Using Data Assimilation Techniques to Validate Fluid Flow Models

Godwin Jeffrey Madho

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

The University of Leeds
EPSRC Centre for Doctoral Training in Fluid Dynamics
January 2020

Declaration

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

For

Franklin Madho, John Madho, Hendry Paul, Lilly Paul and
Edwin Peter

Abstract

Describing turbulent fluid flows remains challenging, despite massive computational resources being devoted to it. This is because turbulent fluids interact on a vast range of spatial and temporal scales and the dynamics is sensitive to small disturbances. In particular, prediction of such systems remains difficult. In many cases, though the models are imperfect, more faithful predictions can be achieved by combining the models with observations of the system. This observational knowledge can then be assimilated into the model to provide better predictive capability, and also to estimate the parameters of an uncertain model. This study aims to test the capabilities of one such assimilation scheme, the Ensemble Kalman Filter (EnKF), in aiding the prediction of the complex behaviour in a thermal rotating annulus.

Four results are presented in this thesis. First, EnKF was applied to the Lorenz model, where tests were performed with varying ensemble sizes, inflation, and gaps between data assimilation. This was to understand the limitations of EnKF on a smaller chaotic system. It was determined that only a small ensemble size of just 7 was required to accurately predict model behaviour. Tests were also done on EnKF capabilities on parameter estimation where the system predicted the values of the parameters very accurately.

Second, the experimental results are presented for thermal rotating annulus experiments which were done at a velocity of 1 rad/s, 2.5 rad/s and 3 rad/s. The experiments showed behaviour that is expected at the various rotation rates. We found that it was difficult to get accurate results for behaviour at the lower levels of the annulus compared to data at higher levels. Overall the one size fits all approach to the PIV settings for analysing images does not seem to work well and future studies will have to fine tune the setting for more accurate analysis.

Third, a twin experiment where EnKF was performed using the MORALS code with a high resolution 'truth' and low resolution ensemble. Studies were done at 1 rad/s, 2 rad/s, 2.5 rad/s and 3 rad/s. At 1 rad/s, EnKF did a good job of tracking the 'truth' at different settings. Looking at 2 rad/s, in most cases the low resolution ensemble system struggled to replicate the m=4 wavenumber observed in the 'truth' and stayed at m=3 wavenumber. Although decreasing the DA length to every 1 minute gave the best result, in many cases the system strayed away from the truth. Going further to 2.5 rad/s, the structures observed at this rotation rate were very volatile with the system finding it hard to track the truth. This was the case even when the time between DA was decreased or the ensemble size increased. Lastly at 3 rad/s the low resolution models in the ensemble always gave the wrong wavenumber and was not able to track the truth. It seems that the resolution for the ensemble needs to be increased for better simulation at higher rotation rates.

Finally, the results for using low resolution ensemble MORALS to predict experimental results were presented where tests were done at 1 rad/s and 2.5 rad/s. At 1rad/s the ensemble can replicate the fluid structures observed in the PIV data with an m=3

wave but the ensemble was never able to tack the ‘truth’ with the latter travelling around the annulus faster. At a higher rotation rate of 2.5rad/s the ensemble is never able to replicate the m=4 wave as observed in the ‘truth’. This is even the case when the θ resolution is increased to 128 points.

From the results presented in this thesis EnKF can be seen as a reasonable solution to help predict behaviour in a thermal rotating annulus setting. It does a good job of giving close results to the truth in most of the twin studies. And although there were problems when using the experimental results there should be improvement in the results when better analysis techniques are used to obtain better observations. EnKF overall has problems at higher rotation rates this problem can also be overcome using different settings in the ensemble and EnKF.

Acknowledgements

I would like to thank my supervisors Professors Steve Tobias, Professor Chris Jones and Dr Sven Van Loo from the University of Leeds and last but not least my industrial supervisor Dr Wayne Arter from the Culham Centre for Fusion Energy (CCFE) for all their advice during this project. I would like to especially thank Steve and Chris for all their feedback during the writing of this thesis.

I would also like to thank Professor Peter Read and Dr Roland Young at the University of Oxford for all their help with the MORALS code and thermal rotating experiments. Thanks to Gareth Keevil at School of Earth and Environment, University of Leeds for help with DynamicStudio. The python script used for controlling the lights and image acquisition on the experimental apparatus was written by Dr Hélène Scolan at the AOPP. All experiment were done with the help of Cheng Qian also at AOPP.

I would like to thank EPSRC for funding the PhD without which this research would not be possible.

I would like to thank my friends who have helped through the years to achieve this goal. This includes both the Physics gang who I have known for the past 10 years and the many new friends I have made through years at the CDT.

I would finally like to thank my family who have supported me throughout the years in my studies. Especially my mother Sanchita Madho for all the hard work she has done to make sure I could do this.

Abbreviations

Definition of symbols used in Chapter 3

a	inner radius of annulus
b	outer radius of annulus
d	depth of annulus
A	advection operator
D	diffusion operator
g	acceleration due to gravity
i	radial grid index
I	number of grid points in the radial direction
j	azimuthal grid index
J	number of grid points in the azimuthal direction
k	vertical grid index
K	number of grid points in the vertical direction
m	time index
r	radial coordinate
t	time
T	temperature
T_0	reference temperature
(u, v, w)	velocity components (r, θ, z)
z	vertical coordinates
α	coefficient of expansion
δ	finite difference derivative
θ	azimuthal coordinate
κ	coefficient of thermal diffusion
ν	coefficient of kinematic viscosity
Π	pressure
$\Delta\rho$	density perturbation
ρ_0	mean density
σ	buoyancy

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Numerical weather forecasting remains challenging despite the fact that massive computational resources are devoted to it. This is because predicting chaotic behaviour has always been difficult. In many cases simplified models can be used to understand the nature of chaotic behaviour, one of the most famous examples being that of Lorenz (1963), this understanding can then be used to derive more complicated models. Many fluid flow systems show chaotic behaviour which have some similarities to that seen in these simplified models.

A key feature of chaotic systems is that they are very sensitive to the initial conditions imposed. Initial conditions in many studies are determined by setting the model to agree as closely as possible with observations. But there is always uncertainty about the true initial conditions, as observations are never perfect and contain a range of possible errors. Sensitivity to initial conditions means that even if these conditions are set as accurately as possible, the model evolution will eventually diverge from the true behaviour. This sensitivity to initial conditions is compounded by the facts that models don't incorporate the full complexity of the real life behaviour and that the parameters used are not entirely accurate. Normally to get more accurate predictions the study has to either use a more complicated model or make the computational parameters more accurate, for example by increasing resolution and decreasing the time step. Both situations result in increased computation time and use of resources. Due to the nonlinear and chaotic nature of these models, even two very accurate models running with the same parameters will not necessarily give the same results.

A way to circumvent this problem is to use data assimilation to correct the numerical models when the models diverge from the 'truth'. A numerical model can be run as normal, but at a certain point, defined by the user or suggested by the error in the model, a reading is taken of the true state of the behaviour, for example using experiments or observations to collect data. These observed values are used to correct variables in the model, which then continues to run. The act of combining the model and observational results to get new initial conditions for the model is called data assimilation. In many of the various methods, that will be discussed later, weights are applied to calculate the new initial conditions. Data assimilation can, at least in principle, overcome the difficulty of chaotic behaviour provided the observations can be made continually and frequently.

1.1 Aim

This study was originally motivated by the need to predict the behaviour of rotating plasma in a tokamak in real time with the aid of data assimilation (Wehner et al., 2011). However, at an early stage it was decided to focus on how DA will be implemented for such a problem and what will be it's limitations. This will be done using simpler models than models currently available for tokamaks. So to obtain this goal many smaller steps will be taken throughout the study. First, a study is done using the Ensemble Kalman filter (EnKF) and the Lorenz model to test its predictive power and to understand how EnKF works. This data assimilation technique will be then applied to the hydrodynamic problem known as the thermal rotating annulus model (Hide, 2010; Young and Read, 2013). This set will provide a good test be to understand how to apply EnKF to a 3D system and learn what modifications need to be made to the vanilla EnKF to get more accurate predictions. Steps can then be taken to apply EnKF to a magnetohydrodynamic model for a tokamak.

Planetary atmospheres and tokamaks exhibit chaotic flow, and there are numerical models that describe them, but to achieve any reasonable degree of accuracy these models require massive computational resources for even a single run. Doing the number of runs required to test data assimilation thoroughly would simply not be feasible. Several systems of low order ODEs exhibit chaotic behaviour (e.g. the Lorenz system) and it is feasible to do extensive data assimilation tests on these systems. But these systems do not model any physical system with sufficient accuracy to compare experimental data against them with any confidence.

The advantages of the rotating annulus model are that firstly, it has been extensively studied experimentally (Vettin, 1857; Fultz et al., 1959; Hide, 1958; Hignett, 1985), so its behaviour is well known. We can, therefore, have a good idea as to whether our data assimilation predictions are working well or not. Second, it is amenable to numerical simulation without the need for excessive computational resources. It has been known for some time (Hignett et al., 1985; Read et al., 1997; Harlander et al., 2011; Vincze, Borchert, Achatz, von Larcher, Baumann, Hertel, Remmler, Beck, Alexandrov, Egbers, Froehlich, Heuveline, Hickel and Harlander, 2014) that numerical simulations of the full three-dimensional annulus problem are in good agreement with data from the experiment itself. The point about these simulations not requiring an excessive computational load is particularly important. Testing data assimilation techniques requires a great many computational runs as many aspects can be varied, for example, intervals between data assimilations, amount of observed data, ensemble size and the magnitude of any inflation imposed on the system. The third major advantage of the annulus model is that it exhibits chaotic behaviour, just as many geophysical fluid systems do. Indeed, the baroclinic waves that are seen in the annulus experiment have similarities to the baroclinic waves that dominate mid-latitude weather systems (Hide, 1958; Ravela et al., 2010; Hide, 2010). A fourth advantage of the annulus model that became apparent in the course of this study is that by varying the rotation rate of the annulus (this is easy to do experimentally and numerically) we can vary the behaviour of the working fluid from mildly chaotic at lower rotation rates to strongly chaotic at higher rotation rates. This is a very desirable property of a model for data assimilation studies, as it enables us to see what needs to be done to maintain effective data assimilation as a system becomes more chaotic.

The numerical code used to simulate a thermal rotating annulus experiment is called the Meteorological Office/Oxford Rotating Annulus Laboratory Simulation (MORALS) code, which is a code maintained by the Atmospheric, Oceanic and Planetary Physics group (AOPP) at the University of Oxford. This code has been used previously to study rotating annulus behaviour as published by Hignett et al. (1985); Read et al. (1997); Young and Read (2013).

This code was used in two different ways in the study presented. First, a procedure known as a twin experiment (Reichle et al., 2002; Burgers et al., 2002) was used in which a high resolution run is regarded as the truth (quite reasonably given the known agreement between high resolution runs and the actual experiment) and low resolution runs are used as the model for data assimilation. The aim is to see whether the low resolution model can be kept in approximate agreement, as time evolves, with the high resolution 'truth' if DA is regularly used on the low resolution models. Second, actual experimental data from the AOPP thermal annulus experiment is used as the truth to see whether the numerical model can give agreement with this truth when data assimilation is used.

1.2 Data Assimilation

The act of combining the model and observational results to get new conditions for the model is called data assimilation (DA). Generally used to obtain a better prediction and reduce the computational cost, data assimilation is also used to tackle other interesting questions such as parameter estimation/optimisation, looking at how many observations are needed for accurate prediction and looking at where the best possible place is for observations. Initially developed for numerical weather prediction, data assimilation is now used in a wide variety of fields such as oceanography, traffic modelling, crime modelling, etc (Evensen, 2003).

As time has gone on many different methods of data assimilation have been created. Modern data assimilation can be divided into mainly three distinct methods. Firstly, variational methods which minimise a cost function. This is done by taking into account the size of the observations, the background and, the errors in the background and observation. This cost function gives the optimal analysis state (Sasaki, 1970; Lorenc et al., 2000; Rawlins et al., 2007). 3D-Var has been used extensively in many NWP problems as it gave the user the ability to do three dimensional data assimilation (Sasaki, 1970; Lorenc et al., 2000; Rawlins et al., 2007). Secondly, ensemble methods that use an ensemble of forecasts to perform data assimilation. Unlike like variational methods, the errors in the ensemble system evolve with time and are calculated using the ensemble data during the analysis. The Ensemble Kalman filter (EnKF) is one of the best examples of ensemble methods (Evensen, 1994; van Leeuwen, 2010; Evensen, 2003). Finally, there are hybrid methods which, as the name suggests, are a combination of different methods. These methods are very new and have been developed to try and keep the advantages of each of the individual methods in mind, while trying to replace their disadvantages with other techniques (Zhang et al., 2009; Bannister, 2017). The optimal method to use in a given situation depends on the model and type of observations available. The technique used for the studies presented in this thesis is the Ensemble Kalman Filter (EnKF) method. Chapter (4) describes how the EnKF works and how it is implemented numerically.

