown in figure 5.5. Although the light expands indefinitely, the portion shown in the figure represents the usable region of the light. The total volume amounts to approximately 16 L.

[2] Abbreviated as CVV

		Symbol	Specification	Unit
Optics	Focal length	f_i	4	[mm]
	Numerical aperture (at z=400mm)	$f_{\#}$	8	$[-]$
Imaging	X tomographic aperture	β_x	8	[˙]
	Y tomographic aperture	β_y	4	[˙]
	Sensor size	$s_x x s_y$	640 x 452	$[px^2]$
	Pixel pitch	δpx	4.8	$[\mu m]$
	Magnification	M	0.01	[-]
	Bit Depth	b	10	[bit]
	Acquisition frequency	f	857	[Hz]

Table 5.1: Specifications for experiment.

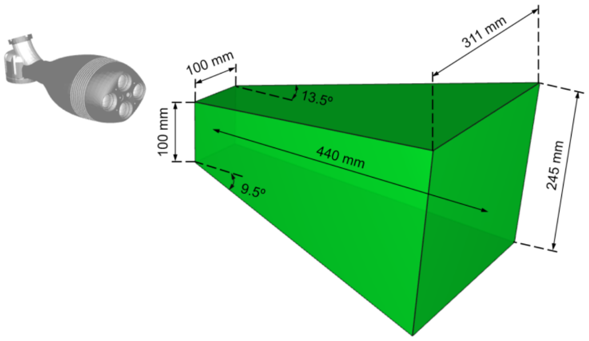


Figure 5.5: Laser cone along with its dimensions.

Robot

The CVV probe is mounted on a *Universal Robot-UR5* arm [3]. The robot cobtains six joints, namely, Base, Shoulder, Elbow, Wrist1, Wrist2 and Tool and has a maximum reach of 850mm along with six degrees of freedom. The specified accuracy of the robot is $\pm 0.01mm$ with a repeatability of $\pm 0.1mm$. The robot is shown in figure 5.7a. The robot comes with a tablet used for its positioning and is also accessible through the *RoboDK* software [4].

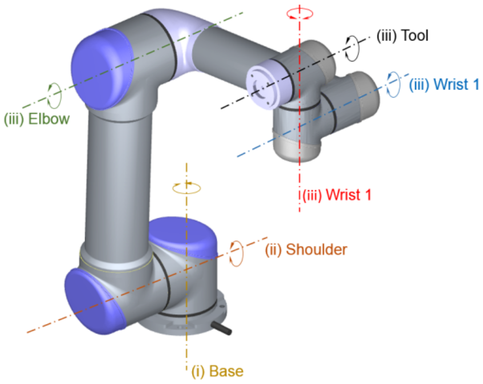


Figure 5.6: UR5 robot.

Apart from the components mentined above, an acquisition PC is used to synchronize and manage all the components. The synchronization, storing and processing of data is achieved through *LaVision Davis 10.05* software used along with a *LaVision Programmable Timing Unit.*

[3] https://www.universal-robots.com/products/ur5-robot/

[4] https://robodk.com/index

5.3. Experimental procedures

Apart from setting up the different components of the experiment, procedures like calibration and the creation of a test matrix have to be followed. The following section describes the different experimental procedures used.

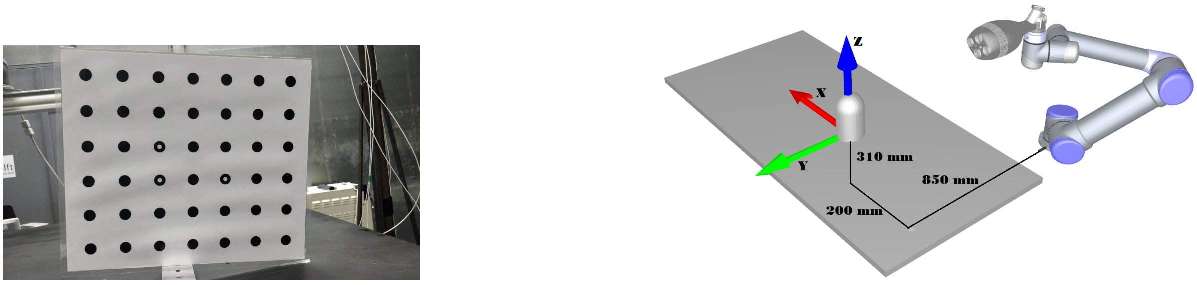
5.3.1. System calibration

Geometric and Optical calibration

A geometric calibration using the pinhole model (Soloff et al. [47]) is used to triangulate three-dimensional particles from two-dimensional projections captured by the cameras. A *LaVision* calibration plate as shown in figure 5.7a is used and the plate is imaged at a distance of 400 mm from the camera lens plane at multiple positions around the plate. The geometric calibration is followed by a volume self calibration (Wieneke [57]) on the previously obtained calibration in a freestream flow to further reduce the mapping errors. Finally, an Optical Transfer Function [5] (Schanz et al. [42]) is obtained. The OTF is used for reprojection from 3-D space to 2-D.

Robot Position Calibration

For all the experiments, the “origin" is considered to be at the base of the flat side of the side mirror model as shown in figure 5.7b along with the coordinate reference frame. This reference frame is defined as the global reference frame and all the data is defined using this frame. The position of the robot base is determined by imaging (around 100 images) a black plate marked with white dots which is placed on the back of the side mirror model and then using IPR to reconstruct the position of the dots. The average position of the dots is then used to determine the exact position of the robot base from the global reference frame and is shown in its final position in figure 5.7b. It must be noted that in the actual measurement campaign, the robot is placed in two positions, one which is shown in figure 5.7b and the other position is the mirror image of this poisition around the x-z plane. This is done to ensure complete measurement of the full volume around the model and enhance measurements which may have been obscured by shadows when measuring from the first position.



(a) Calibration plate.

(b) Robot position.

Figure 5.7: Calibration plate and position of robot with respect to the model.

5.3.2. Measurement Volume

In the previous study using a similar geometry (DeVilliers [16]) discusses the main flow features that can be expected around a car side mirror model including the position of the reattachment point approximately 2.25 D downstream of the model. In the present study, which is conducted at a lower Reynolds number than that of DeVilliers, it was decided that the measurement volume should include a region that extends up to 2.5 D downstream of the model. In total, 15 overlapping sub-volumes are measured around the car side mirror, with a larger overlap percentage in the wake region to form a total measured region of approximately 40L. The measured volume is shown in figure 5.8.

A typical measurement run includes starting the wind tunnel, letting the flow stabilize, switching on the HFSB seeder and using the live view option in *DaVis* to monitor the seeding concentration. Once the seeding is allowed to stabilize and the seeding concentration looks nominal, the model is cleaned for contamination of soap and the images are acquired. Once the images are obtained, the seeder is turned off and the images are quickly inspected visually followed by moving the robot to the next position and then repeating the same

[5] Abbreviated as OTF

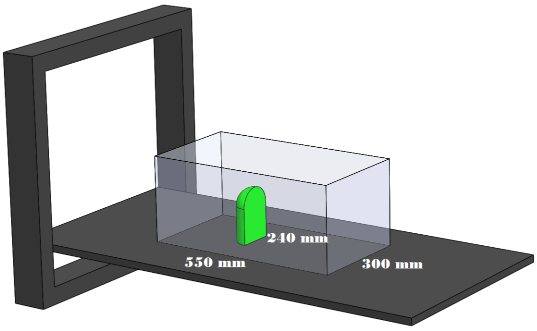


Figure 5.8: Total measured volume along with dimensions.

procedure.

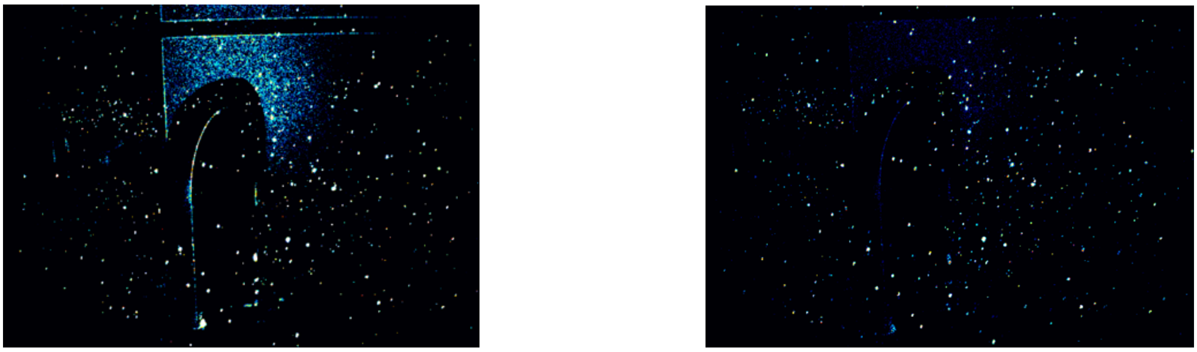
For each sub-volume, 20,000 images are acquired at 857 Hz. The number of images acquired is large to reduce the uncertainty in velocity in the wake region. The maximum standard deviation of velocity from the mean in the wake region is observed at around 40%. For obtaining a velocity field, not all 20,000 images are used at every point since some images might not have particles in them at a particular location. Rather, the number of particles in the region over the entire imaging time is what is used to calculate the velocity field through ensemble averaging (covered in section 5.4.3). For the wake region, the number of particles is observed to be around 5000, therefore for a freestream velocity of 12 m/s the uncertainty in velocity with a 95 % confidence level is ϵ_u= 0.14 m/s or a relative uncertainty of 1 %.

5.4. Data Reduction

Once the raw images from the experiment are obtained, they are subjected to a series of procedures to finally obtain a velocity field. These include pre-processing, particle tracking and ensemble averaging.

5.4.1. Image pre-processing

Pre-processing techniques are applied on the raw images to enhance the quality of the images are reduce any background noise that might be present due to reflection of the laser light from the model or parts of the wind tunnel. For the present study, a Butterworth high pass filter (Sciacchitano and Scarano [45]) with a filter length of seven images is used. The difference in the raw image and the image obtained after pre-processing is shown in figure 5.9.There is a stark difference between the two images as the reflections from the model and the background which are clearly seen in the raw image are removed in the filtered image.



(a) Raw image.

(b) After application of Butterworth filter.

Figure 5.9: Image before and after pre-processing.