1.3 Rotating Fluids

Rotating fluid behaviour is observed in a wide variety of areas, from the atmosphere and oceans to centrifuges and tokamaks. The study of rotating fluids is therefore important when trying to understand these systems. In most cases, to understand the phenomenon a model (experimental or computational) is created which serves as a testbed for theories and measurement techniques. If an experimental study is not ideal then a numerical approach can also be considered. Most modern studies use both experimental and numerical approaches to study behaviour. With advances in computer technology, numerical studies have become the way to study phenomena such as the circulation of the atmosphere or the ocean.

The Meteorological Office/Oxford Rotating Annulus Laboratory Simulation code (MORALS), is a Fortran code maintained at the Atmospheric, Oceanic and Planetary Physics (AOPP) group at the University of Oxford. It is written in a mixture of Fortran 77 and Fortran 90 and is designed to run on Linux. A general outline of how MORALS operates is given in chapter (3), which describes the equations and the methods that are used to solve them. The chapter goes to explain how MORALS creates the stretched grid used in its model along with how the Arakawa type C grid (Arakawa and Lamb, 1977) is used to store data. It also describes the implementation of the MORALS code in 2D, the mapping of 2D data into 3D and how the code operates in 3D. More information about the MORALS code can be obtained on the website MORALS (2020).

1.4 Lorenz Model

The effectiveness of any Data Assimilation (DA) method can only be ascertained when tested against a model that pushes its limits to give accurate predictions. In many cases, the Lorenz model has been used to fill this niche as its nonlinear behaviour at particular parameters provides a robust testbed to look at the different aspects and limitations of a DA method (Evensen, 1997, 2009; Annan and Hargreaves, 2004).

The Lorenz model (Lorenz, 1963) is a set of three ordinary differential equations which give a strongly nonlinear system. The model was conceived to describe the flow in a thermal convection experiment but has since been used in numerous studies related to nonlinear systems. The system is known for its non-periodic/chaotic behaviour and unpredictability for certain parameter values.

In the study presented in chapter (5), the Ensemble Kalman Filter (EnKF) is tested using the Lorenz model. In this chapter, EnKF's capabilities will be tested using a variety of ensemble sizes, inflation and time between DA to see how the system can cope with the changes. It is also tested for parameter estimation to see if EnKF can predict what were the parameters used to create the 'truth'. The model has been previously used to test data assimilation techniques (Miller, 1994; Evensen, 1997). The model has also been used to conduct parameter estimation studies (Annan and Hargreaves, 2004).

1.5 Experiment

With a lack of computational power in the early days of weather prediction, laboratory experiments were used to replicate atmospheric conditions and to understand the fundamental physics that underlies atmospheric behaviour (Vettin, 1857; Fultz et al., 1959; Hide, 2010). As modern technology has made computational studies easier and more accurate, it has also made experiments more accurate, using modern apparatus. As such, experiments still play a very important role in understanding atmospheric behaviour and being used as a testbed for computational methods.

To generate real data to be used for DA, experiments were done at AOPP. Previous studies by Hignett et al. (1985); Read et al. (1997) performed at AOPP utilised an older experimental apparatus for a rotating thermal annulus, see Hignett et al. (1985) for more details on this older setup. In the DA study presented by Young and Read (2013) using this setup, the rotation rate was changed every 20 minutes. These experiments focused on how the behaviour of the fluid changed with the change in rotation rate (eg. wavenumber changes when increasing or decreasing rotation rates).

The results for the new experiments are presented in chapter 6 and focused on long runs at a single rotation rate. These runs were used to test the new apparatus, along with examining the long term behaviour of the fluid and to see how stable the waves formed at these rotation rates are. Experiments were done at 1rad/s, 2.5rad/s and 3rad/s. Particle image velocimetry (PIV) was done on the experiment to obtain data that was then analysed and plotted.

The results presented in this chapter will be used as the truth for DA done using EnKF along with the MORALS code serving as the ensemble model to predict the results. The results for this study is presented in chapter (8).

1.6 Twin Experiment

The process of DA is not as simple as just coupling real life observation and simulation, and then using it for prediction. The more complex the model, the more time is needed to fine tune the DA settings and parameters. In most cases, synthetic data is used for a twin experiment to evaluate DA settings and their sensitivity to different parameters (Reichle et al., 2002; Burgers et al., 2002).

Chapter (7) presents the results for a twin experiments which is carried out using EnKF and the MORALS code. The twin experiment is used to test different scenarios for EnKF with the aim to find suitable settings for the model and EnKF which can be applied for observations obtained via experiments to test MORALS' predictions. In particular experiments will be carried out at different rotation rates (1 rad/s, 2 rad/s, 2.5 rad/s and 3 rad/s) with different observational settings.

To conduct the twin experiment a high resolution MORALS model is first run to obtain data that should match closely with what is observed in experiments. This data will be used to create the 'truth' that will be used in DA. An ensemble of low resolution MORALS models will then be used to test the capabilities of EnKF. This twin experiment allows the testing of EnKF while knowing all the details/settings about the 'truth' and the ensembles. The twin experiment also makes it easier to line up observation points with the ensemble model when it comes to data assimilation. This provides an opportunity to identify how the parameters of EnKF can best be chosen for the different scenarios. This will help provide a baseline for EnKF's abilities when it comes to predicting and tracking observations.

1.7 Experiment and DA

After conducting a twin experiment to understand how EnKF method behaves under different conditions for MROALS, DA is done using the experimental data as the 'truth'. The results for this test is presented in chapter (8) with the aim to try and replicate or better the results obtained by Young and Read (2013).

Using EnKF on real life data provides new challenges for the method as now the amount and quality of the data used as the observations are subject to how the experiment was conducted and how the data was recorded. The error in the data is associated with the instruments that were used to conduct the experiments and the method that was used to process the data to obtain velocity vectors. The amount of data gathered is also subject to the resolution of the instruments used. The results for the experimental data set are discussed in chapter (6), along with how the data was obtained and processed. In this chapter, the processed results of the experiments were used without any further changes.

To predict the behaviour seen in the experiments, the Met Office/Oxford Rotating Annulus Laboratory Simulation (MORALS) code was used which solves the Navier-Stokes equations for a rotating annulus. This is the same code that was used in the twin experiment. More details on the working of the code can be found in chapter (3). In the study presented in this chapter, the experimental data were used as the truth and the MORALS code was used to create an ensemble of low resolution models which was used to predict the experimental results with the help of EnKF.

Chapter 2

The Thermal Rotating Annulus

Rotating fluid behaviour is observed in a wide variety of application, from the atmosphere and ocean circulation to centrifuges and tokamaks. The study of rotating fluids is therefore important when trying to understand these systems. In many cases a model, experimental or numerical, is created which serves as a testbed for theories and measurement techniques. If an experimental study is not ideal, then a numerical approach can be considered. Most modern studies use both experimental and numerical approaches to study rotating fluid behaviour. With advances in computer technology, numerical studies have become the usual way to study complex behaviour such as the circulation of the atmosphere or the ocean. These numerical models can be very complicated, so a simple experiment, such as the the thermal rotating annulus, can help understand some of the underlying behaviour of these more complex models.

An introduction to thermal rotating annulus is given in section (2.1), and the different variations of the annulus model that have been used for studies are referenced. Section (2.2) introduces some of the experiments that have been done using a thermal rotating annulus. Section (2.3) looks at studies that involved both a numerical code and laboratory experiments. Finally, section (2.4) describes some of the studies that used data assimilation on the thermal rotating annulus.

2.1 Introduction

There are a few different variations of the rotating annulus experiment that have been studied throughout the years. In the first case, a cylinder was filled with a liquid and was rotated (Taylor, 1921), and obstacles were moved through the fluid and the resulting fluid flow observed. In another annulus experiment, the gap between two concentric cylinders was filled with fluid, a configuration now known as Taylor–Couette flow (Taylor, 1923). In this case, both cylinders can be rotated independently to produce a range of different behaviours in the fluid flow. In both these cases, the annulus walls can be differentially

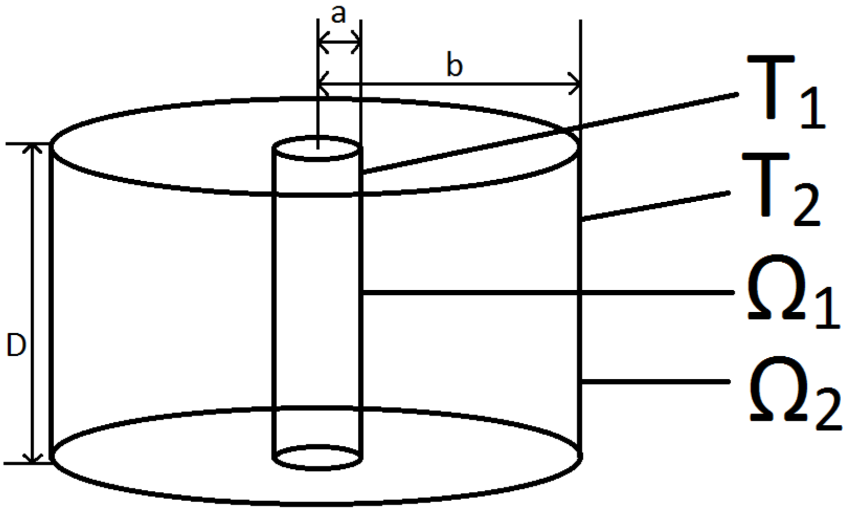


Figure 2.1: Study setup. a and b were the radii of the cylinders and D is the height of the system, Ω_1 and Ω_2 were the rotation velocities for the cylinders and T_1 and T_2 were the temperatures.

heated to add to the complexity of the studies (Vettin, 1857; Chandrasekhar, 1953). The system can be further complicated by using sloped surfaces or spherical geometry, but for the study presented here, these were ignored.

For the study presented in this thesis, a thermal rotating annulus was used as shown in figure (2.1). Here the inner and outer wall rotate at the same angular velocity (Ω) in the same direction. The outer and inner walls are kept at different temperatures(T_1 and T_2). In most cases, the inner wall is cooled and the outer wall is heated. This is to simulate a planetary setting, where the poles are cool and the equator is hot. The two cylinders have a radius a and b and both have the same height D. Depending on the rotation rate of the annulus, different behaviour can be observed in the working fluid.

Some common dimensionless parameters used in studying rotation fluids are the Taylor number, Ta,

$$Ta = \frac{4\Omega^2(b-a)}{\nu^2 D}, \tag{2.1}$$

Hide number (thermal Rossby number), H,

$$H = \frac{Dg\alpha\Delta T}{\Omega^2(b-a)^2}, \tag{2.2}$$

the Reynolds number, Re,

$$Re = \frac{u(b-a)}{\nu} \tag{2.3}$$

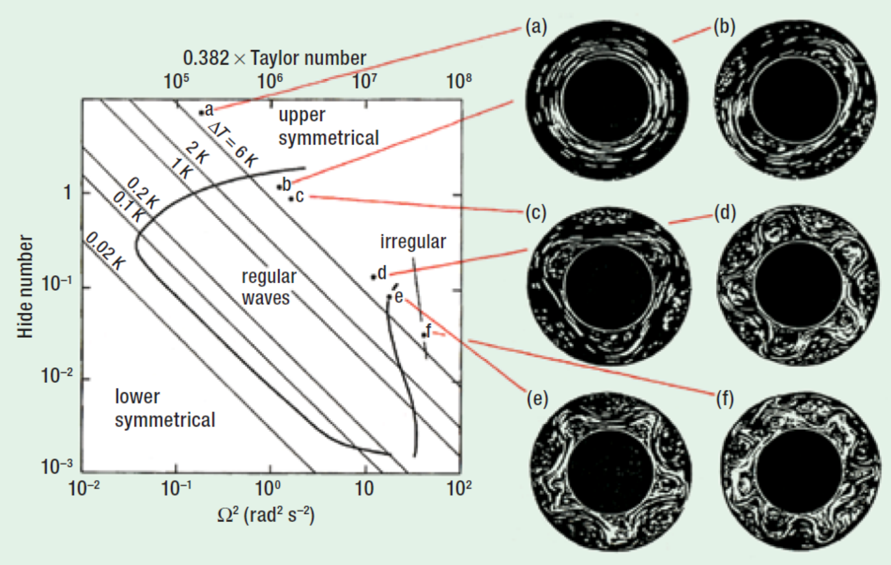


Figure 2.2: Schematic regime diagram for a thermal rotation annulus for different rotation rate (Hide, 2010).

and the Prandtl number, Pr,

$$Pr = \frac{\nu}{\kappa}. \tag{2.4}$$

Here a and b are the inner and outer radius of the annulus and D is the annulus depth. The rotation rate of the annulus is given by Ω, ΔT is the temperature difference between the inner and outer radius of the annulus. g is the acceleration due to gravity, α is the thermal expansion coefficient, ν is the kinematic viscosity and κ is the thermal diffusivity.