5.4.2. Particle Tracking

After pre-processing, the images are then subjected to the next step of processing that is particle detection and tracking. Once the position of the particles is reconstructed from the images, they are processed with the

Group	Parameter	Value	Unit
Volume settings	X min	-230	[mm]
	X max	+250	[mm]
	Y min	-130	[mm]
	Y max	+180	[mm]
	Z min	215	[mm]
	Z max	+600	[mm]
Particle detection	2D particle detection intensity threshold	10	[counts]
	Allowed triangulation error	1.5	[px]
Shaking	Outer loop iterations	4	[-]
	Inner loop iterations	4	[-]
	Particle shaking	0.1	[vox]
	Remove particles if closer than	15	[vox]
	Weak intensity particle removal threshold	10	[%]
Particle image shape and intensity	OTF size factor	0.5	[-]
	Residuum OTF radius	1	[px]
Acceleration limits	Maximum absolute particle shift	7	[vox]
	Maximum relative particle shift	35	[%]
Tracking	Minimum track length	4	[-]
	Polynomial fit order	2	[-]
Velocity limits	Vx	-15 ± 38	[vox]
	Vy	0 ± 20	[vox]
	Vz	0 ± 20	[vox]

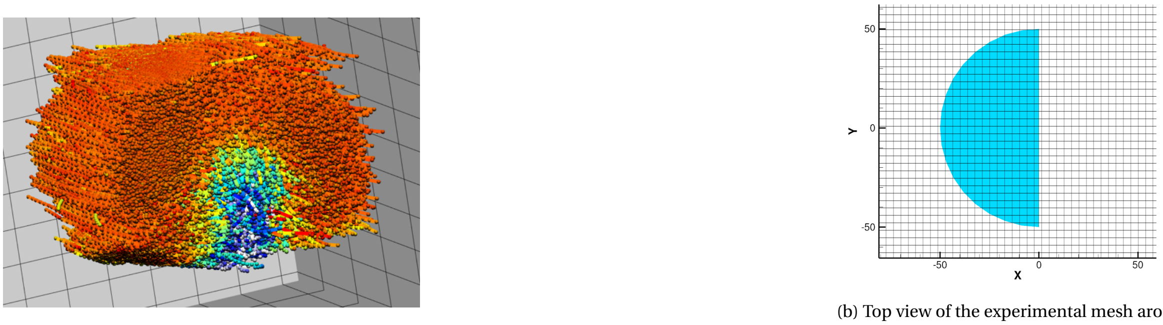
Table 5.2: Parameters for STB

STB algorithm in *DaVis 10.05* which uses temporal information of the particles to construct particle tracks. The algorithm contains different settings which can be optimized. The final settings are shown in table 5.2. Interested readers are encouraged to go through the *DaVis* manual [6] for an understanding of what each parameter signifies.

[6] https://www.lavision.de/en/downloads/manuals/systems.php

5.4.3. Ensemble averaging

The velocities obtained after STB are particle tracks otherwise known as a *Lagrangian* description of the flow as shown in figure 5.10a. However, it is often convenient to work with an *Eulerian* description, especially for the purpose of the present study where the experimental data is interpolated onto a CFD grid. This operation is performed by discretizing the domain into interrogation volumes, commonly referred to as bins, and then averaging the data over these bins. Different methods of ensemble averaging are presented in Aguera et al. [4]. For the purpose of the present study, a Gaussian filter is employed. A bin size of 15 mm has been chosen as the final grid size based on making a trade-off between better spatial resolution and data convergence. Better spatial resolution can be obtained by using smaller bin sizes but it comes at the cost of not having fully converged data as smaller bin sizes mean lower number of particles. This tradeoff was made after evaluating different bin sizes between 5 and 30 mm. Even though the spatial resolution is limited, the number of data points on the grid is increased by using an overlap of 75 % between bins. This effectively provides a data point every 3.75 mm. The grid points are shown in figure 5.10b.



(a) Example of the particle tracks obtained after STB.

(b) Top view of the experimental mesh around the car side mirror.

Figure 5.10: Conversion from a lagrangian to an eulerian description of the flow.

After obtaining the velocity field, the final stage of the present work is to apply the experimental data as ground truth for 3-D DA. The results from the experiments and the DA are provided in chapter 6.

6

Experimental assessment

Once the data from the experiment has been processed and is represented on an eulerian grid, it can be used for DA. This chapter firsts presents the experimental results and the important topological features that can be observed in section 6.1, then a brief discussion of the numerical grid and boundary conditions for the 3-D steady RANS around the side mirror model will be provided in section 6.2 followed by application of DA and its results in section 6.3.

6.1. Flow topology of experimental results

Even though the experimental velocity obtained is ensemble averaged, it can be considered as the time averaged flow due to the long measurement times for each cone ($\approx 25s$). Due to the geometry of the body (quarter sphere on top of a half cylinder), there is vortex shedding observed in the wake of the side mirror across the half cylinder region. Assuming a Strouhal number of 0.2 and considering the length scale to be the diameter of the model, for a freestream velocity of 12 m/s the frequency of shedding is 24 Hz meaning that for each measurement around 620 cycles of shedding are captured thereby justifying the assumption of time-averaged flow.

6.1.1. Velocity contours and streamlines

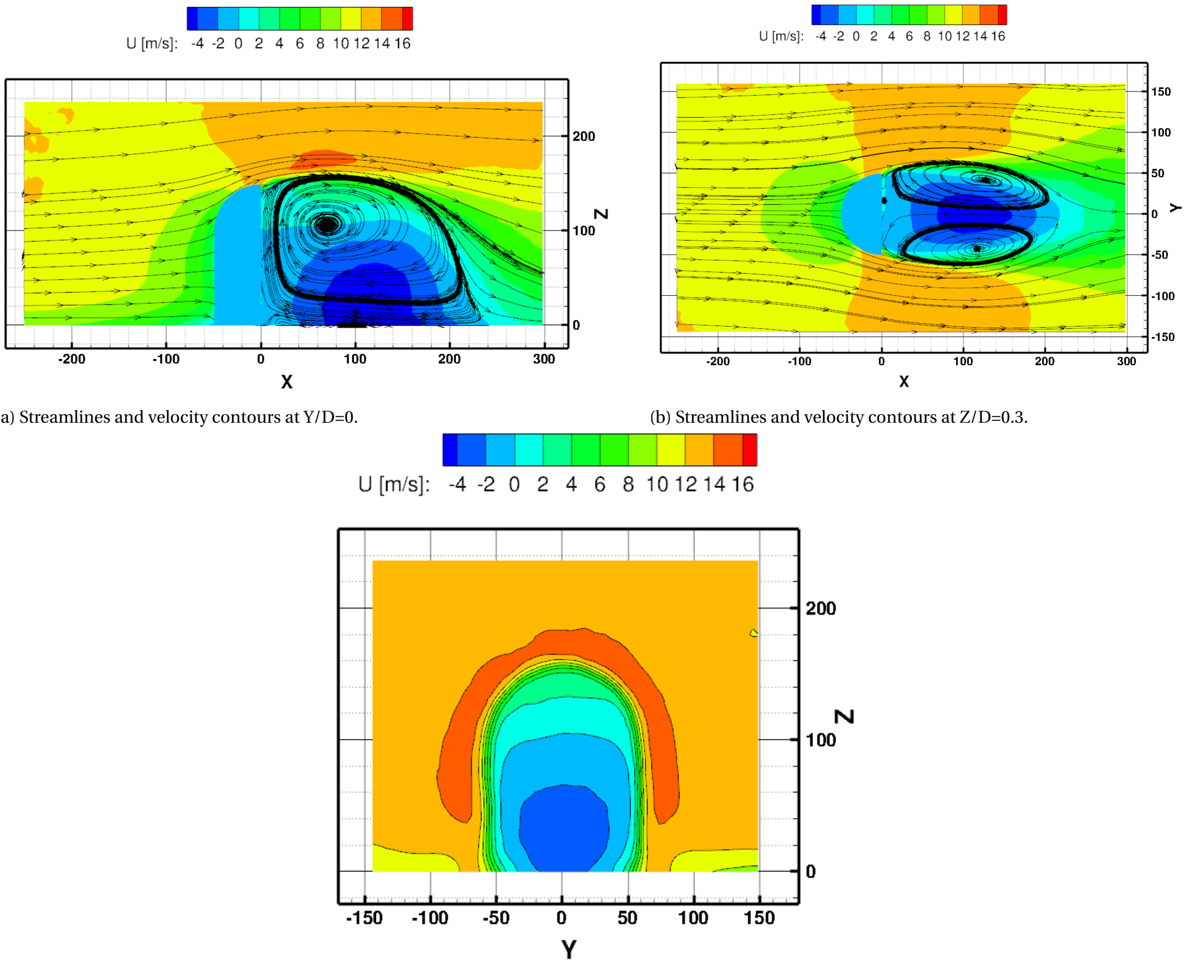
The time averaged streamwise velocity contours and streamlines for the flow are shown in figure 6.1. The flow is from left to right and exhibits some key features.

- There is a large wake region behind the model which contains a distinct recirculation zone that is bounded on all side by free shear layers.
- The reattachment point which is defined as the stagnation point of steamwise velocity in the Y=0 plane at a height of the first grid point from the ground (Z/D=0.0375) is at a distance of **X/D=2.4±0.0375**.
- The flow is symmetric on either side of the mirror as is expected and can be seen from figure 6.1b.

6.1.2. Vortical structures

The time-averaged flow around the car side mirror exhibits many vortical structures and these are shown in figures 6.2 as isosurfaces of streamwise vorticity in figures 6.2a and 6.2c and isosurfaces of spanwise vorticity in figure 6.2b. Since the model is placed on a flat plate, the boundary layer that develops over the flat plate is under the influence of an adverse pressure gradient from the model resulting in its roll-up and wrapping around the model forming a distinct horseshoe vortex. The horseshoe vortex does not exhibit a significant effect on the mean flow except to draw higher momentum fluid to the base. A secondary vortex is also visible near the base of the model and is of opposite sign to the horseshoe vortex. Another prominent structure is a pair of counter-rotating vortices and are referred to in the present study as tail vortices. Figure 6.2c shows that the origin of these vortices coincides with the downwash in the recirculation region which is visible when the streamlines from the top shear layer start to diverge. A vortex is formed at the foot of the model and below the recirculation region and is shown in figure 6.2b and is further referred to as the base vortex. The main flow features discussed bear resemblance to those found in [16]. However, no mention of the tail vortices found in the present study are found. Furthermore, the author also finds a secondary horseshoe vortex around the base of the model which could not be resolved in the present study.

One of the objectives of the present study is to apply experimental data directly as it is obtained from the experiment and not to apply any filters to reduce the divergence error in the data. Figure 2.1a shows the divergence error at 0.1 Hz around the model for the original data. This is the same data that is used for DA.