The waves that develop are a manifestation of a barolcinic instability and develop to transfer heat around the annulus (Read et al., 2015). The baroclinic instabilities that develop into a wave are often dominated by the dominant wave number and its harmonics. These waves can be steady, chaotic or quasi-periodic. The state obtained in an experiment depend on the initial condition. Further, hysteresis is observed in the system when observing transition between states when slowly changing external parameters (Hide and Mason, 1975; von Larcher and Egbers, 2005). Vacillation amplitude (AV) and structural (SV) are observed in experiments. In amplitude vacillation the amplitude of the wave is modulated as it interacts with the background zonal flow. The wave amplitude grows to a maximum before before decaying and repeating the cycle (see figure 2.3). Structural vacillation is observed when approaching the irregular flow regime. Here the structure of the waves tilt in horizontal and radial axis periodically in the wave peaks and troughs (Weng et al., 1986; Weng and Barcilon, 1987).

All modern studies are a combination of laboratory experiments and numerical simulations. Experiments are used to test accuracy of numerical models as they have an infinite resolution. But numerical models are more versatile as their parameters can easily be changed. A proven numerical model can be used to fill in the gaps for data that might be hard to collect in experiments. Numerical models can also be used to study

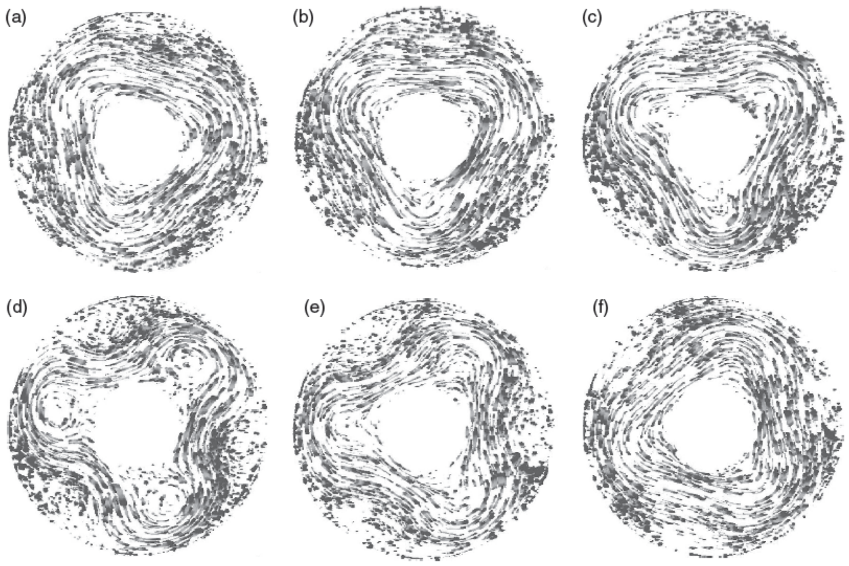


Figure 2.3: Horizontal flow fields during a typical amplitude vacillation cycle (Read et al., 2015).

parameters which can be difficult for experiments.

2.2 Experiments

Vettin (1857) is the first known laboratory experiment on general circulation using a thermal rotating annulus. In this study, ice was used at the centre to create a temperature gradient and air was used as the working fluid. Using smoke from a cigar, Hadley type circulation was observed. The modern setup for the thermal rotating annulus can be credited to Fultz et al. (1959) and Hide (1958). Here instead of air, a fluid such as water was used and the walls were kept at different temperatures using baths that contained heated or cooled water. More modern studies such as Hignett (1985) used a water-glycerol mix and kept the walls at constant temperature using water that was pumped at their respective temperatures next to their respective walls.

Some of the experiments using a thermal rotating annulus were made to study phenomena such as amplitude and structural vacillation. This is the modulation of the amplitude and the dominant wavenumber. Pfeffer and Fowlis (1968) was one of the first to report wave dispersion and structural vacillation showing the presence of $m = 4$ and $m = 5$ waves simultaneously as observed in figure 5 and 6 in their paper. Studies such as Hide and Mason (1975) and Hignett (1985) gave examples of amplitude vacillation. Hignett (1985) reported that amplitude vacillation is observed at $m = 2$, $m = 3$ and $m = 4$ wavenumbers when $\Delta T \gtrsim$ 2K. Früh and Read (1997) discussed how the interaction between dominant and weaker modes can lead to amplitude and structural vacillations. This was confirmed by Harlander et al. (2011).

2.3 Rotating Fluids and numerical studies

With the limitations of experimental measurement, and the increase in computational power, numerical models were created to simulate and study thermal rotating annulus behaviour. Hignett et al. (1985) made a comparison between experimental and numerical results and reported good agreement for major features in the horizontal flow field and temperature, but they did not study rotating rates higher than 2 rad/s. Read et al. (1997) also compared experimental results with the 3D numerical model, in the context of an internally heated rotating thermal annulus, the annulus being heated by making the working fluid electrically conducting and passing a current through it. They successfully simulated the structure and behaviour of the regular wave flow seen in their experiments. As the rotational velocity was increased, the numerical model displayed structural vacillation, as did the experiments, but these structures broke down more often into irregular flow than they did in the experiments.

Vincze, Borchert, Achatz, von Larcher, Baumann, Hertel, Remmler, Beck, Alexandrov, Egbers, Froehlich, Heuveline, Hickel and Harlander (2014) aimed to create a laboratory experiment which would be a testbed for numerical techniques and models aiming to investigate geophysical flows. Experiments were done at the fluid dynamics laboratory of the Brandenburg Technical University at Cottbus-Senftenberg (BTUCS). The experiments were done using rotational velocities in the range of $\Omega = 2.26$ rpm up to a $\Omega = 20.91$ rpm (0.2 - 2.2 rad/s). Measurements were taken for both ”spin-up” and ”spin-down” sequences in steps of $\approx$ 1rpm hence giving a total of 17×2 measurements. The temperature for all the experiments was kept constant at $\Delta T = 8 \pm 0.5K$. In contrast to the earlier annulus experiments, the top boundary was open to the atmosphere.

Vincze, Borchert, Achatz, von Larcher, Baumann, Hertel, Remmler, Beck, Alexandrov, Egbers, Froehlich, Heuveline, Hickel and Harlander (2014) also studied the system numerically in a collaboration between Goethe University Frankfurt, University of Heidelberg, FU Berlin, TU Dresden and TU Munich called MetStörm. In the collaboration they used five different numerical models to create “benchmarks” that could be used as test cases for atmospheric numerical model validations in the future. Each model computed solutions to the hydrodynamic equations of motion using the Boussinesq approximation with a stress-free upper boundary, but they differed in their initializing procedure, grids, time steps, boundary conditions and sub-grid parametrising schemes. The geometry of the simulated domain was the same as the experimental setup.

Both the experiments and the numerical models found well defined baroclinic waves with a dominant azimuthal wavenumber m, but time-averaged Fourier spectra showed the presence of harmonics of the basic wavenumber m. All baroclinic waves observed were of the steady wave type and no vacillation was observed. In experiments, the harmonic wave numbers were more visible than in simulations, with $m = 4$ accompanied by $m = 8$ or $m = 3$ with $m = 9$. The authors speculated that this mismatch might be due to surface effects, such as air currents at the open boundary, not being properly simulated in numerical models.

Considerable hysteresis was also observed in both experiments and numerics. The dominant wavenumber found at a particular rotation rate depended on whether that rotation rate was reached by spinning up or spinning down. This behaviour was consistent with previous experimental studies (Miller and Butler, 1991; Sitte and Egbers, 2000; von Larcher and Egbers, 2005). By analysing the hysteresis in the numerical models, it was concluded that different routes of releasing the stored potential energy played a role in the azimuthal wave number selection. During spin-up, kinetic energy is released due to baroclinic wave excitation and during spin-down, kinetic energy is released through the overturning background flow. However, the critical rotation rate at which baroclinic instability onsets (the first appearance of nonaxisymmetric modes), $\Omega_{crit} = 3\,\text{rpm}$, was unaffected by initial conditions in all experimental studies.

Another important 'benchmark' of Vincze, Borchert, Achatz, von Larcher, Baumann, Hertel, Remmler, Beck, Alexandrov, Egbers, Froehlich, Heuveline, Hickel and Harlander (2014) was the study of the drift rate of the dominant wavenumber around the annulus. Here Fourier transforms were used to measure azimuthal distance travelled by one of the structures. No wave dispersion was observed and all Fourier components propagate with the same drift rate. In the experiments and the numerics, it was observed that the drift rate was a decreasing function of Ω, roughly as expected from Eady (1949). They found the drift rate did depend on the dominant azimuthal wavenumber, so that with hysteretic cases where two different wavenumbers can be found at the same rotation rate, the $m = 3$ modes drifted faster than the $m = 4$ modes. Most numerical models over-predicted the drift rates in the system, which might be due to boundary layer drag at the sidewalls not being properly represented. However, increasing the grid resolution did lead to better prediction of the drift rate, showing that the system is dependent on grid resolution, especially to resolve the boundary layers. Comparing the structures observed, the models could not capture the small-scale variability but produced the large-scale patterns; thus showing that the coupling of the large and small scales does not play a critical role in this flow regime. According to Fein (1973), different behaviours were observed in experiments with rigid lids and free surface leading to different drift rates.

2.4 Rotating Fluids and Data Assimilation

As the thermal rotating annulus presents lots of complex behaviour related to planetary settings it has been proposed as a testing ground for data assimilation (DA). And although DA has been applied to many different areas its use in the area of thermal rotating annulus has been very limited.

Ravela et al. (2010) aimed to create a realtime observatory which produced evolving state estimates in realtime which were closer to the laboratory flow than either observation or numerical models predict separately. For this, the General Circulation Model

(GCM) was coupled with rotating annulus experiments using a localised version of the Ensemble Kalman filter (EnKF) (Evensen, 1994) which is related to Local Ensemble Kalman Filter (LEKF) created by Ott et al. (2004). It was thought that the experiment would guide the models on what to simulate and in turn the model would provide a guide on when and where to take measurements. The numerical model could be used to study parameters that could not be directly measured in the experiments (pressure, heat transport, etc) helping to predict behaviour in the experiment. This should lead to a better understanding of topics such as parameter estimation, model error and adaptive sampling.

The MIT GCM model solves the equations for a Boussinesq fluid in hydrostatic balance using the finite difference method in cylindrical coordinates (Marshall, Adcroft, Hill, Perelman and Heisey, 1997; Marshall, Hill, Perelman and Adcroft, 1997). It has crude parametrisations along with approximate numerical schemes and uncertain initial and boundary conditions. This leads to large errors between a simulation and the truth, hence the model needs input (initial conditions, etc) from the observations to make a better prediction. To test the system, the study used the tracking method as it can be easily replicated and the experiment repeated. The tracking method enables the numerical model to follow a specific structure in the experiment and reproduce the same movement and drift rates.

The scheme not only required fewer numerical simulations (smaller ensemble size) than purely sampling initial conditions would have done. But also prevented ensemble collapse by maintaining a justifiable uncertainty. Localization (4.7.1 for an explanation) of the scheme lead to an algorithm which could be parallelised and prevented spurious long-range correlations. Realtime performance was obtained through parallelisation, domain reduction, spectral-reduction and compute updates.

Young and Read (2013) used analysis correction method (Lorenc et al., 1991) to link their experiential data to the MORALS code and used this as a testbed for techniques not yet in operational use in meteorology. MORALS is a rotating annulus code that solves the Navier-Stokes equations in the context of a Boussinesq, viscous, incompressible thermally conducting liquid using a finite difference scheme. The laboratory data used for observation were obtained in a previous study Young et al. (2015) with a flat bottom annulus where horizontal velocity data were recorded every $5s$. The temperature gradient is kept at $\Delta T \sim 4K$ and spin-up and spin-down experiments were conducted.

Using this setup, irregularly distributed observations were successfully assimilated. This represents a more realistic situation of how observations are obtained in meteorology. In both regular and chaotic flow regimes, all residual errors between the analysis and observations were better than observational errors that would have resulted from only using the mean observed velocity. No statistical correlations were observed between residual errors and the rotation rate. Assimilation across wavenumber transitions were accurate in the regular flow regime but not as accurate in the chaotic flow regime, although the errors could be reduced by tuning the assimilation parameter.

At wavenumber $m = 2$ steady flow (2S), the assimilation converged to the optimum analysis in $25\,$sec, comparing well with observations. The assimilation free runs were the most accurate at $z = 7\,\text{cm}$ height but the residual error increased over time as the model and observation drift out of phase from each other. At the wave number 3, they found amplitude vacillation (3AV), and their aim was for the assimilation to track the vacillation and the wave itself and this was successful.

As the rotation rate is reduced from 0.825 to $0.800\,\text{rad/s}$ a transition from 3AV to 2S is observed. Here the assimilation model does well to track the transition change. A wavenumber $m = 3$ structural vacillation (3SV) is observed at around $\Omega = 2.3\,\text{rad/s}$. Due to the chaotic dynamics, assimilation is more of a challenge. As in the previous sections the assimilation can track the system well with the residual error remaining constant. Increasing the number of observed data points does not seem to decrease the residual error. The free running simulations do show chaotic dynamics, but there is an increase in residual error due to the simulation drifting out of phase.

To further test the assimilation the 3SV regime was kept but the rotation rate was abruptly changed from $\Omega = 2.4\,\text{rad/s}$ to $\Omega = 2.5\,\text{rad/s}$. After the change, there is a burst of high azimuthal velocity before settling down to the same state as the observations. The observed flow changes faster than the observation rate, and due to this the quality of the data that is assimilated is degraded increasing the residual error as well. This error could be lowered by reducing the time between assimilations.

At an even faster rotation rate of $\Omega = 3.1\,\text{rad/s}$ the assimilation keeps track of the different behaviour, such as vortex shedding, but the residual error oscillates substantially. This might just be due to observation and simulation drifting apart before being corrected.

2.5 Summary

This chapter presented an overview of the thermal rotating annulus. Brief descriptions were presented on experimental studies done with an annulus followed by a section on numerical studies that have been used to replicate experimental behaviour. Finally, the last section described two studies that have been done recently applying data assimilation to the thermal rotating annulus. Here both studies are trying to better predict experimental behaviour by using numerical methods and DA.

In the study presented in this thesis, the aim is to recreate the results from Young and Read (2013) using an updated experimental rig, MORALS code and EnKF method. The implementation technique used by Ravela et al. (2010) is a good place to start on understanding how this can be done.

Chapter 3

The MORALS code

The Meteorological Office/Oxford Rotating Annulus Laboratory Simulation (MORALS) code, is a Fortran code maintained at the Atmospheric, Oceanic and Planetary Physics (AOPP) group at the University of Oxford and is used to study a thermal rotating annulus. The code is written in a mixture of Fortran 77 and Fortran 90 and is designed to run on the UNIX/Linux operating system. Technical information about the model can be found on MORALS (2020). This chapter aims to give a brief summary of the equations and methods used in the code to simulate a rotating thermal annulus.

A general outline of how MORALS operates is given in section (3.1) along with equations and methods that are used. Section (3.2) details how MORALS creates the stretched grid used in its model along with how the Arakawa type C grid (Arakawa and Lamb, 1977) is used to store data. Section (3.3) describes the implementation of the MORALS code in 2D. Lastly the implementation of the code in 3D is given in section (3.4).

3.1 General Outline

MORALS integrates the Navier-Stokes equations along with the heat and mass continuity equations for a Boussinesq, viscous, incompressible, thermally conducting liquid contained in a rotating annulus using cylindrical polar coordinates. The equations are solved using a finite difference scheme for the axisymmetric case in 2D and for a full 3D model. Equations (3.2), (3.3) and (3.4) show the 3D version of the main equations solved in the model, for the 2D version $\partial/\partial\theta = 0$. The equations are solved for cylindrical coordinates (r, θ, z).

The schematic of the model used for the study is shown in figure (2.1). As this thesis presents a comparative study between model and experiment, the dimensions of the numerical model match the dimensions of the experimental setup used at AOPP. In accordance with the experiment, the inner radius of the cylinders is $a = 2.5\,\text{cm}$ and the outer radius is $b = 8.0\,\mathrm{cm}$ with the height being $D = 14.0\,\mathrm{cm}$. The geometry of the model is given in equation (3.1). In all the studies the temperature of the system is kept constant with the inner wall being 18° C and the outer wall being 22° C. The working fluid is a 17% by volume mixture of glycerol in water.

From figure (2.1), the coordinates are in the following ranges,

$$\begin{aligned} a \le r \le b&,\\ 0 \le \theta \le 2\pi&,\\ 0 \le z \le D&. \end{aligned} \tag{3.1}$$

A stretched grid is used in the radial (r) and vertical direction (z) to better resolve the boundaries at the inner and outer wall of the cylinder, and the top and bottom of the cylinder while the azimuth (θ) is uniformly spaced. An Arakawa-C grid (Arakawa and Lamb, 1977) being used for the discretized equations. More information about the grid is given in section (3.2). The three components of velocity (u, v, w) and temperature (T) are the main variable that are stored in the system. The four variables and a Poisson equation are used to solve for the kinetic pressure ($\Pi = p/\rho_0$) which is a diagnostic.

The equations solved are given on MORALS (2020). The kinematic viscosity ν, thermal diffusivity κ and the density perturbation $\Delta\rho$ are taken as simple functions of the temperature, with the constants used in their definition given in table 1 of Hignett et al. (1985). The velocity components (u, v, w) are given by the equations of motion

$$\begin{aligned} \frac{Du}{Dt} &= -\frac{\partial \Pi}{\partial r} + v\left(2\Omega + \frac{v}{r}\right) + \Omega^2 r \frac{\Delta\rho}{\rho_o} + \Delta^2(\nu, u) \\ &\quad + \frac{1}{r}\frac{\partial (ru)}{\partial r}\frac{\partial \nu}{\partial r} + \frac{\partial w}{\partial r}\frac{\partial \nu}{\partial z} - \frac{2}{r^2}\frac{\partial (\nu v)}{\partial \theta} + \frac{1}{r^2}\frac{\partial (vr)}{\partial r}\frac{\partial \nu}{\partial \theta}, \\ \frac{Dv}{Dt} &= -\frac{1}{r}\frac{\partial \Pi}{\partial \theta} - u\left(\frac{v}{r} + 2\Omega\right) + \Delta^2(\nu, v) \\ &\quad + \frac{1}{r}\frac{\partial u}{\partial \theta}\frac{\partial \nu}{\partial r} + \frac{1}{r^2}\frac{\partial v}{\partial \theta}\frac{\partial \nu}{\partial \theta} + \frac{1}{r}\frac{\partial w}{\partial \theta}\frac{\partial \nu}{\partial z} + \frac{2}{r^2}\frac{\partial (u\nu)}{\partial \theta}, \\ \frac{Dw}{Dt} &= -\frac{\partial \Pi}{\partial z} - \frac{g\Delta\rho}{\rho_o} + \widetilde{\Delta}^2(\nu, w) + \frac{\partial u}{\partial z}\frac{\partial \nu}{\partial r} + \frac{\partial w}{\partial z}\frac{\partial \nu}{\partial z} + \frac{1}{r}\frac{\partial v}{\partial z}\frac{\partial \nu}{\partial \theta}, \end{aligned} \tag{3.2}$$

$$\text{where} \qquad \frac{D}{Dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial r} + \frac{v}{r}\frac{\partial}{\partial \theta} + w\frac{\partial}{\partial z},$$

the continuity equation

$$\frac{1}{r}\frac{\partial (ru)}{\partial r} + \frac{1}{r}\frac{\partial v}{\partial \theta} + \frac{\partial w}{\partial z} = 0, \tag{3.3}$$

and the heat equation

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial r} + \frac{v}{r}\frac{\partial T}{\partial \theta} + w\frac{\partial T}{\partial z} = \tilde{\Delta}^2(\kappa, T). \tag{3.4}$$

The diffusion operators in these equations are defined by

$$\Delta^2(\lambda, A) = \frac{1}{r}\frac{\partial}{\partial r}\left\{\lambda r\frac{\partial}{\partial r}\left(\frac{A}{r}\right)\right\} + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(\lambda\frac{\partial A}{\partial \theta}\right) + \frac{\partial}{\partial z}\left(\lambda\frac{\partial A}{\partial z}\right), \tag{3.5}$$

and

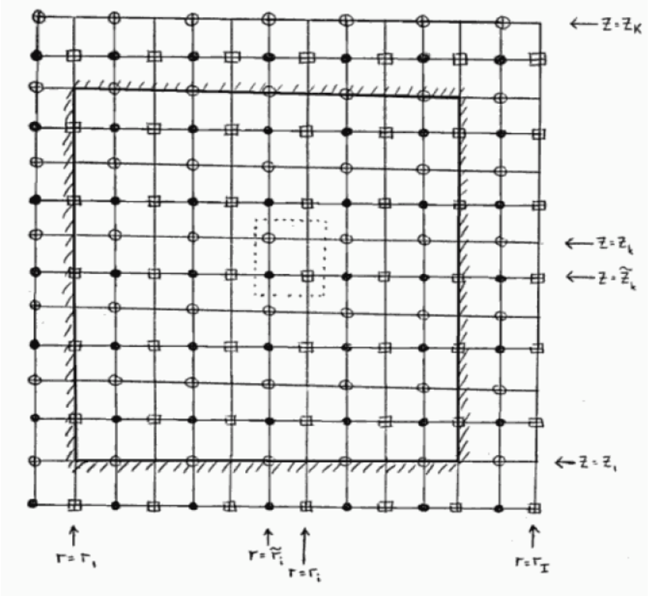
$$\tilde{\Delta}^2(\lambda, A) = \frac{1}{r^2}\frac{\partial}{\partial r}\left\{\lambda r^3\frac{\partial}{\partial r}\left(\frac{A}{r}\right)\right\} + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(\lambda\frac{\partial A}{\partial \theta}\right) + \frac{\partial}{\partial z}\left(\lambda\frac{\partial A}{\partial z}\right). \tag{3.6}$$

with boundary conditions

$$\begin{array}{lllll} \text{At } R = a: & & & & \\ T = T_a & \dfrac{d\Pi}{dr} = 0 & u = 0 & v = 0 & w = 0 \\ \text{At } R = b & & & & \\ T = T_b & \dfrac{d\Pi}{dr} = 0 & u = 0 & v = 0 & w = 0 \\ \text{At } z = 0, d & & & & \\ \dfrac{dT}{dz} = 0 & \dfrac{d\Pi}{dz} = 0 & u = 0 & v = 0 & w = 0 \end{array} \tag{3.7}$$

3.2 The Grid

Resolving the boundary layers is always an issue when running simulations. One of the most common boundary layers observed in atmospheric and oceanographic studies is the Ekman layer, where there is a balance between frictional forces, Coriolis force and the pressure gradient. When it comes to a rotating annulus it is easier to resolve the structure and behaviour of a fluid in the middle of the cylinder than it is at the boundaries where an Ekman layer forms (Gill, 1982). Normally increasing the resolution of the model can help overcome these issues but the higher resolution slows down the integration of the model, and as mentioned above the higher resolution is not necessary to resolve behaviour in the middle of the model. A good way to tackle this problem is to have a stretched grid, where there are more grid points at the boundary than the middle. This way the Ekman layer can be resolved without the need to having more points in the



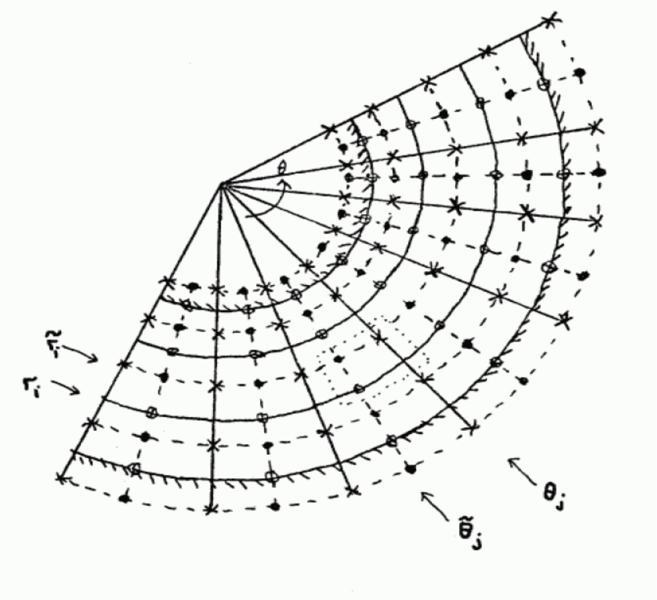
The finite–difference grid (mapped in the regular (x, y) space). The location of the boundaries is marked by shading; the (i, k) cell is delineated by the dotted line. Flow variables are evaluated at the following points:

⊕ vertical velocity w

⊞ radial velocity u

● temperature T, azimuthal velocity v, pressure Π

Figure 3.1: 2D Axisymmetric Grid (Farnell and Plumb, 1975)



The finite–difference grid seen from above. The location of the boundaries is marked by shading; the (i, j) cell is delineated by the dotted line. Flow variables are evaluated at the following points:

● vertical velocity w, pressure Π, and temperature T

⊕ radial velocity u

⊞ azimuthal velocity v

Figure 3.2: 3D Grid (Farnell and Plumb, 1976)

middle of the model to slow down the integration.