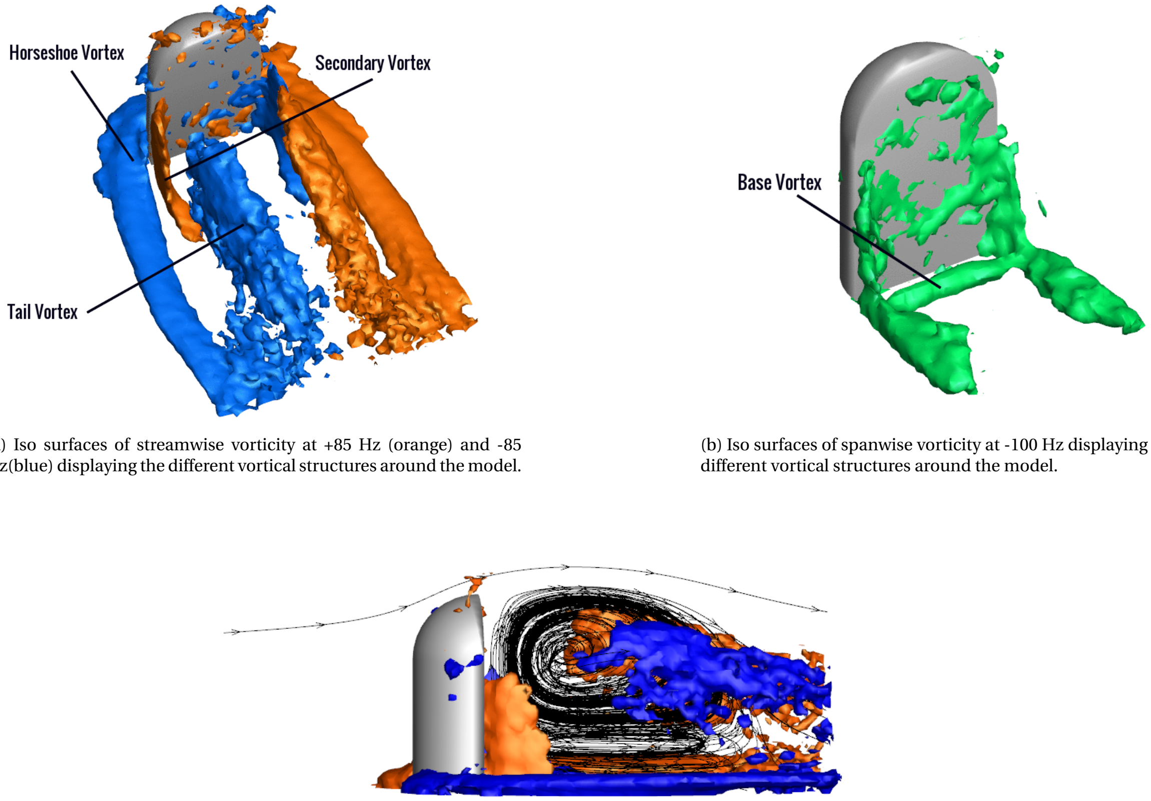


(a) Streamlines and velocity contours at Y/D=0.

(b) Streamlines and velocity contours at Z/D=0.3.

(c) Velocity contours at X/D=0.5.

Figure 6.1: Streamwise velocity contours and streamlines of time-averaged flow around car side mirror. All dimensions are in mm.



(a) Iso surfaces of streamwise vorticity at +85 Hz (orange) and -85 Hz(blue) displaying the different vortical structures around the model.

(b) Iso surfaces of spanwise vorticity at -100 Hz displaying the different vortical structures around the model.

(c) Iso surfaces of streamwise vorticity along with 3-D streamlines. Side view of the model is shown.

Figure 6.2: Vortical structures around model.

6.2. Computational setup for 3-D RANS

RANS simulations for the 3-D case are also performed using OpenFOAM v1706. The computational domain used for the simulations is shown in figure 6.3. The origin for the domain has been considered as the middle of the base of the flat surface of the side mirror to facilitate easy data comparison when used with experimental data. The domain extends for 20 D downstream of the origin, 10 D upstream, 10 D on either side and 10 D above the mirror. The domain consists of 7 boundary patches namely, inlet , outlet, ground, top, 2 sides and the side mirror.

6.2.1. Boundary and initial conditions

The boundary and initial conditions used follow the same principles as in the 2-D half cylinder case covered in section 4.2.1. However, in the 2-D half cylinder case, the ground is assigned a slip velocity. In the 3-D case, the ground is assigned as a no-slip wall to replicate the conditions experienced in the experiment. The inlet velocity is assigned a uniform velocity and no profile is specified. The $k-omega\ SST$ model has been used for all the simulations. Additionally, a new field variable called $UAvg$ is defined that computes the average of the velocity field for every iteration after stabilization of the solution which is specified by starting the averaging after a set number of iterations have progressed. Averaging is required since the side mirror exhibits unsteady behavior that resembles vortex shedding. Table 6.1 shows the different boundary and initial conditions used in the 3-D simulations.

6.2.2. Discretization

Discretization or meshing of the domain is done by dividing the domain into different blocks that grow in size starting from the side mirror to the boundaries of the domain. The different blocks are assigned different element sizes and the jump between blocks is made smooth. The mesh is made finest near in the block closest

Variable	Units	Inlet	Outlet	Top	Ground	Sides	Side mirror	Initial Field
p	m^2/s^2	0	0	symmetry	symmetry	symmetry	$\partial p/\partial n = 0$	0
u	m/s	(12 0 0)	$\partial u/\partial n = 0$	symmetry	no-slip	symmetry	no-slip	(12 0 0)
ω	$1/s$	1.8	$\partial \omega/\partial n = 0$	symmetry	symmetry	symmetry	Wall function	1.8
k	m^2/s	0.086	gradient=0	symmetry	symmetry	symmetry	1E-10	0.086
$EVol$	m/s^2	(0 0 0)	(0 0 0)	symmetry	symmetry	symmetry	(0 0 0)	(0 0 0)

Table 6.1: Boundary conditions for 3-D case.

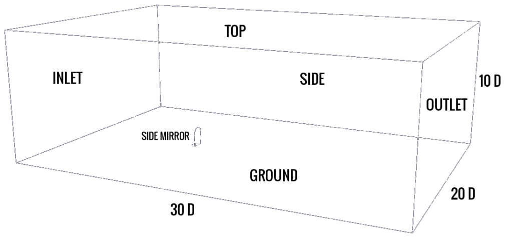


Figure 6.3: Computational domain for 3-D simulations.

to the side mirror and subsequently increased in coarseness. The element size of the smallest mesh is around 3 mm [1]. This roughly corresponds to the same spatial resolution of the experimental mesh and ensures minimal errors when interpolating experimental data onto the CFD grid. Furthermore, this mesh was able to provide converged results in a baseline simulation of the side mirror where no forcing was applied, thus proving its suitability for DA. The total number of elements in the domain is ≈ 3.5 million. The maximum aspect ratio is 470, average non-orthogonality is 2.5 degrees and maximum skewness is 2.53. The final mesh is shown in figure 6.4 and its comparison to the experimental mesh is shown in figure 6.5. In the present study, no analysis has been made on the effect of the grid size on the working of DA and only the mesh shown has been used for all simulations.

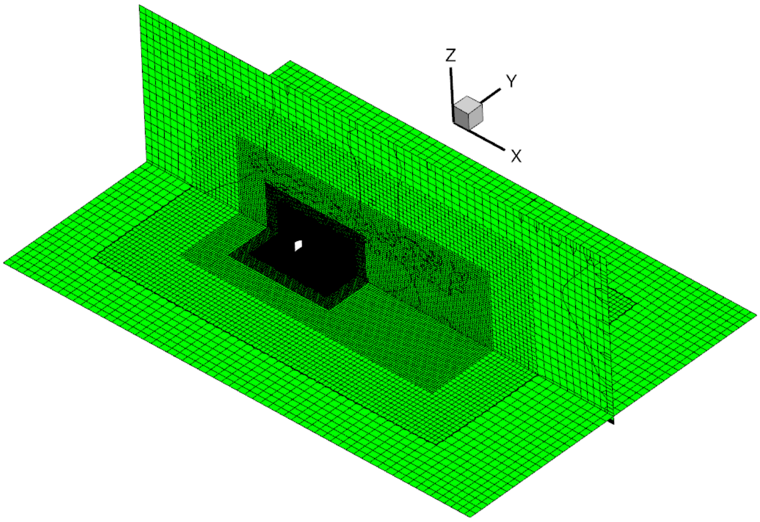
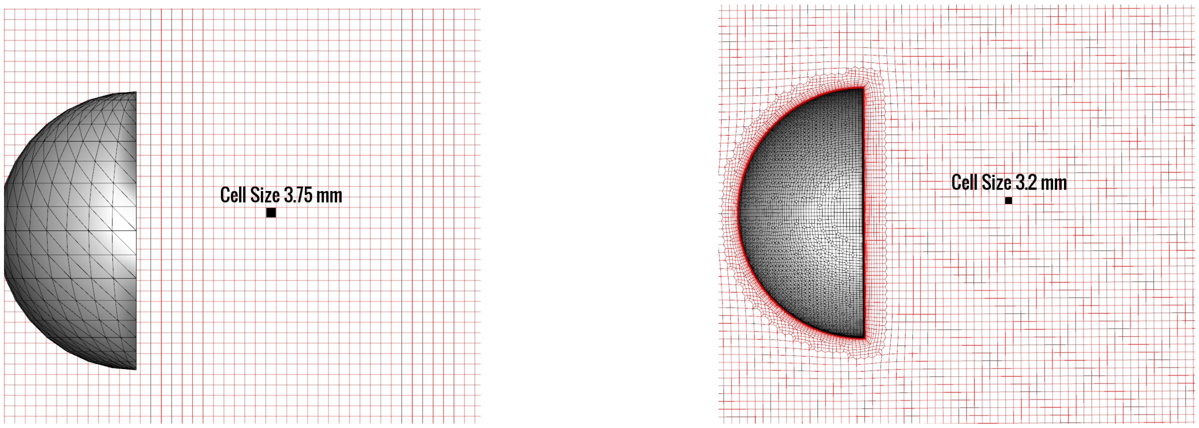


Figure 6.4: Computational Mesh shown in two planes.

6.2.3. Solver settings

For all the simulations performed, the $k - omega\ SST$ model has been used. The required tolerance for convergence has been specified as 1E-8 for all variables and relaxation factors of 0.9 for velocity and 0.8 for

[1] This does not include the prism layers added near the walls



(a) Experimental mesh.

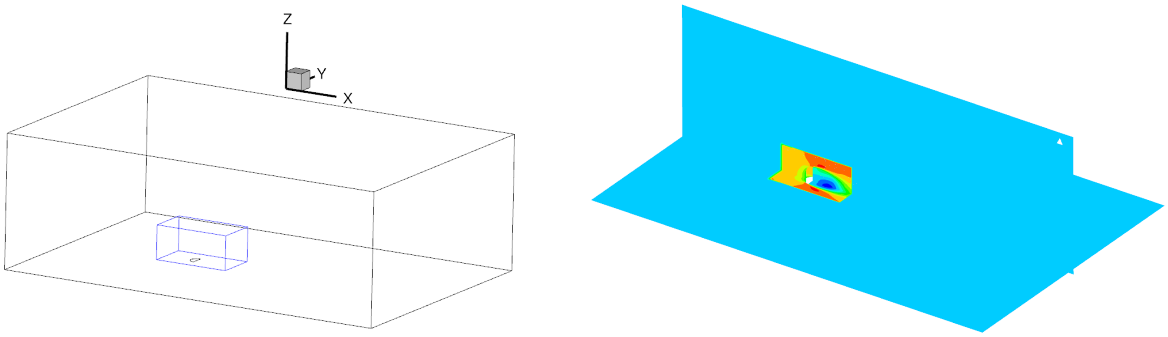
(b) Computational mesh near side mirror.

Figure 6.5: Comparison of experimental and computational meshes.

pressure. The low tolerance ensures that the simulation does not converge before the iteration limit, which is set to 5000, is reached thereby ensuring all simulations run for the same number of iterations, making their convergence analysis easier. The velocity field is averaged for every iteration starting from 3000.

6.3. Random forcing for side mirror

Random forcing for the side mirror case is different from the Burgers equation and the half cylinder case in that those two cases had the possibility of generating u^* for the entire domain. However, for the side mirror case, the u^* can only be generated in the volume where data from the experiment is available. The comparison of the domain size for the computational grid and the domain where experimental data is available os shown in figure 6.6a where the blue box indicates the region available from experimental data. If all the experimental data is interpolated onto the CFD grid around 32 % of the total number of cells in the domain contain u^*. For random forcing, the same two parameters ,N and K , are varied and their effects studied. However, as mentioned, here N=100 % corresponds to only 32 % of the total mesh and is shown in figure 6.6b. Figure 6.5 shows a key difference between the experimental and computational grid. In the computational grid, the mesh near the boundary is refined to capture boundary layer effects while it remains the same as the other parts of the domain in the experimental grid. The experimental results are not resolved enough to capture the velocity profiles near the boundary layer and ensemble averaging over bin sizes that are large compared to the boundary layer thickness, smoothens gradients. CFD on the other hand needs to have a good estimate of the boundary layer profile and using experimental data which is not resolved enough near boundaries as u^* causes divergence of the solution. Thus for all the CFD simulations, a buffer layer of 1.5 cm from all no-slip wall boundaries and 2 cm from the ground is provided where forcing is not applied. The different cases that are simulated are shown in table 6.2. In addition to the different cases that are run, a baseline simulation is run where no forcing is applied.



(a) Experimental domain (blue) compared to computational domain.

(b) N=100 % interpolated onto the CFD grid.

Figure 6.6: Comparison of experimental and computational domains.

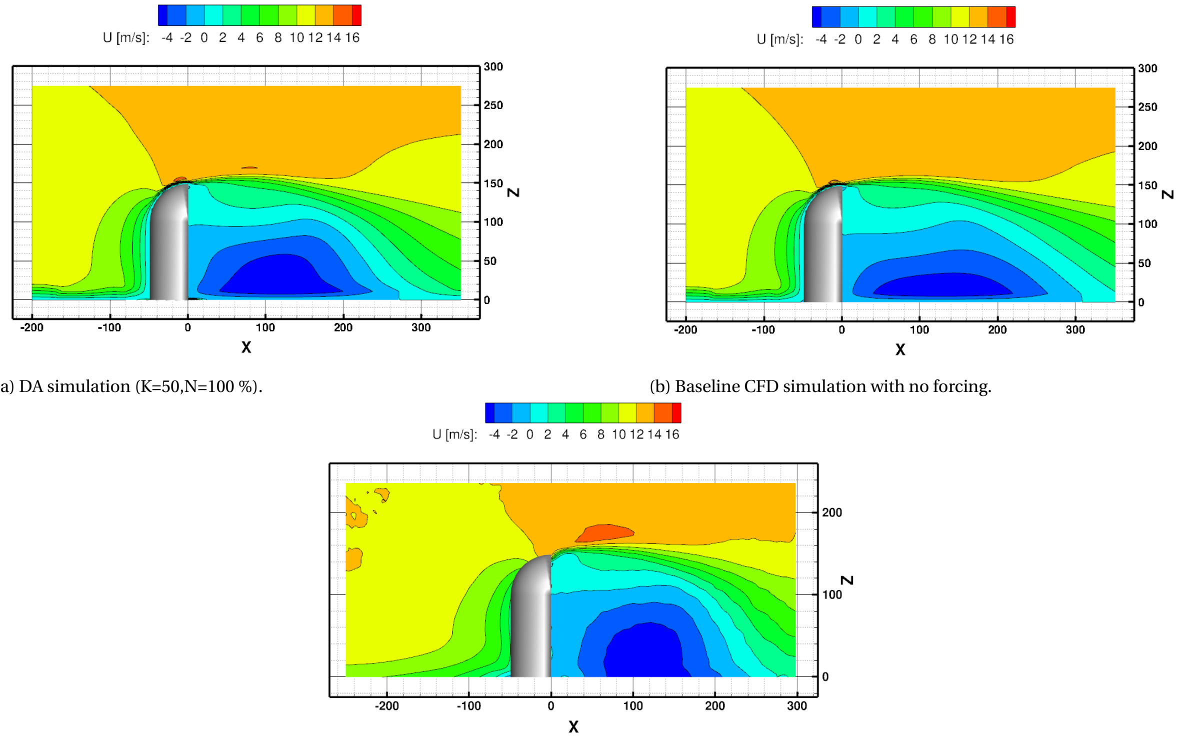
$N \vert K$	10	20	30	50
15%	✓	✓	✓	✓
30%	✓	✓	✓	✓
45%	✓	✓	✓	✓
60%	✓	✓	✓	✓
100%	✓	✓	✓	✓

Table 6.2: Different runs for DA on side mirror. Tick mark indicates that the simulation converged.

6.3.1. Comparison of topological features- A qualitative assessment

Qualitative analysis of the DA is performed for the case with the highest forcing (N=100 %, K=50). The streamwise velocity contours for the case with DA, the baseline case and the PIV results are shown in figures 6.7,6.8 and 6.9. When no forcing is applied (baseline simulation), there are significant differences between the experimental result and the simulation, especially in the wake region. In the baseline simulation, the wake region is longer, with the reattachment point at X/D=3.07 ± 0.0375, differing by almost 28 % from the experimental value. The wake is more elongated and shaped differently than the experimental wake as seen from an isosurface plot of zero velocity in figure 6.10. In general, the wake of the baseline case is more V-shaped while that of the experiment is oval. When forcing is applied, the simulation starts emulating the experimental velocity although there are still noticeable differences. With the highest forcing, the reattachment point is at X/D= 2.7 ± 0.0375 downstream, constituting a reduction in error in position of the reattachment point from 28 % to 12.5 % from the experimental value. The shape of the wake also changes from the V-shape and comes closer the shape exhibited by the experimental results.

A comparison of the vortical structures are shown in figure 6.11. Both the baseline and the DA simulation produce horseshoe vortices that are farther away from the model than the experimental data. While the core of the horseshoe vortex is around 1 D away from the centreline of the model on each side for the experimental data, it is around 1.3 D away from the model for the baseline case and does not change significantly in position or structure in the DA simulation. The main differences occur in the secondary and the tail vortices. The secondary vortex for the DA resembles that of the experiment in structure. It should be noted here that the actual magnitudes of vorticity are not compared since vortex cores, particularly in the present study are small regions of high velocity gradients and the relatively large bin size and the gaussian filter used for ensemble averaging smoothens gradients. The tail vortex in the baseline case moves outbound with respect to the median Y/D=0 plane as it propogates downstream. The DA simulation however exhibits a structure similar to that of the experiment where the tail vortex propgates dowstream while also moving towards the ground. A noticeable difference occurs further downstream where in the experiment, the tail vortex seems to start interacting with the horseshoe vortex and in the DA simulation, a different structure arises where another vortex with the same orientation of the horseshoe and the tail vortex appears under the tail vortex and merges with the tail vortex downstream. Another difference between the DA simulation and the experimental result shows the tail vortices moving closer towards each other in the simulation while this effect though present, is not as pronounced in the experimental results This is also seen in figure 6.12.

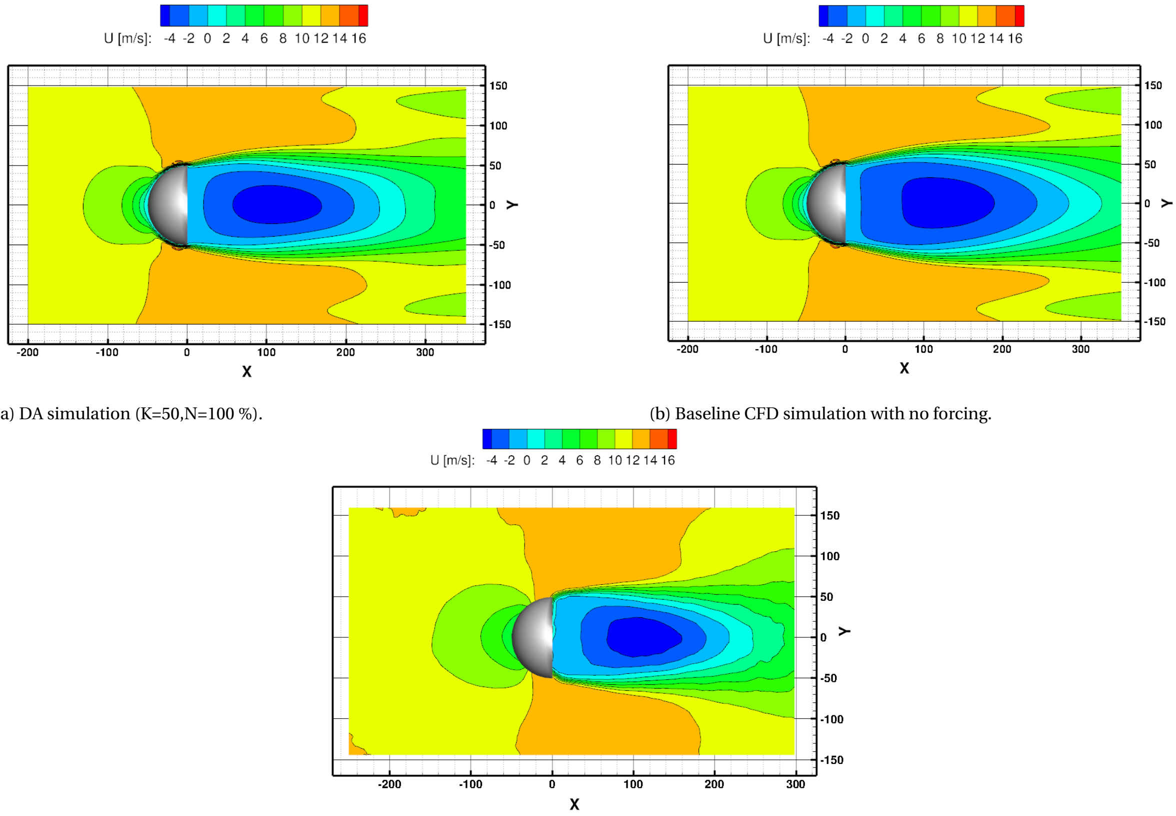


(a) DA simulation (K=50,N=100 %).

(b) Baseline CFD simulation with no forcing.

(c) Ground truth u^*

Figure 6.7: Comparison of streamwise velocity profiles at Y/D=0. All dimensions are in mm.