The stretched grid in MORALS is created using a hyperbolic tangent for both the radial and vertical direction as shown in equation (3.8). The azimuthal grid in the 3D model is regularly spread. A hyperbolic tangent stretching is used,

$$\begin{aligned} r - \frac{1}{2}(b-a) &= \xi(x) = r_0 \tanh(\alpha_r x), \\ z - \frac{1}{2}d &= \eta(y) = z_0 \tanh(\alpha_z y), \end{aligned} \tag{3.8}$$

where a and b are the inner and outer radius of the cylinder and d is the height of the cylinder. The constants r_0, z_0, α_r and α_z are to be calculated. First a regular grid is defined in x,y coordinates using the equations (3.9) below.

$$\begin{aligned} x_i &= (b-a)\left(\frac{2i-I}{2I-4}\right), \\ y_k &= d\left(\frac{2k-K}{2K-4}\right), \end{aligned} \tag{3.9}$$

where I and K are the user-defined number of grid points in r and z respectively. The range in the new coordinates is then fixed, i.e $-x_1 = x_{I-1} = \frac{1}{2}(b-a)$ in the radial direction and $-y_1 = y_{K-1} = \frac{1}{2}d$ in the vertical direction. Using equations (3.8) and the range of the new coordinates to ensure that $x_1 = a$ and $y_1 = 0$ gives

$$\begin{aligned} \frac{b-a}{2r_0} &= \tanh\left[\frac{\alpha_r}{2}(b-a)\right], \\ \frac{d}{2z_0} &= \tanh\left[\frac{\alpha_z}{2}d\right]. \end{aligned} \tag{3.10}$$

To get a stretched grid the degree of stretching needs to be determined. This is done using

$$\frac{d\xi}{dx} = \frac{\Delta r}{\Delta x} = G_r, \tag{3.11}$$

$$\frac{d\eta}{dy} = \frac{\Delta z}{\Delta y} = G_y. \tag{3.12}$$

To get the values of Δr and Δz, first the Ekman number Ek and the Ekman layer thickness Ek_z are calculated,

$$\begin{aligned} Ek &= \frac{\nu_0}{\Omega d^2}, \\ Ek_z &= \sqrt{\frac{\nu_0}{\Omega}}, \end{aligned} \tag{3.13}$$

where the viscosity (ν_0) along with the rotation rate (Ω) and the height of the cylinder (d) is defined by the user. Δx and Δy are worked out using the grid increment in the x and y space respectively

$$\begin{aligned} \Delta r =& \frac{(b-a) * Ek^{\frac{1}{3}}}{BI}, \\ \Delta x =& \frac{b-a}{I-2}, \\ \Delta z =& \frac{Ek_z}{BK}, \\ \Delta y =& \frac{d}{K-2}, \end{aligned} \tag{3.14}$$

here BI and BK are the boundary layer thickness factors defined as $BI = 3.0$, $BK = 3.0$, which were found by numerical experiment by Farnell and Plumb (1976). In using equations (3.14) we are assuming the sidewalls have a Stewartson layer whose thickness scales as $Ek^{1/3}$ and the horizontal endwalls have an Ekman layer thickness that scales as $Ek^{1/2}$, (James et al., 1981). Given that the stretched gradient is small $0 < \boldsymbol{G}_r, \boldsymbol{G}_z < 1$, the function

$$\varsigma(\gamma) = \tanh\left[G\frac{\gamma}{1-\gamma^2}\right] - \gamma, \tag{3.15}$$

is solved using numerical iteration. The first value of γ which gives a positive solution for the function is termed γ_0. Differentiating the equation (3.8), and using equation (3.10) gives,

$$\begin{aligned} \boldsymbol{G}_r =& \alpha_r r_0 \left[1 - \left(\frac{b-a}{2r_0}\right)^2\right], \\ \boldsymbol{G}_z =& \alpha_z z_0 \left[1 - \left(\frac{d}{2z_0}\right)^2\right]. \end{aligned} \tag{3.16}$$

Now α_r and α_z can be eliminated to give

$$\begin{aligned} r_0 &= \frac{b-a}{2\gamma_o}, \\ z_0 &= \frac{d}{2\gamma_o}. \end{aligned} \tag{3.17}$$

So once equation (3.15) is solved r_0 can be calculated using (3.17) and then α_r using (3.16). Now the the constants have been calculated, the values for r and z can be worked out using (3.8). The intermediate points can now also be calculated

$$\begin{aligned} \tilde{x}_i &= \frac{1}{2}\left(x_i + x_{i-1}\right), \\ \tilde{y}_k &= \frac{1}{2}\left(y_k + y_{k-1}\right), \end{aligned} \tag{3.18}$$

and then the same procedure is used to find the corresponding values of r and z. The azimuthal grid is regularly spaced in the 3D model hence only one grid increment is needed ($\delta\theta$). To ensure that the grid is periodic the points are arranged such that $\theta_J + \delta\theta = \theta_1$. Both θ_J and $\widetilde{\theta}_J$ can be easily calculated giving the distribution of the grid.

Due to the staggered grid a few changes need to be made to the finite difference scheme. On a regular grid

$$\left(\frac{\partial\psi}{\partial x}\right)_{x=\tilde{x}_i} = \frac{\psi(x_i) - \psi(x_{i-1})}{\Delta x} + O\left(\Delta x^2\right), \tag{3.19}$$

so that

$$\left(\frac{\partial\psi}{\partial r}\right)_{r=\tilde{r}_i} = \frac{\psi(r_i) - \psi(r_{i-1})}{\Delta x \xi'\left(\tilde{x}_i\right)} + O\left(\Delta x^2\right), \tag{3.20}$$

with smooth $\xi\left(\tilde{x}_i\right)$ on the staggered grid,

$$\begin{aligned} \left(\frac{\partial\psi}{\partial r}\right)_{r=\tilde{r}_i} &\simeq \frac{\psi(r_i) - \psi(r_{i-1})}{\delta r_i} \equiv \left(\delta_r\psi\right)_{r=\tilde{r}_i}, \\ \left(\frac{\partial\psi}{\partial r}\right)_{r=r_i} &\simeq \frac{\psi(\tilde{r}_{i+1}) - \psi(\tilde{r}_i)}{\delta\tilde{r}_{i+1}} \equiv \left(\delta_r\psi\right)_{r=r_i}, \\ \left(\frac{\partial\psi}{\partial z}\right)_{z=\tilde{z}_k} &\simeq \frac{\psi(z_k) - \psi(z_{k-1})}{\delta z_k} \equiv \left(\delta_z\psi\right)_{z=\tilde{z}_k}, \\ \left(\frac{\partial\psi}{\partial z}\right)_{z=z_k} &\simeq \frac{\psi(\tilde{z}_{k+1}) - \psi(\tilde{z}_k)}{\delta\tilde{z}_{k+1}} \equiv \left(\delta_z\psi\right)_{z=z_k}. \end{aligned} \tag{3.21}$$

Variable	r	z	θ
u	$r = r_i$	$z = \tilde{z}_i$	$\theta = \tilde{\theta}_i$
v	$r = \tilde{r}_i$	$z = \tilde{z}_i$	$\theta = \theta_i$
w	$r = \tilde{r}_i$	$z = z_i$	$\theta = \tilde{\theta}_i$
T	$r = \tilde{r}_i$	$z = \tilde{z}_i$	$\theta = \tilde{\theta}_i$

Table 3.1: Grid position of stored variables

The second derivative

$$\left(\frac{\partial^2\psi}{\partial r^2}\right)_{r=\tilde{r}_i} \simeq \left[\delta_r\left(\delta_r\psi\right)\right]_{r=\tilde{r}_i} = \frac{\left(\delta_r\psi\right)_{r=r_i} - \left(\delta_r\psi\right)_{r=r_i-1}}{\delta r_i}, \tag{3.22}$$

both accurate to $O\left(\Delta x^2\right)$

An Arakawa-C staggered grid is then created to store the variables at different positions in the grid. This increases accuracy and reduces the need to do averaging. Table (3.1) gives a summary of where the variables are stored on r, θ and z coordinates. Figures (3.1) and (3.2) show a sketch of what the 2D and 3D grids look like along with where each of the variables are stored.

3.3 The 2D code

To solve the code for a 2D system the equations need to be evaluated for axisymmetric motion i.e. $\partial/\partial\theta = 0$. The equations (3.2) are changed into their finite difference forms as shown below

$$\delta\bar{u}^t = -A_1(u) - \delta_r\Pi + [2\Omega v]^* + \left[\frac{v^2}{r}\right]^* + \boldsymbol{D}_u^{\dagger}, \tag{3.23}$$

$$\delta\bar{v}^t = -\frac{1}{2}A_1(v) - \frac{2\Omega}{r}(\overline{ru}^r) + \boldsymbol{D}_v^{\dagger}, \tag{3.24}$$

$$\delta\bar{w}^t = -A_1(w) - \delta_z\Pi + [\sigma]^* + \boldsymbol{D}_w^{\dagger}, \tag{3.25}$$

$$\delta\bar{T}^t = -A_2(T) + \boldsymbol{D}_T^{\dagger}, \tag{3.26}$$

where the asterisk (*) in the equations denotes non-standard averaging, see equations (3.31) and (3.32) below, and the dagger denotes that a DuFort-Frankel scheme is used to solve these parts. Looking at the equation in detail, A represents the advection terms

$$\begin{aligned} A_2(f) =& \frac{1}{r}\delta_r\left[\widetilde{ru}\overline{f}^{r}\right] + \delta_z\left[\widetilde{w}\overline{f}^{z}\right], && (3.27)\\ A_1(f) =& \frac{1}{2}\left\{A_2(f) + \frac{1}{r}\overline{[(\widetilde{ru})\delta_r f]}^{r} + \overline{[\widetilde{w}\delta_z f]}^{z}\right\}, && (3.28) \end{aligned}$$

in which the tilde ($\tilde{a}$) indicates that the variables may have to be interpolated onto their respective grid points. This is due to the Arakawa-C grid being used, where the variables are stored at different point in the grid. The D terms represent diffusion and have the form:

$$\begin{aligned} D_u =& \frac{1}{r^2}\delta_r\left\{\nu r^3\delta_r\left(\frac{u}{r}\right)\right\} + \delta_z\left\{\overline{\overline{\nu}}^{rz}\delta_z u\right\} + \frac{1}{r}\delta_r\left(\overline{ru}^{r}\right)\delta_r\nu + \delta_r\overline{w}^{z}\delta_z\overline{\nu}^{r},\\ D_v =& \frac{1}{r^2}\delta_r\left\{\overline{\nu}^{r}r^3\delta_r\left(\frac{v}{r}\right)\right\} + \delta_z\left\{\overline{\nu}^{z}\delta_z v\right\},\\ D_w =& \frac{1}{r^2}\delta_r\left\{\overline{\overline{\nu}}^{rz}r^3\delta_r\left(\frac{u}{r}\right)\right\} + \delta_z\left\{\nu\delta_z w\right\} + \delta_z\overline{u}^{r}\delta_r\overline{\nu}^{z} + \delta_z\overline{w}^{z}\delta_z\nu,\\ D_T =& \frac{1}{r}\delta_r\left\{\kappa r\delta_r T\right\} + \delta_z\left\{\overline{\kappa}^{z}\delta_z T\right\}, \end{aligned} \qquad (3.29)$$

here ν (kinematic viscosity) and κ (thermal diffusivity) are functions of temperature

$$\begin{aligned} \nu =& \nu_0\left\{1 + \nu_1\left(T - T_0\right) + \nu_2\left(T - T_0^2\right)\right\},\\ \kappa =& \kappa_0\left\{1 + \kappa_1\left(T - T_0\right) + \kappa_2\left(T - T_0^2\right)\right\}. \end{aligned} \qquad (3.30)$$

Being functions of temperature these terms are evaluated at $\tilde{r}_i$, $\tilde{z}_k$ and are then interpolated to be used in other equations as needed. The σ in the equation is the buoyancy term given by

$$[\sigma]^{*m}_{i,k} = -g\left[\rho_1\left(\overline{T}^{zm}_{i,k} - T_0\right) + \rho_2(T^m_{i,k} - T_0)\left(T^m_{i,k+1} - T_0\right)\right]. \qquad (3.31)$$

The Coriolis and centrifugal terms

$$\begin{aligned} [2\Omega v]^* =& \frac{\Omega}{\widetilde{\delta r}_{i+1}}\left(v^m_{i,k}\delta r_i + v^m_{i+1,k}\delta r_{i+1}\right), && (3.32)\\ \left[\frac{v^2}{r}\right]^* =& v^m_{i,k}v^m_{i+1,k}\frac{\tilde{r}^2_{i+1} - \tilde{r}^2_i}{\tilde{r}_i\tilde{r}_{i+1}\tilde{\delta}r_{i+1}}. && (3.33) \end{aligned}$$

The pressure (Π) is solved for using the Poisson equation by the alternating direction iteration (ADI) method (Farnell and Plumb, 1975). Here the equation is solved for the radial direction (r) before solving for the vertical direction (z), i.e.

$$(O_r + O_z)\,\Pi = F. \tag{3.34}$$

Finally the continuity equation is given as

$$\Delta \equiv \frac{1}{r}\delta_r(ru) + \delta_z w = 0. \tag{3.35}$$

In the first step of integration the pressure field (Π) is calculated using the fields of other variables (u, v, w, T) at time m and $m-1$ along with the pressure field at times $m-1$ and $m-2$. The finite difference scheme is then used to calculate the value of the variables for the next time step ($m+1$) for all the interior points. A subroutine applies the boundary conditions for all the different fields and the stored fields for the previous time step ($m-1$) are discarded. The fields are smoothed around every 21 time steps to dampen any instabilities that can arise due to the ‘leap-frog’ time differencing scheme. The smoothing happens by averaging the fields over adjacent time steps and then restarting the calculations using these values.