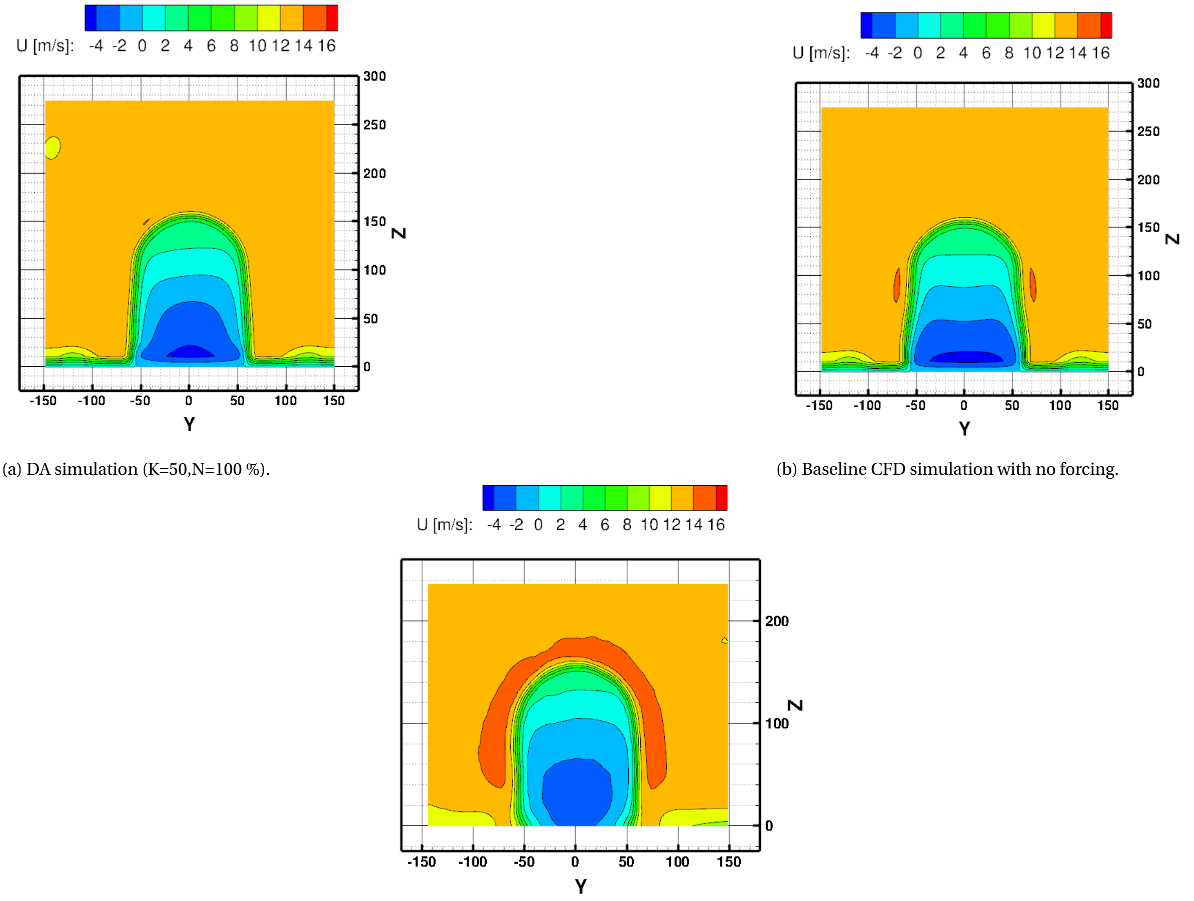


(a) DA simulation (K=50,N=100 %).

(b) Baseline CFD simulation with no forcing.

(c) Ground truth u^*

Figure 6.8: Comparison of streamwise velocity profiles at Z/D=0.3. All dimensions are in mm.

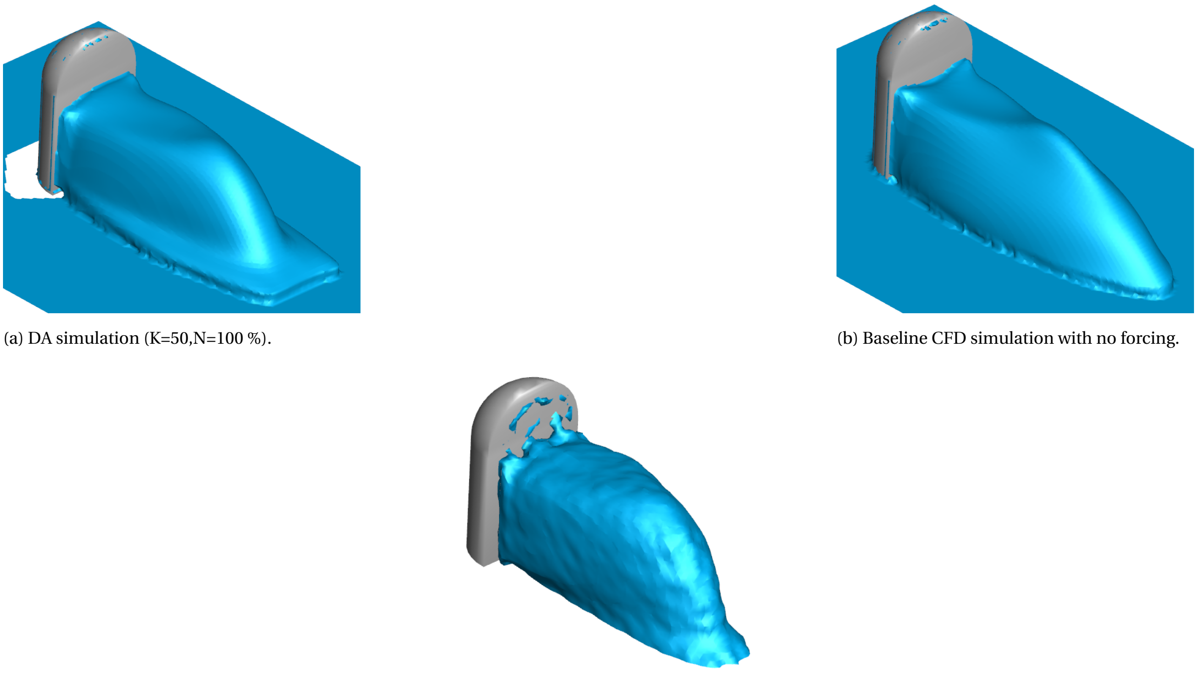


(a) DA simulation (K=50,N=100 %).

(b) Baseline CFD simulation with no forcing.

(c) Ground truth u^*.

Figure 6.9: Comparison of streamwise velocity profiles at X/D=0.5. All dimensions are in mm.

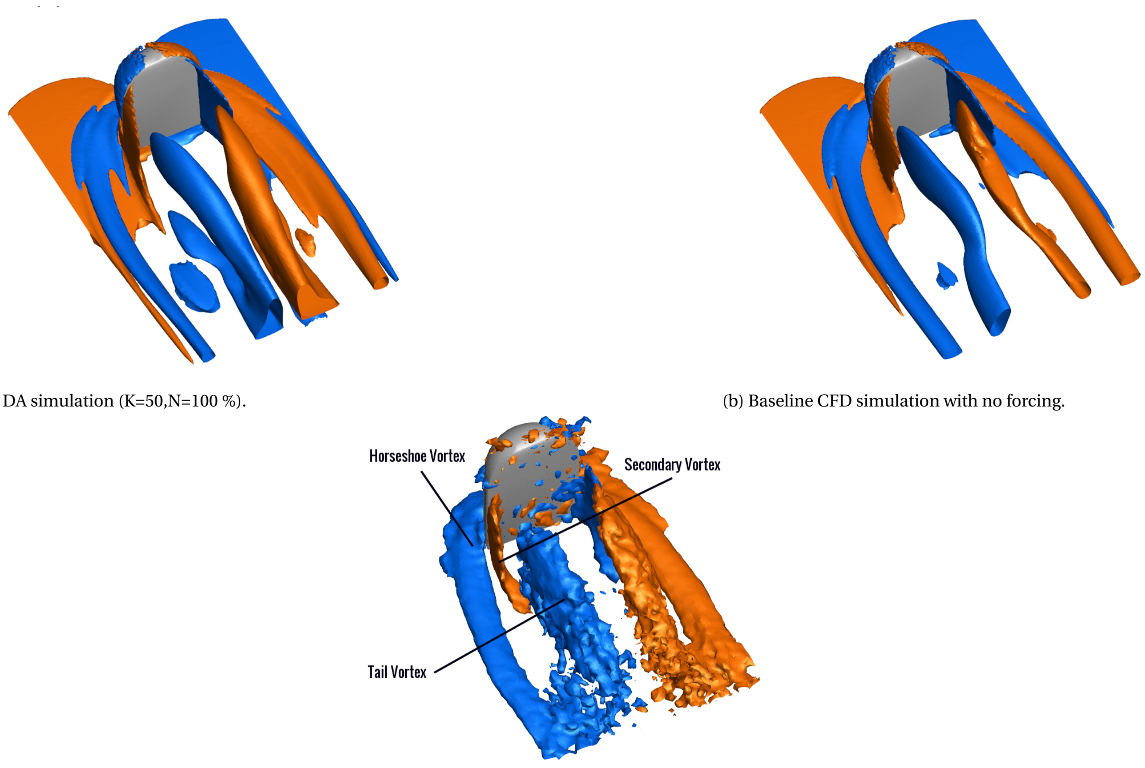


(a) DA simulation (K=50,N=100 %).

(b) Baseline CFD simulation with no forcing.

(c) Ground truth u^*

Figure 6.10: Iso-surfaces of zero streamwise velocity.

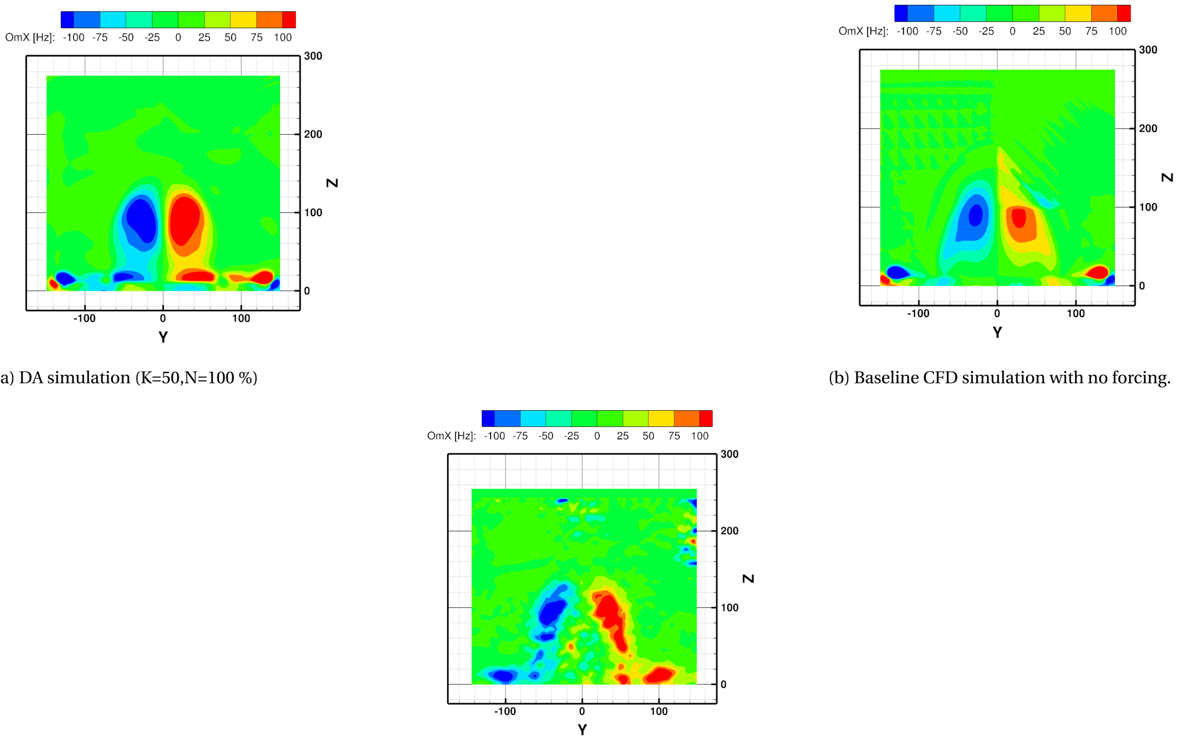


(a) DA simulation (K=50,N=100 %).

(b) Baseline CFD simulation with no forcing.

(c) Ground truth u^*

Figure 6.11: Iso-surfaces of streamwise vorticity at +85 Hz (orange) and -85 Hz(blue) around model.



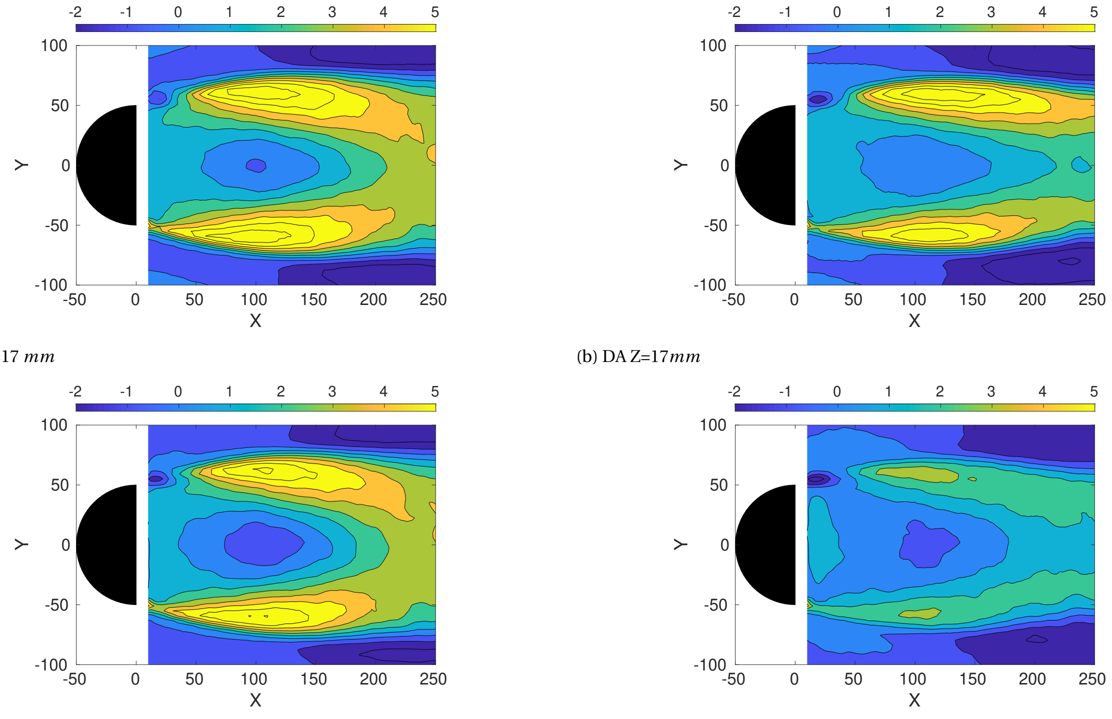
(a) DA simulation (K=50,N=100 %)

(b) Baseline CFD simulation with no forcing.

(c) Ground truth u^*

Figure 6.12: Iso-contours of streamwise vorticity at X/D=1.8.

It comes as no surprise that the horseshoe vortices are not affected by the DA. Since the DA is not applied at a height below 2 cm, where the horseshoe vortices exist (see figure 6.12), the velocity field in those regions does not conform to what the experimental results show. Figure 6.13 shows the difference in velocity $u^* - u_{DA}$ and $u^* - u_{CFD}$ for two different Z locations. The first (figures 6.13a and 6.13b) are taken at Z=17 mm which is a region where forcing is not applied, and the second (figures 6.13c and 6.13d) are taken at Z=24 mm which is just above where the forcing starts. At the first height, the error in velocity for both the DA and the baseline case is approximately the same while in the second location, the error in DA is significantly lower than that of the baseline case. All the figures for DA are plotted for the case of N=100 % and K=50. The base vortex as well is not replicated due to its proximity to the boundary.



(a) CFD Z=17 mm

(b) DA Z=17mm

(c) CFD Z=24 mm

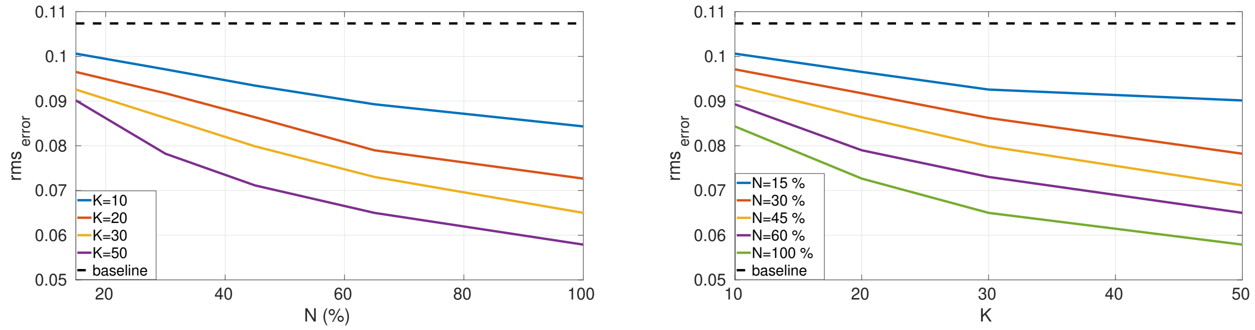
(d) DA Z=24 mm

Figure 6.13: Contour plot of error in streamwise velocity (m/s) for two different heights comparing baseline CFD and DA (K=50,N=100 %).

Figure 6.14 shows the rms error in velocity for different values of N and K. The error shows similar behaviour to the 1-D and 2-D cases. Here the rms error is defined in the same way as equation 4.6 but with $u_{ref} = 12m/s$ and considering all three components of velocity. The error asymptotically reduces to a value above zero error and even the simulation with the least forcing reduced the error in velocity as compared to the baseline case, which is shown as the dotted black line.

6.4. Comparison of wake profiles

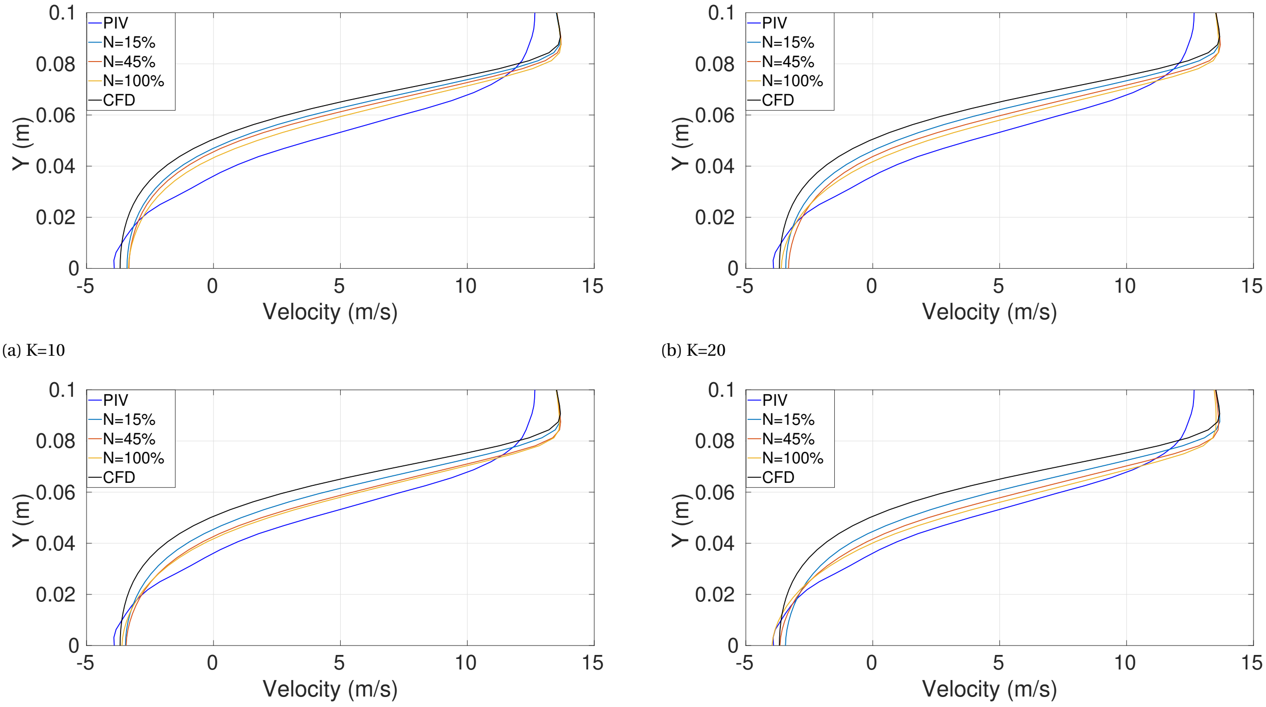
The performance of varying the different parameters of the DA is assessed by analyzing the wake profiles at a distance of 0.25 X/D downstream and a height of 0.5 Z/D from the ground. The results from the different parameters are shown in figure 6.15 and are shown only for one side (Y/D=0 to Y/D=1) as the other side is symmetric. The profiles are shown for three cases of N(15,45 and 100 %) for the sake of clarity, but the remaining profiles also follow the same trend. From the profiles it can be seen that the DA simulations approach the PIV velocity. Increasing the values of either of the parameters brings the profile closer to the experimental result.



(a) RMS error for different values of K

(b) RMS error for different values of K

Figure 6.14: RMS error for different values of N and K for the 3-D case. The back dotted line represents the rms error for the baseline case.



(a) K=10

(b) K=20

(c) K=30

(d) K=50

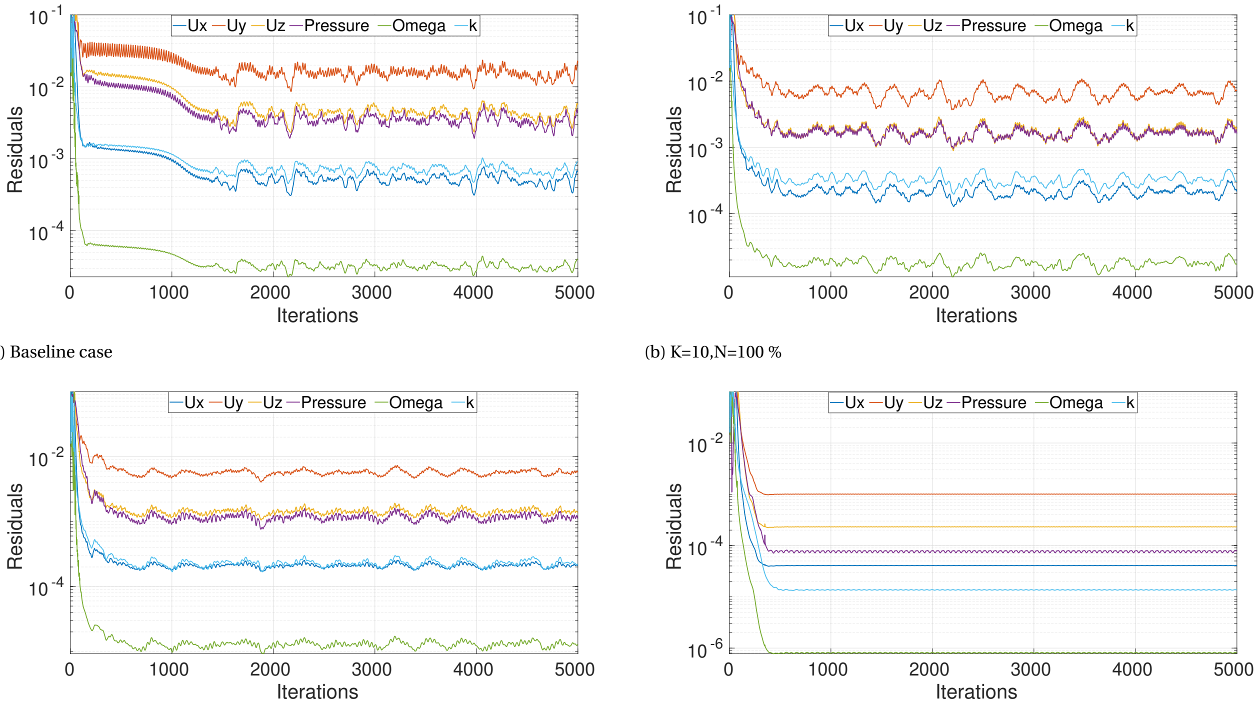
Figure 6.15: Wake streamwise velociyty profiles for different values of N and K. Each individual plot is for a single K.