A lot of the parameters can be changed in the 2D parameter file (**sm2d**). For this study most of these parameters were kept constant to the default value. This is because the focus of the study is to test the data assimilation side of the study, so it is assumed that the values given will give a reasonable results for simulation. The few parameters that were changed were the number of grid points, rotation rate and the time step length.

3.4 The 3D code

All of the discretization for the code used in the 2D model is carried forward for the 3D code but in addition the variation in the azimuthal direction is factored in. The grids in the radial and vertical direction are stretched in the 3D model as well to resolve the boundary layers, with the azimuthal grid now also in use, which is regularly spaced.

The finite-difference version of the equations used in the 3D code are,

$$\delta\bar{u}^t = -A_1(u) - \delta_r\Pi + [2\Omega v]^* + \left[\frac{v^2}{r}\right]^* + \boldsymbol{D}_u^\dagger, \tag{3.36}$$

$$\delta\bar{v}^t = -\frac{1}{r}A_1(rv) - \frac{1}{\tilde{r}}\delta_\theta\Pi - \frac{2\Omega}{\tilde{r}}(\overline{ru}^r) + \boldsymbol{D}_v^\dagger, \tag{3.37}$$

$$\delta\bar{w}^t = -A_1(w) - \delta_z\Pi + [\sigma]^* + \boldsymbol{D}_w^\dagger, \tag{3.38}$$

$$\delta\bar{T}^t = -A_2(T) + \boldsymbol{D}_T^\dagger. \tag{3.39}$$

The asterisk (*) in the equations again denotes non-standard averaging and the dagger denotes that a DuFort-Frankel scheme is used to solve these parts.

The buoyancy term is unchanged from before, while the Coriolis and centrifugal terms are modified slightly to include the azimuthal term to give

$$[\sigma]^{*m}_{i,j,k} = - g\left[\rho_1\left(\overline{T}^{zm}_{i,j,k} - T_0\right) + \rho_2(T^m_{i,j,k} - T_0)\left(T^m_{i,j,k+1} - T_0\right)\right], \tag{3.40}$$

$$[2\Omega v]^* = \frac{\Omega}{\widetilde{\delta r}_{i+1}}\left(\overline{v}^\theta_{i,j,k}\delta r_i + \overline{v}^\theta_{i+1,j,k}\delta r_{i+1}\right), \tag{3.41}$$

$$\left[\frac{v^2}{r}\right]^* = \frac{v_{i,j,k}v_{i+1,j,k} + v_{i,j-1,k}v_{i+1,j-1,k}}{4\tilde{\delta r}_{i+1}}\left(\frac{\tilde{r}_{i+1}}{\tilde{r}_i} - \frac{\tilde{r}_i}{\tilde{r}_{i+1}}\right). \tag{3.42}$$

The advection terms are given by

$$A_2(f) = \frac{1}{r}\delta_r\left[\left(\widetilde{ru}\right)\overline{f}^r\right] + \delta_z\left[\tilde{w}\overline{f}^z\right] + \frac{1}{r}\delta_\theta\left[\tilde{v}\overline{f}^\theta\right], \tag{3.43}$$

$$A_1(f) = \frac{1}{2}\left\{A_2(f) + \frac{1}{r}\overline{\left[\left(\widetilde{ru}\right)\delta_r f\right]}^r + \overline{\left[\widetilde{w}\delta_z f\right]}^z + \frac{1}{r}\overline{\left[\widetilde{v}\delta_\theta f\right]}^\theta\right\}. \tag{3.44}$$

The diffusion terms are

$$\boldsymbol{D}_u = \frac{1}{r^2}\delta_r\left\{\nu r^3\delta_r\left(\frac{u}{r}\right)\right\} + \delta_z\left\{\overline{\overline{\nu}}^{rz}\delta_z u\right\} + \frac{1}{r^2}\delta_\theta\left\{\overline{\overline{v}}^{r\theta}\left(\delta_\theta u - \overline{v}^r\right)\right\} \tag{3.45}$$

$$+ \frac{1}{r}\delta_r\nu\delta_r(\overline{ru}^r) + \frac{1}{r^2}\delta_r(\overline{rv}^\theta)\delta_\theta\overline{\nu}^r + \delta_r\overline{w}^z\delta_z\overline{\nu}^r, \tag{3.46}$$

$$\boldsymbol{D}_v = \frac{1}{r^2}\delta_r\left\{\overline{\overline{\nu}}^{r\theta}r^3\delta_r\left(\frac{v}{r}\right)\right\} + \delta_z\left\{\overline{\overline{\nu}}^{\theta z}\delta_z v\right\} + \frac{1}{r^2}\delta_\theta\{\nu(\overline{u}^r + \delta_\theta v)\} \tag{3.47}$$

$$+ \frac{1}{r}\delta_r\overline{\nu}^\theta\delta_\theta\overline{u}^r + \frac{1}{r^2}\delta_\theta\overline{v}^\theta\delta_\theta\nu + \frac{1}{r}\delta_z\overline{\nu}^\theta\delta_\theta\overline{w}^z, \tag{3.48}$$

$$\boldsymbol{D}_w = \frac{1}{r}\delta_r\left\{\overline{\overline{\nu}}^{rz}r\delta_r w\right\} + \delta_z\left\{\nu\delta_z w\right\} + \frac{1}{r^2}\delta_\theta\left\{\overline{\overline{\nu}}^{\theta z}\delta_\theta w\right\} \tag{3.49}$$

$$+ \delta_z\overline{u}^r\delta_r\overline{\nu}^z + \frac{1}{r}\delta_z\overline{v}^\theta\delta_\theta\overline{\nu}^z + \delta_z\overline{w}^z\delta_z\nu, \tag{3.50}$$

$$\boldsymbol{D}_T = \frac{1}{r}\delta_r\left\{\overline{\kappa}^r r\delta_r T\right\} + \delta_z\left\{\overline{\kappa}^z\delta_z T\right\} + \frac{1}{r^2}\delta_\theta\left\{\overline{\kappa}^\theta\delta_\theta T\right\}. \tag{3.51}$$

The continuity equation becomes

$$\Delta \equiv \frac{1}{r}\delta_r(ru) + \delta_z w + \frac{1}{r}\delta_\theta v = 0. \tag{3.52}$$

The equations for the kinematic viscosity and thermal diffusivity are the same as for the 2D model given in equation (3.30). The Poisson equation is solved using Fourier transforms in the azimuthal direction. This automatically applies periodicity as the boundary condition. The alternating direction iteration (ADI) method is still used to solve for the pressure in the radial and vertical directions at each of the azimuthal points.

The way the equations are solved for the fields in the 3D model is similar to how they were solved in the 2D one. The main issue the 3D model has is how to store the large amount of information that the system produces. Four backup data sets are used to juggle the data during various parts of the simulations, these are numbered 1, 2, 3 and 5. The values of the fields are initially stored in dataset 1. Velocity source components are first calculated from dataset 1 and stored in dataset 3. The guess for the pressure field is also calculated and stored in dataset 5. Using the velocity source information from dataset 3 and the pressure field from dataset 5, velocity and temperature are calculated and stored in dataset 2 along with pressure. The boundary conditions are also applied and all the data is stored in the same format as they were in dataset 1. The integration is now complete.

3.5 Running the code

Generally first the 2D model is run using **sm2d**. This result will be used as a base to create the ensemble. The 2D model is converted into a 3D model using **stp2d3d**. During the conversion perturbation is added with different seeding leading to each ensemble having unique initial conditions. Finally the 3D model is run using **sm3d** file. **stp2d3d** and **sm3d** are run sequentially and repeated to create the ensemble. The results created for each ensemble are stored in specific folders related to their ensemble to avoid data being overwritten. Data created was stored in ensemble specific folders to make sure data was not overwritten.

3.6 Summary

In this chapter a general outline was given on how the MORALS code works. In section (3.1) the main equations solved by the code are shown along with the parameter values used in the calculation. A general outline of how the code works is also given. Section (3.2) outlines the creation of the stretched grid along with how the staggered grids are created for the 2D and 3D model.

Details are given on how the 2D code works in section (3.3), finally section (3.4) details how the 3D code works works. The finite difference versions of the equations solved in 2D and 3D code are shown in their respective sections, along with how the different elements in the equations are solved. More technical information about the algorithm used in the model can be found in Farnell and Plumb (1975, 1976).

Chapter 4

Data Assimilation and Ensemble Kalman Filter

Data assimilation (DA) is a method of combining model data and observations. Initially developed for numerical weather prediction (Panofsky, 1949; Gilchrist and Cressman, 1954), data assimilation is now used in a wide variety of fields such as oceanography, traffic modelling, crime modelling, etc (Evensen, 2003). Generally used to obtain a better prediction, data assimilation is also used to tackle other interesting questions such as parameter estimation. Here the DA method is used to predict the values of the parameters for the model. This technique becomes important when there is uncertainty with the values of the parameters that should be used. DA can also be used to optimise observation to look at how many or what type of observations are needed for accurate prediction and look at best possible places for observations which should lead to better prediction.

As time has gone on many different approaches for data assimilation have been created. Modern data assimilation can be divided into three distinct methods: 1. variational methods, 2. ensemble methods, 3. hybrid methods (Bannister, 2017). The optimal method to use in a given situation depends on the model and type of observations available. Houtekamer and Zhang (2016) gives a detailed review of the different variations of EnKF that are currently in use for NWP. The aim of this chapter is to give a simple overview which will be helpful for the results presented in this thesis.

This chapter aims to give a brief overview of DA and its current limitations and methods used to overcome them. Section (4.1) gives a general outline of DA. Section (4.2) talks about the different flavours of DA currently in use, while section (4.3) looks at the pros and cons of variational and ensemble data assimilation. In section (4.4) an introduction to Kalman Filtering (KF), including the Ensemble Kalman Filter (EnKF) is given. Section (4.5) gives a formulation of EnKF using a Bayesian approach. Section (4.6) gives a brief description of how EnKF is implemented into the python code which was used for the study presented in this thesis. This includes giving examples of 1D and multidimensional problems. Section (4.7) looks at the limitations of EnKF and what tuning parameters are currently used to overcome them such as multiplicative and additive inflation and localisation.

4.1 General outline

Numerical weather prediction (NWP) simulations are initial-value problems where the determination of the initial condition is very important (Kalnay, 2003). In some cases, observations can be used to obtain the initial condition but these can be non-uniformly spread in both space and time. In the case of observations not being available or useful, a best guess can be used to help generate the initial conditions on all grid points. Using these initial conditions the model can be run and is called the background or forecast. After a certain amount of time this background is compared to the observations or 'truth'. Using these two datasets a few calculations such as variable averages, variance, covariance, etc are done and new values for the model variables are worked out. This correction/new initial condition is called the analysis and is used to continue the model run, hopefully, leading to a more accurate prediction. Figure (4.1) outlines the process discussed. The cycle of analysis and forecast can be continued at user defined intervals or through an adaptive setup.

More mathematically, data assimilation can be explained by defining a forecast, x^f, an observation, y^o and a true state for the model and observations, x^t, y^t. Both the forecast (x^f) and observations (y^o) are a combination of the true state plus some error in the system as seen in equation (4.1) below,

$$\begin{aligned} x^f &= x^t + \mu, \\ y^o &= y^t + \epsilon, \end{aligned} \tag{4.1}$$

where μ and ϵ are the error in the model and observations respectively. Now the model and observations can be combined as seen in equation (4.2) below to obtain an analysis,

$$x^a = x^b + W\left[y^o - H(x^f)\right], \tag{4.2}$$

here H is a operator that maps the model to the observation in case they have different dimensions. W is a weight that is used to help correct the model. Most assimilation schemes are based on equation (4.2) with the only difference between them being how they combine the observation and background to produce the weight and the analysis.