6.5. Convergence analysis

Forcing affects the convergence characteristics of the DA simulations. All the simulations are run for 5000 iterations. An example of the convergence plots for a few of the simulations are shown in figure 6.16. In the baseline simulation, it takes around 1300 iterations for the solution to stabilize and even then, the residuals for spanwise velocity and pressure are quite high. As the forcing increases, the time taken for the solution to stabilize reduces and the value of the residuals also drop. Between the baseline simulation and the one with the highest forcing, the residuals drop by almost two orders of magnitude. The oscillation of the residuals also decreases with increased forcing.

6.6. Conclusions

The DA has succesfully been implemented onto a 3-D case using experimental data despite the ground truth comprising velocity that is not divergence free. It is seen that with sufficiently high forcing, the velocity in the CFD simulations approaches that of the ground truth and in the process is able to better recreate with



(a) Baseline case

(b) K=10,N=100 %

(c) K=30,N=45 %

(d) K=50,N=100 %

Figure 6.16: Residual plots for different simulations.

reasonable accuracy, some of the topological features of the experimental result as compared to a simulation without forcing. Furthermore, forcing helps converge the solution faster and also decreases the residuals as compared to a simulation with no forcing.

7 Conclusions and Recommendations

In this chapter, a brief summary of the motivation for the project, the methodology applied and the results obtained are provided followed by some recommendations for future work.

7.1. Conclusions

The integration of computational and experimental fluid dynamics is a subject that is widely researched for its promise of compensating mutual weaknesses in both domains by leveraging their individual advantages. Out of the many methods of DA outlined in section 2.2, the state observer method is the simplest in terms of implementation and computational cost but has not been studied in great detail. Previous studies of the state observer method have not followed a consistent approach to forcing techniques and have applied the method only on simplified 2-D models (Nisugi et al. [33],Suzuki et al. [51]) or in cases where it has been applied on 3-D, the ground truth velocities have been taken from other numerical experiments (Hayase and Hayashi [23], Imagawa and Hayase [26]). Furthermore, they have all only applied the method for unsteady flows. The present work has applied the state observer method for a time averaged flow around a 3-D bluff body and simulations have been performed using a steady RANS solver. Particular attention has been paid to the method in which the forcing term is implemented into the governing equations and how the forcing is applied on the entire control volume. The main research objective as stated previously is:

> *Development and assessment of a state-observer based data assimilation method for steady Reynolds averaged Navier-Stokes simulation using a complex 3-dimensional flow around a bluff body as a test case.*

To fulfill this objective, the present work has been carried out in different steps. First, the equations used for the method and its implementation are discussed in chapter 3. The method works on the principle of proportional feedback in the momentum equation. In terms of implementation, two different methodologies for forcing are used. The first is random forcing, where points in the control volume are chosen randomly and forcing is applied only at those points and the second method is volume forcing, where a complete region of the domain is forced. All simulations are performed using the open source CFD solver OpenFOAM v1706.

Chapter 4 deals with the application of the state observer method to 1-D and 2-D cases. From the 1-D case (the viscous Burgers equation), an understanding of the physical mechanisms involved while forcing are garnered and the effect of forcing at different control volumes (N) and strength of feedback term (K) are studied. The same analysis is carried out for the 2-D case (steady flow over a half cylinder). Both analysis showed that the error as both N and K are varied reduced asymptotically and there is always a residual error even when the full domain is forced at the highest forcing possible to maintain stability of the solution. The rate of convergence increased for both the 1-D and 2-D cases.

To apply the method to 3-D steady flows, an experiment was conducted in a wind tunnel to capture the time averaged flow over a simplified car side mirror model using the robotic volumetric velocimetry technique with HFSB as tracer particles. The velocities are averaged onto a grid which has a grid size of 3.75 mm. Key topological features from the experiment including the location of the reattachment point and the different vortical structures over the body are observed. DA is then applied using the random forcing method onto a 3-D computational grid. Despite the experimental data not being divergence free, application of the method did not result in any divergence or issues in continuity. The resultant velocity fields are forced to satisfy divergence and also approach the velocity fields specified by the experiment. A very similar behaviour to the 1-D and 2-D cases in terms of reduction in error is observed as both N and K are increased. Also, the main topological features were better represented when DA was applied as opposed to a baseline case where no forcing was applied. The error in position of the reattachment point was reduced by more than 50 %. The horseshoe vortices found from the experiment were not replicated in DA due to the fact that forcing was not applied close to the walls which is where the horseshoe vortices exist. The shape of the wake and the tail vortices were able to approach the structures found in the experiment. The 3-D simulation converged successfully for all values of K applied and no stability issues arose predominantly due to the fact that the experimental data existed only in a small region of the entire computational domain. Applying DA also showed better convergence characteristics both in terms of the value of the residual and the rate of convergence.

Overall, the working mechanisms of the state observer based technique were understood and it was successfully applied to three different cases which all showed successful reduction in error when forcing using a ground truth were applied. The main topological features from the experiment were replicated with reasonable accuracy wherever forcing was applied.

7.2. Recommendations

On conclusion of the present work, many avenues of improvement can be investigated. First, an extension of the method by adding an integral term to the present construction of the feedback term can be studied. With only a proportional term, the solution attempts to reach an equilibrium of sorts between its original solution and the ground truth and as the error decreases, the magnitude of the forcing term decreases. Furthermore, to reduce the error to a minimum, the forcing term and the number of forcing points should be increased to their maximum possible value, creating issues of stability. An integral term, like in a PID controller, would ensure that the magnitude of the forcing term does not decrease as the solution progresses, and also reduce the risk of going into stability issues as the forcing term gradually builds up to its maximum value. This would also permit the use of even fewer points to achieve the same reduction in error as compared to a proportional term.

In engineering applications, fluid interaction with a surface influences development of the flow. Capturing of velocity near no-slip boundaries can improve the results of DA significantly. However, resolution of the flow up to the wall is a challenge in PIV and could increase the complexity of the experiment. A possible solution is the use of pressure taps on a model to get information at the boundary and using that as an input for DA.

While this method has proven to improve both numerical and experimental data, it can still be a cumbersome task to perform experiments using a complex setup and then perform simulations. However, this work lays the foundation for the application of the steady state observer method in the hope that further improvements in the method will lead to the point where minimal information from an experiment can be used to generate accurate flowfield information in large domain domains using minimal computational effort.

A

Appendix A

This appendix contains the code used in the file main.C which contains the implementation of the state observer. This file calls the other requisite functions to run the SIMPLE solver. The main code also calls the header files UAvg.H and DataAssimilationConstantReader.H which contain the code for calculating the field average of velocity and the different constants used throughout the different files respectively. The main code calculates the forcing term EVol which is added to the momentum predictor (sectionA.2).

A.1. Main Code

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2016 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

Application
    DasimpleFoam

Group
    grpIncompressibleSolvers

Description
    Steady-state solver for incompressible flows with turbulence modelling.
    \heading Solver details
    The solver uses the SIMPLE algorithm to solve the continuity equation:

        \f[
            \div \vec{U} = 0
        \f]

    and momentum equation:

        \f[
            \div \left( \vec{U} \vec{U} \right) - \div \gvec{R}
          = - \grad p + \vec{S}_U
        \f]

    Where:
    \vartable
        \vec{U} | Velocity
        p       | Pressure
        \vec{R} | Stress tensor
        \vec{S}_U | Momentum source
    \endvartable

    \heading Required fields
    \plaintable
        U       | Velocity [m/s]
        p       | Kinematic pressure, p/rho [m2/s2]
        \<turbulence fields\> | As required by user selection
    \endplaintable

\*---------------------------------------------------------------------------*/

#include "fvCFD.H"
#include "singlePhaseTransportModel.H"
#include "turbulentTransportModel.H"
#include "simpleControl.H"
#include "fvOptions.H"
#include "wallDist.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

int main(int argc, char *argv[])
{
    #include "postProcess.H"

    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"
    #include "createControl.H"
    #include "createFields.H"
    #include "createFvOptions.H"
    #include "initContinuityErrs.H"

    turbulence->validate();

    // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

```
dimensionedScalar onemeter
(
"onemeter",
dimensionSet(0,1,0,0,0,0,0),
scalar(1.0)
);

Info<< "\nStarting␣time␣loop\n" << endl;
y = wallDist(mesh).y();

// Reading the DataAssimilation constants
#include "dataAssimilationConstantReader.H"

while (simple.loop())
{
    Info<< "Time␣=␣" << runTime.timeName() << nl << endl;

    // Reading the DataAssimilation constants
    #include "dataAssimilationConstantReader.H"
    if(runTime.value()>startIterAvg_)
    {
    #include "UAverage.H"
    }

    if(runTime.value()>startIter_)
   {

    // Calculation of the velocity differece E
    E = (UStar - U)*mag(UStar-U);

    // Calculation of the differential term
    forAll(EVol, cellI)
    {
        if ((UStar[cellI].component(0) == 0.0) || (y[cellI] < wallDist_))
        {
                EVol[cellI].component(0) = 0;
                EVol[cellI].component(1) = 0;
                EVol[cellI].component(2) = 0;
        }
        else
        {
            EVol[cellI].component(0) = kDADiff_/onemeter.value()*E[cellI].component(0);
            EVol[cellI].component(1) = kDADiff_/onemeter.value()*E[cellI].component(1);
            EVol[cellI].component(2) = kDADiff_/onemeter.value()*E[cellI].component(2);

        }
    }


   }
    // --- Pressure-velocity SIMPLE corrector
    {
        #include "DaUEqn.H"
        #include "pEqn.H"
    }
```

```
        laminarTransport.correct();
        turbulence->correct();

        runTime.write();

        Info<< "ExecutionTime␣=␣" << runTime.elapsedCpuTime() << "␣s"
            << "␣␣ClockTime␣=␣" << runTime.elapsedClockTime() << "␣s"
            << nl << endl;
    }

    Info<< "End\n" << endl;

    return 0;
}


// ************************************************************************* //
```

A.2. Momentum Predictor

The main function calls upon the momentum predictor DaUEqn.H which is where the EVol term is added to the momentum equation. This is given below:

```
    // Momentum predictor

    MRF.correctBoundaryVelocity(U);

    tmp<fvVectorMatrix> tUEqn
    (
        fvm::div(phi, U)
      + MRF.DDt(U)
      + turbulence->divDevReff(U)
     ==
        fvOptions(U)
    );
    fvVectorMatrix& UEqn = tUEqn.ref();

    UEqn.relax();

    fvOptions.constrain(UEqn);

    if (simple.momentumPredictor())
    {
        solve(UEqn == -fvc::grad(p) + EVol);

        fvOptions.correct(U);
    }
```

Bibliography

[1] W-tunnel, delft university of technology, https://www.tudelft.nl/lr/organisatie/afdelingen/aerodynamics-wind-energy-flight-performance-and-propulsion/facilities/low-speed-wind-tunnels/w-tunnel/.