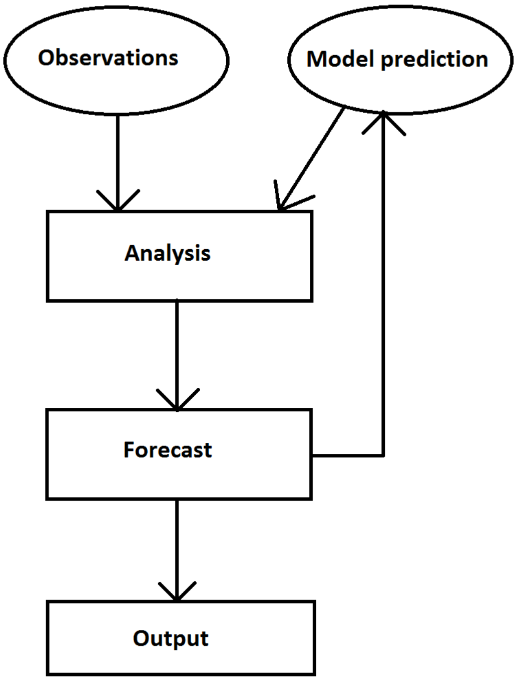


Figure 4.1: General schematic for Data assimilation

4.2 Flavours of Data Assimilation

The story of data assimilation is fairly new with the earliest approaches to data assimilation in numerical weather prediction (NWP) just being hand interpolated observation on a model grid (Panofsky, 1949; Gilchrist and Cressman, 1954). As this approach was such a time consuming method it became clear that a more automated method needs to be developed. Over the years with advances in computational technology new more reliable methods have been introduced, with the most successful being variational methods. With the addition of Ensembles methods and hybrid methods that have been developed in recent years, it has become important to look at the advantages and disadvantages of each approach. In the following sections, a summary is given for a few important methods that have been developed for DA.

4.2.1 *Sequential method*

One of the earliest methods to be developed for DA were Sequential methods (Bergthórsson and Döös, 1955; Barnes, 1964; Lorenc et al., 1991). Here the assimilation corrects the first guess using the observation and an error. The error depends on the first guess and observation. An important thing to note is that the sample size used for calculating the correction is not fixed. The method collects the data and evaluates them sequentially until a convergence (minimising the distance in a given metric) is obtained. As a convergence can be reached without needing to look at all the data, a conclusion can be reached much earlier compared to classical estimation thus saving computational time and leading to faster prediction. But since the statistics calculated will be different depending on how much of the data is used compared to when all of the data is used, the results concluded are biased and hence can lead to incorrect corrections to the model.

4.2.2 *Gradient descent filter*

Gradient descent filters use full nonlinear models to look at 'trajectories' of the date which are close to the observations (Stemler and Judd, 2009; Judd et al., 2008). The method uses something called a cost function, which is a metric that designates a value associated with an event. In many cases, this is a function to find the difference between the estimated and true values of an event. The idea is to minimize this cost function using the steepest descent.

The gradient descent algorithm does not use the exact values of the forecast. The forecast values are always augmented using weights to add stochasticity to the data. Hence the final calculated weight will never reach their optimal value in the absolute sense. However, convergence is possible when looking at the mean, as even though the weights may change by small amounts, the changes are close to the optimal weight. But if the variance with which the weights change is large, convergence in mean would be misleading. This problem may occur if the value of step-size of the gradient is not chosen properly.

4.2.3 *Variational method*

Variational data assimilation methods also minimise a cost function. This is done by taking into account the size of the observations, the background and, the errors in the background and observation. This cost function gives the optimal analysis state (Sasaki, 1970; Lorenc et al., 2000; Rawlins et al., 2007). Variational methods normally use linearised observation vector and a linearised version of the forecast model. The background and observational error covariance matrix are specified.

3D-Var method has been used extensively in many NWP problems as it gave the user the ability to do three dimensional data assimilation (Sasaki, 1970; Lorenc et al., 2000; Rawlins et al., 2007). 4D-Var (three space dimensions and time) method was developed as it allows for asynoptic data to be used for assimilation. In meteorological terms, this means that apart from the normal surface weather observations the method can also use supplementary observations such as satellite radiance. 4D-Var is computationally more expensive than it's previous iteration 3D-Var, but has proven to be significantly more accurate. Variational methods have been very popular in NWP with 4D-Var being used by several weather prediction centres such as ECMWF, Frace, UK, Japan, Canada (Kalnay et al., 2007).

4.2.4 *Ensemble method*

As the name suggests Ensemble methods use an ensemble of forecasts to perform data assimilation. Unlike like variational methods, the errors in the ensemble system evolve with time and are calculated using the ensemble data during the analysis. The system also does not require a linearised version of the forecast model. This means non-linear systems can be used for data assimilation. In almost all versions of data assimilation studies, the model used to predict the truth is a high resolution model which is computationally expensive to solve. With ensemble methods lots of lower resolution models are used which are computationally less expensive, the idea being that even if these low resolution models lead to inaccurate predictions they should have a Gaussian distribution around the truth, so when averaged they will lead to the correct answer.

The Ensemble Kalman filter (EnKF) is one of the best examples of Ensemble methods (Evensen, 1994; van Leeuwen, 2010). Since the introduction in 1994, EnKF has been used in a wide variety of areas (Evensen, 2003). The ensemble size is a major factor on how effective EnKF can be with large ensemble sizes leading to a better distribution around the truth and hence a better prediction. But a large ensemble lead to more computation even with the lower resolution models. To tackle the need for large ensembles variations of the ensemble methods have been developed, this includes the deterministic and stochastic ensemble methods along with the Ensembles square root filter (EnSRF) and the Local Ensemble Transform Filter (LETKF) (Houtekamer and Zhang, 2016).

4.2.5 *Hybrid methods*

As the name suggests these are methods which are a combination of different methods. These methods are very new and have been developed to try and keep the advantages of each of the methods in mind while trying to replace their disadvantages with other methods. A good review of all the different developments of hybrid models currently in use is given in Bannister (2017). An example of hybrid methods are the 4D-Var/EnKF assimilation schemes (‘EnVar’). Variations of EnVar exists but one of the simplest is a method which uses Var analysis scheme on an ensemble system as used by Zhang et al. (2009) using Lorenz (1996) and Bowler et al. (2008) with the the Met Office Global and Regional Ensemble Prediction System (MOGREPS).

4.3 Variational Vs Ensemble methods

Two of the most important methods currently used are Variational and Ensemble data assimilation methods. As discussed in section (4.2.3), Variational methods are based on minimising a cost function. Variational methods have been used in NWP for some time where 3D-Var became the workhorse (Parrish and Derber, 1992). In recent years 4D-Var taken over the reins with many centres (ECMWF, France, UK, Japan, Canada) using it as 4D-Var has proven to be more accurate compared to 3D-Var (Kalnay et al., 2007).

Ensemble methods are comparatively very new with the Ensemble Kalman filter (EnKF) being proposed by Evensen (1994). Unlike variational methods, EnKF can use non-linear models and have an evolving error covariance. Whitaker et al. (2008) showed that EnKF gives better predictions when compared to 3D Var when using the same observations. Changes made to the Canadian perturbed EnKF method showed that it is comparable to 4D-var joint model especially for processes that become non-linear (Kalnay et al., 2007). 4D-var also requires long assimilation windows to give its best results when compared to ensemble methods such as EnKF. The disadvantage of EnKF is the sampling error which is introduced in the background error due to the low dimensionality of the ensemble but this can be corrected using tuning methods such as localisation (see section 4.7). Both methods can take asynchronous observational data (Sakov et al., 2010) and use nonlinear operators.

Number of Observations

Having an optimum number of observations for DA is very important to make accurate predictions. In many cases however it is difficult to have control on the number of observations provided, so it becomes very important to study how DA behaves when given not only suboptimal observations, but also a suboptimal sample of observations. Variations of EnKF such as the Ensembles square root filter (EnSRF) are very efficient with a low number of observations and become less efficient at high numbers of observations (Houtekamer and Zhang, 2016). Local Ensemble Kalman Filter (LEKF) and Local Ensemble Transform Filter (LETKF) methods only assimilate the local points of the grid this makes them more ideal when it comes to high numbers observations(Houtekamer and Zhang, 2016). For accurate prediction Var method needs a long assimilation window (Kalnay et al., 2007) to observe the data. This is disadvantageous in situation where observations are hard to obtain without being intrusive on the experiment.

Errors

Both 4D-Var and EnKF have different approaches to errors in the model and observation. In 4D-Var the background error covariance matrix does not evolve with the model and a weak-constraint is used to account for model errors. As mentioned before 4D-Var relies on linearised and adjoint versions of numerical model with only the first moment of the analysis being calculated.

EnKF, on the other hand, calculates its model errors statistics using the ensemble and does not require a linearised or adjoint versions of the model or the observations. EnKF is also designed to produce an ensemble of analysis which contains the analysis error and can be cycled to future analysis time. As the error calculation relies on the ensemble size, a small ensemble can lead to a sampling error hence the statistics will be suboptimal. This can be seen as noisy correlation and small variance. Various methods can be used to overcome these issues such as inflation and localisation which are discussed in section (4.7).

Conclusion
Kalnay et al. (2007) has a good discussion of the advantages and disadvantages of each of the method. But as mentioned in this section EnKF seems to have an advantage on 4D-Var methods for nonlinear systems. It also has an advantage when the number of observations are low and has multiple tuning methods that can be used to calculate accurate errors in the system. When it comes to coding EnKF is much superior, being the easiest to implement and code. For these reasons the ensemble method and more specifically EnKF was selected to be used for the study presented in this thesis.

4.4 Kalman Filtering

The Kalman filter (Kalman, 1960; Kalman and Bucy, 1961) has been in use for a long time for DA in areas of guidance, navigation and control, etc. But implementation for NWP was very limited due to the size of the models generally used in the applications. Modifications of the Kalman filter such as the Extended Kalman filter (EKF)(Smith et al., 1962) and Ensemble Kalman filter (EnKF)(Evensen, 1994) were proposed to help tackle non-linear problems such as in NWP and oceanography. Since the introduction of EnKF applied to many different areas of research as reported in Evensen (2003). This has also lead to modifications to EnKF as well which are used in different situations. As with other DA methods Kalman filtering compromises of two steps,

1. a forecast step that advances the computational model,

2. an analysis step which combines the forecast, the observations and error statistics to create the optimal state to reinitialise the model.

The analysis step can be run at user defined or error dependent intervals to obtain optimal conditions for the forecast. The weights used to correct the forecast are calculated using the covariance and are usually talked about in terms of the Kalman gain (K). The next few sections give the brief introduction to three different versions of the Kalman filter that have been used for a number of years.

4.4.1 The Kalman Filter (KF)

Kalman filter (KF) (Kalman, 1960; Kalman and Bucy, 1961) is a sequential filter method where a linear model is integrated forward in time along with a linear forward observation operator. Noise is added to the model data to make them stochastic and to make up for any assumption in the model. The Kalman filter combines the model predictions with observations using the Kalman gain. This process is carried out at every step with the previous estimate and Kalman gain being used as information in the following step. Hence the Kalman filter needs only the previous and the current estimate to get the new state instead of the entire history of the model. Due to the linear nature of the model and observation operator, problems start to occur when studying nonlinear behaviour.

4.4.2 Extended Kalman Filter (EKF)

To tackle problems with nonlinear behaviour the Extended Kalman filter was created (Smith et al., 1962). Although the working of EKF is the same as normal KF, EKF can use nonlinear models by using Taylor expansion to extend the statistical model to the first order. The truncation of the model to the first order, however, leads to suboptimal predictions leading to the model covariance being underestimated. This is due to the linearisation not being able to fully model the statistics of the nonlinear behaviour. Higher order EKF has been presented but does not provide any performance benefits (Einieke, 2012).

4.4.3 Ensemble Kalman Filter (EnKF)

The Ensemble Kalman Filter (EnKF) (Evensen, 1994) can use a fully nonlinear model making it ideal for studies with NWP. Unlike the normal Kalman Filter where the system is corrected at every timestep, in EnKF an ensemble of models run for a certain number of timesteps before the current predictions and observations are used to find the Kalman gain and obtain new predictions. Hence unlike other data assimilation systems, EnKF does not need to look at the history of the model to make predictions, giving it a distinct advantage when it comes to real life problem where observation might be scarce in time and space.