[2] Ronald J Adrian. Scattering particle characteristics and their effect on pulsed laser measurements of fluid flow: speckle velocimetry vs particle image velocimetry. *Applied optics*, 23(11):1690–1691, 1984.

[3] Ronald J Adrian. Twenty years of particle image velocimetry. *Experiments in fluids*, 39(2):159–169, 2005.

[4] Nereida Agüera, Gioacchino Cafiero, Tommaso Astarita, and Stefano Discetti. Ensemble 3d ptv for high resolution turbulent statistics. *Measurement Science and Technology*, 27(12):124011, 2016.

[5] Iliass Azijli and Richard P Dwight. Solenoidal filtering of volumetric velocity measurements using gaussian process regression. *Experiments in Fluids*, 56(11):198, 2015.

[6] J Bardina, P Huang, T Coakley, J Bardina, P Huang, and T Coakley. Turbulence modeling validation. In *28th Fluid dynamics conference*, page 2121, 1997.

[7] Peter Bradshaw and George P Huang. The law of the wall in turbulent flow. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 451(1941):165–188, 1995.

[8] Johannes Martinus Burgers. A mathematical model illustrating the theory of turbulence. In *Advances in applied mechanics*, volume 1, pages 171–199. Elsevier, 1948.

[9] Julián Simón Calero. *The genesis of fluid mechanics 1640-1780*, volume 22. Springer Science & Business Media, 2008.

[10] Giuseppe Carlo Alp Caridi, Daniele Ragni, Andrea Sciacchitano, and Fulvio Scarano. Hfsb-seeding for large-scale tomographic piv in wind tunnels. *Experiments in Fluids*, 57(12):190, 2016.

[11] Pietro Catalano and Marcello Amato. An evaluation of rans turbulence modelling for aerodynamic applications. *Aerospace science and Technology*, 7(7):493–509, 2003.

[12] Mattias Chevalier, Jérôme Hœpffner, Thomas R Bewley, and Dan S Henningson. State estimation in wall-bounded flow systems. part 2. turbulent flows. *Journal of Fluid Mechanics*, 552:167–187, 2006.

[13] Donald Coles. The law of the wake in the turbulent boundary layer. *Journal of Fluid Mechanics*, 1(2): 191–226, 1956.

[14] Roger Daley. *Atmospheric data analysis*. Number 2. Cambridge university press, 1993.

[15] Charitha M de Silva, Jimmy Philip, and Ivan Marusic. Minimization of divergence error in volumetric velocity measurements and implications for turbulence statistics. *Experiments in fluids*, 54(7):1557, 2013.

[16] Eugene De Villiers. The potential of large eddy simulation for the modeling of wall bounded flows. *Imperial College of Science, Technology and Medicine, London (UK)*, 2006.

[17] AHP Derksen. Numerical simulation of a forced-and freely-vibrating cylinder at supercritical reynolds numbers. 2019.

[18] George Ellis. *Observers in control systems: a practical guide*. Elsevier, 2002.

[19] Gerrit E Elsinga, Fulvio Scarano, Bernhard Wieneke, and Bas W van Oudheusden. Tomographic particle image velocimetry. *Experiments in fluids*, 41(6):933–947, 2006.

[20] David Engler Faleiros, Marthijn Tuinstra, Andrea Sciacchitano, and Fulvio Scarano. Generation and control of helium-filled soap bubbles for piv. *Experiments in Fluids*, 60(3):40, 2019.

[21] Dimitry PG Foures, Nicolas Dovetta, Denis Sipp, and Peter J Schmid. A data-assimilation method for reynolds-averaged navier–stokes-driven mean flow reconstruction. *Journal of Fluid Mechanics*, 759:404–431, 2014.

[22] Toshiyuki Hayase. Numerical simulation of real-world flows. *Fluid Dynamics Research*, 47(5):051201, 2015.

[23] Toshiyuki Hayase and Satoru Hayashi. State estimator of flow as an integrated computational method with the feedback of online experimental measurement. *Journal of fluids engineering*, 119(4):814–822, 1997.

[24] Gabor T Herman and Arnold Lent. Iterative reconstruction algorithms. *Computers in biology and medicine*, 6(4):273–294, 1976.

[25] Jérôme Hœpffner, Mattias Chevalier, Thomas R Bewley, and Dan S Henningson. State estimation in wall-bounded flow systems. part 1. perturbed laminar flows. *Journal of Fluid Mechanics*, 534:263–294, 2005.

[26] Kentaro Imagawa and Toshiyuki Hayase. Numerical experiment of measurement-integrated simulation to reproduce turbulent flows with feedback loop to dynamically compensate the solution using real flow information. *Computers & Fluids*, 39(9):1439–1450, 2010.

[27] Antony Jameson and Luigi Martinelli. Mesh refinement and modeling errors in flow simulation. *AIAA journal*, 36(5):676–686, 1998.

[28] Constantin Jux. Robotic volumetric particle tracking velocimetry by coaxial imaging and illumination. 2017.

[29] Constantin Jux, Andrea Sciacchitano, Jan FG Schneiders, and Fulvio Scarano. Robotic volumetric piv of a full-scale cyclist. *Experiments in Fluids*, 59(4):74, 2018.

[30] Eugenia Kalnay. *Atmospheric modeling, data assimilation and predictability.* Cambridge university press, 2003.

[31] Tomislav Maric, Jens Hopken, and Kyle Mooney. The openfoam technology primer. 2014.

[32] A Melling. Tracer particles and seeding for particle image velocimetry. *Measurement Science and Technology*, 8(12):1406, 1997.

[33] Keisuke Nisugi, Toshiyuki Hayase, and Atsushi Shirai. Fundamental study of hybrid wind tunnel integrating numerical simulation and experiment in analysis of flow field. *JSME International Journal Series B Fluids and Thermal Engineering*, 47(3):593–604, 2004.

[34] Suhas Patankar. *Numerical heat transfer and fluid flow.* CRC press, 1980.

[35] Kaustubh Patil. Surface pressure measurements using coaxial volumetric velocimetry. 2019.

[36] Yves Pomeau, Martine Le Berre, Philippe Guyenne, and Stephan Grilli. Wave-breaking and generic singularities of nonlinear hyperbolic equations. *Nonlinearity*, 21(5):T61, 2008.

[37] Markus Raffel, Christian E Willert, Fulvio Scarano, Christian J Kähler, Steve T Wereley, and Jürgen Kompenhans. *Particle image velocimetry: a practical guide.* Springer, 2018.

[38] William C Reynolds. The potential and limitations of direct and large eddy simulations. In *Whither turbulence? Turbulence at the crossroads*, pages 313–343. Springer, 1990.

[39] Lewis Fry Richardson. *Weather prediction by numerical process.* Cambridge University Press, 2007.

[40] Fulvio Scarano and Peter Moore. An advection-based model to increase the temporal resolution of piv time series. *Experiments in fluids*, 52(4):919–933, 2012.

[41] Fulvio Scarano, Sina Ghaemi, Giuseppe Carlo Alp Caridi, Johannes Bosbach, Uwe Dierksheide, and Andrea Sciacchitano. On the use of helium-filled soap bubbles for large-scale tomographic piv in wind tunnel experiments. *Experiments in Fluids*, 56(2):42, 2015.

[42] Daniel Schanz, Sebastian Gesemann, Andreas Schröder, Bernhard Wieneke, and Matteo Novara. Non-uniform optical transfer functions in particle imaging: calibration and application to tomographic reconstruction. *Measurement Science and Technology*, 24(2):024009, 2012.

[43] Daniel Schanz, Sebastian Gesemann, and Andreas Schröder. Shake-the-box: Lagrangian particle tracking at high particle image densities. *Experiments in fluids*, 57(5):70, 2016.

[44] Jan FG Schneiders, Fulvio Scarano, Constantin Jux, and Andrea Sciacchitano. Coaxial volumetric velocimetry. *Measurement Science and Technology*, 29(6):065201, 2018.

[45] Andrea Sciacchitano and Fulvio Scarano. Elimination of piv light reflections via a temporal high pass filter. *Measurement Science and Technology*, 25(8):084009, 2014.

[46] Andrea Sciacchitano, Richard P Dwight, and Fulvio Scarano. Navier–stokes simulations in gappy piv data. *Experiments in fluids*, 53(5):1421–1435, 2012.

[47] Steven M Soloff, Ronald J Adrian, and Zi-Chao Liu. Distortion compensation for generalized stereoscopic particle image velocimetry. *Measurement science and technology*, 8(12):1441, 1997.

[48] RM Stringer, J Zang, and AJ Hillis. Unsteady rans computations of flow around a circular cylinder for a wide range of reynolds numbers. *Ocean Engineering*, 87:1–9, 2014.

[49] Takao Suzuki. Reduced-order kalman-filtered hybrid simulation combining particle tracking velocimetry and direct numerical simulation. *Journal of Fluid Mechanics*, 709:249–288, 2012.

[50] Takao Suzuki. Pod-based reduced-order hybrid simulation using the data-driven transfer function with time-resolved ptv feedback. *Experiments in fluids*, 55(8):1798, 2014.

[51] Takao Suzuki, Hui Ji, and Fujio Yamamoto. Unsteady ptv velocity field past an airfoil solved with dns: Part 1. algorithm of hybrid simulation and hybrid velocity field at re= 10 3. *Experiments in fluids*, 47(6): 957, 2009.

[52] Olivier Talagrand. Assimilation of observations, an introduction (gtspecial issueltdata assimilation in meteology and oceanography: Theory and practice). *Journal of the Meteorological Society of Japan. Ser. II*, 75(1B):191–209, 1997.

[53] Edward Tinoco. Validation and minimizing cfd uncertainty for commercial aircraft applications. In *26th AIAA Applied Aerodynamics Conference*, page 6902, 2008.

[54] BW Van Oudheusden. Piv-based pressure measurement. *Measurement Science and Technology*, 24(3): 032001, 2013.

[55] HK Versteeg and W Malalasekera. Computational fluid dynamics. *The finite volume method*, 1995.

[56] J Westerweel. Efficient detection of spurious vectors in particle image velocimetry data. *Experiments in Fluids*, 16(3-4):236–247, 1994.

[57] B Wieneke. Volume self-calibration for 3d particle image velocimetry. *Experiments in fluids*, 45(4):549–556, 2008.

[58] Bernhard Wieneke. Iterative reconstruction of volumetric particle distribution. *Measurement Science and Technology*, 24(2):024008, 2012.

[59] BFA Wieneke. *PIV Uncertainty Quantification and Beyond.* PhD thesis, Delft University of Technology, 2017.