The system for EnKF can be given as follows

$$X^f = M(x_N), \tag{4.3}$$

$$Y = y + R, \tag{4.4}$$

Here the forecast X^f is given by model M, with variable x and the members of ensemble N. The observation Y is given by variable y and an error of R. Using the ensemble, a covariance can be calculated and used as the error in the forecast, P, the ensemble is treated as random vectors and the covariance is defined as

$$P = \frac{(X_i^f - \overline{X}^f) \cdot (X_j^f - \overline{X}^f)^T}{N-1}, \tag{4.5}$$

here $\overline{X}^f$ is the ensemble average. The Kalman gain (K) can now be calculated with the use of the background error covariance matrix (P) and the observational error (R). This is given by

$$K = PH^T \cdot (R + HPH^T)^{-1}, \tag{4.6}$$

here H is a observation matrix which maps the model to the observation. Finally using the Kalman gain the equations for the analysis (X^a) can be solved giving the new initial conditions to be used in the model

$$X^a = X^f + K \cdot (Y - HX^f), \tag{4.7}$$

Variations of EnKF

There a few different versions of EnKF that have been developed since its inception. The two main versions are the Stochastic Ensemble Kalman filter and the Deterministic Ensemble Kalman filter. The stochastic method treats the observations as random variables (Burgers et al., 1998; Houtekamer and Mitchell, 1998), where random noise is added to create an ensemble of observations. To try and maintain a good spread of the ensemble random noise is also added to the ensemble analysis.

The deterministic method (Sakov and Oke, 2008; Sakov et al., 2010) was created as it was realised that the spurious correlation between the background error and observational error can lead to degeneration of the quality of the analysis. This lead to a creation of a few flavours of EnKF such as the Ensemble Square Root Filter (EnSRF) and the Ensemble Transform Kalman filter (ETKF). In these filters the observation is not randomly perturbed as in the stochastic version, a few other changes are made depending on the method such as a tunable parameter used for the Kalman gain in EnSRF. And a way to inflate the analysis depending on the covariance values (Sakov et al., 2010). All versions of EnKF can use tuning methods such as localisation and inflation to overcome issues relating to ensemble sizes and error, more on this in section (4.7).

4.5 Formulation

This section gives simple examples for the formulation of the Ensemble Kalman filter. Before going into the formulation some simple statistical terms need to be defined. These terms will be useful later on for all the examples.

Given a model x which is run N number of times to create an ensemble $(x_1, x_2, \ldots, x_N)$, the average for the ensemble can be defined as,

$$\mu = \overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{4.8}$$

here μ and $\overline{x}$ denote the average. Now the standard deviation (σ) can be defined,

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}, \tag{4.9}$$

further the variance (σ^2) of the system can be defined,

$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(x_i - \overline{x}\right)^2, \tag{4.10}$$

and finally the covariance (C_{xy}) between two variables x and y can be defined,

$$C_{xy} = \sigma^2(x,y) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y}), \tag{4.11}$$

Note the $N-1$, due to ensemble being treated as random variables.

Looking at probability distribution of the results it is assumed that they are Gaussian and are given by a Gaussian distribution formula

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]. \tag{4.12}$$

4.5.1 Simple example

Using the statistics defined in above a simple formulation of data assimilation can be worked out. The aim of data assimilation is to obtain a good estimate for the true state, ψ^t, of the system. Both the model forecast, ψ^f, and observation, d, can be thought of as the true state with an error associated with them. Using this train of thought a few equations can be defined to see how the model and observations are related to the true state as shown below,

$$\begin{aligned}\psi^f &= \psi^t + p^f,\\ d &= \psi^t + \epsilon,\end{aligned} \tag{4.13}$$

where ϵ is the error in the observation and p^f is the error model/forecast. Here an assumption is made that the error distributions are Gaussian and the observation and model error are uncorrelated. Using this assumption a few statistics can be defined about the errors,

$$\begin{gathered}\overline{p^f} = 0, \qquad \overline{\epsilon} = 0, \qquad \overline{\epsilon p^f} = 0,\\ \overline{(p^f)^2} = C^f_{\psi\psi}, \qquad \overline{(\epsilon)^2} = C_{\epsilon\epsilon},\end{gathered} \tag{4.14}$$

here the overbar ($\overline{a}$) denotes averaging over the sample/ensemble and $C^f_{\psi\psi}$, $C_{\epsilon\epsilon}$ refer to error covariance of the ensemble and observations respectively. Two equations can now be written for the analysis state, ψ^a; one where the forecast and observations are combined using a linear estimator and the second which shows the relation between the analysis and the true state,

$$\psi^a = \alpha_1 \psi^f + \alpha_2 d, \tag{4.15}$$
$$\psi^a = \psi^t + p^a, \tag{4.16}$$

here α_1 and α_2 denote weights which determine how the model and measurement are combined. p^a denotes the error in the analysis after the combination. Similar to before it is assumed that the distribution is Gaussian, so the error in the analysis, p^a, has similar characteristics to before,

$$\overline{p^a} = 0, \qquad \overline{(p^a)^2} = C^a_{\psi\psi}. \tag{4.17}$$

To get a complete understanding the values for the weights, α_1, α_2, need to be worked out. For this, first the model and observations equations (4.13) are substituted into the analysis equations (4.15) giving,

$$\begin{aligned} \psi^t + p^a &= \alpha_1(\psi^t + p^f) + \alpha_2(\psi^t + \epsilon), \\ &= (\alpha_1 + \alpha_2)\psi^t + \alpha_1 p^f + \alpha_2 \epsilon. \end{aligned} \tag{4.18}$$

The like terms can be grouped together to give two equations. First giving a relationship between the weights and the true state and the second giving the weights and the errors in the system,

$$\psi^t = (\alpha_1 + \alpha_2)\psi^t, \tag{4.19}$$
$$p^a = \alpha_1 p^f + \alpha_2 \epsilon. \tag{4.20}$$

Looking at the relationship between the true state and the weights in equation (4.19), a relation between the two weights can be formulated,

$$\alpha_1 + \alpha_2 = 1, \qquad or \qquad \alpha_1 = 1 - \alpha_2, \tag{4.21}$$

this relation between the two weights is substituted in the analysis equation (4.15) to give,

$$\begin{aligned} \psi^a &= (1 - \alpha_2)\psi^f + \alpha_2 d, \\ &= \psi^f + \alpha_2(d - \psi^f), \end{aligned} \tag{4.22}$$

a relation for the analysis error equation can also be formed using the weights,

$$\begin{aligned} p^a &= p^f(1-\alpha_2)+\alpha_2\epsilon, \\ &= p^f+\alpha_2(\epsilon-p^f). \end{aligned} \tag{4.23}$$

Using this analysis error equation the value of α_2 can now be formulated by first using the definition for covariance,

$$\begin{aligned} p^a &= p^f+\alpha_2(\epsilon-p^f), \\ \overline{(p^a)^2} = C^a_{\psi\psi} &= \overline{(p^f+\alpha_2(\epsilon-p^f))^2}, \qquad (4.24) \\ &= \overline{(p^f)^2}+2\alpha_2\overline{p^f(\epsilon-p^f)}+\overline{\alpha_2^2(\epsilon^2-2\epsilon p^f+(p^f)^2)}, \\ \therefore C^a_{\psi\psi} &= C^f_{\psi\psi}-2\alpha_2 C^f_{\psi\psi}+\alpha_2^2(C_{\epsilon\epsilon}-C^f_{\psi\psi}), \qquad (4.25) \end{aligned}$$

now the equation is differentiated and minimised,

$$\frac{dC^a_{\psi\psi}}{d\alpha_2} = -2C^f_{\psi\psi}+2\alpha_2(C_{\epsilon\epsilon}-C^f_{\psi\psi}) = 0, \tag{4.26}$$

finally the value of α_2 can be easily calculated by rearranging the equation giving,

$$\alpha_2 = \frac{C^f_{\psi\psi}}{C_{\epsilon\epsilon}-C^f_{\psi\psi}}. \tag{4.27}$$

Using this result, the analysis equation (4.15) and the error covariance equation (4.25) can be can be rewritten to give the final equations for DA,

$$\psi^a = \psi^f+\frac{C^f_{\psi\psi}}{C_{\epsilon\epsilon}-C^f_{\psi\psi}}(d-\psi^f), \tag{4.28}$$

$$C^a_{\psi\psi} = C^f_{\psi\psi}\left(1-\frac{C^f_{\psi\psi}}{C_{\epsilon\epsilon}-C^f_{\psi\psi}}\right), \tag{4.29}$$

here the first equation shows how the model, the observation and the covariance can be combined to get a new analysis state. The second equation shows how the error in the system can be tracked by just using the covariance. Note the resemblance to equations to (4.6) and (4.7).

4.5.2 Vector example and KF formulation

Using the previous simple example, the formulation can updated for a system with multiple vectors using a Bayesian approach. Lets assume that the number of measurements/observations (y_{obs}) and the model forecast can be related to the model (ψ) the same way it was done for the simple example. This relation is given by the equations

$$\begin{aligned} \psi^f &= \psi + p^f, \\ y_{obs} &= H\psi + \epsilon, \end{aligned} \tag{4.30}$$

where H is a linear map/matrix which maps the model grid points to the observed points. This matrix is of a size/shape $n_\psi \times p_y$, where n and p are the number of observation vectors (p_y) and model vectors (n_ψ). As before the errors in the observation and model are given by ϵ and p^f respectively.

Given a probability density function for the forecast ($f(\psi)$), the observation ($f(y_{obs})$) and a likelihood function $f(y_{obs}|\psi)$, Bayes' theorem can be used to find the probability of ψ given a measurement y_{obs},

$$f(\psi|y_{obs}) = \frac{f(\psi)f(y_{obs}|\psi)}{f(y_{obs})}. \tag{4.31}$$

Here $f(\psi)$ shows our understanding of the distribution of ψ before observing y_{obs} and is called the *prior*. $f(\psi|y_{obs})$ gives the distribution of ψ given the observation of y_{obs} and is called the *posterior*. $f(y_{obs}|\psi)$ describes the likelihood of the value of y_{obs} given the value of ψ and hence is called the *likelihood function*. Finally $f(y_{obs})$ is a normalisation factor and can be evaluated as a post-processing step. This means that equation (4.31) can be simplified to

$$f(\psi|y_{obs}) \propto f(\psi)f(y_{obs}|\psi). \tag{4.32}$$

Assuming the distribution is Gaussian, the probability distribution functions for the *prior* and the *likelihood* can be defined as

$$f(\psi) \propto exp\left(-\frac{1}{2}(\psi - \psi^f)(C^f_{\psi\psi})^{-1}(\psi - \psi^f)\right), \tag{4.33}$$

$$f(y_{obs}|\psi) \propto exp\left(-\frac{1}{2}(H\psi - y_{obs})(C_{\epsilon\epsilon})^{-1}(H\psi - y_{obs})\right), \tag{4.34}$$

where $C^f_{\psi\psi}$ and $C_{\epsilon\epsilon}$ are the covariance matrix for the forecast and observation, and H is the matrix that maps the forecast to the observations as described in the previous section.

These equations can be substituted in the Bayes' equation to give

$$f(\psi|y_{obs}) \propto \exp\Bigg(-\frac{1}{2}(\psi - \psi^f)(C^f_{\psi\psi})^{-1}(\psi - \psi^f)\Bigg) \\ + \exp\Bigg(-\frac{1}{2}(\psi - y_{obs})(C_{\epsilon\epsilon})^{-1}(\psi - y_{obs})\Bigg). \quad (4.35)$$

Before solving the above equation it can first be simplified using the completing-the-square formula

$$x^T Vx - 2d^T x = (x - V^{-1}d)^T V(x - V^{-1}d) - d^T V^{-1}d, \quad (4.36)$$

the integrand for $f(\psi|y_{obs})$ can now be reformulated.

$$I = -\frac{1}{2}\left[(\psi - \psi^f)(C^f_{\psi\psi})^{-1}(\psi - \psi^f) + (\psi - y_{obs})(C_{\epsilon\epsilon})^{-1}(\psi - y_{obs})\right], \quad (4.37)$$

Using a substitute of

$$V = C^{-1}_{\psi\psi} + H^T C^{-1}_{\epsilon\epsilon} H, \qquad d = H^T C^{-1}_{\epsilon\epsilon} y_{obs} + C^{-1}_{\psi\psi}\psi^f, \quad (4.38)$$

the integrand can be rewritten as

$$I = (\psi - V^{-1}d)^T V(\psi - V^{-1}d) - d^T V^{-1}d + y^T C^{-1}_{\epsilon\epsilon} y_{obs} + \psi^{fT} C^{-1}_{\psi\psi}\psi^f. \quad (4.39)$$

Using this an equation for covariance matrix and the analysis equation can be written as

$$\begin{aligned} C^a_{\psi\psi} &= V^{-1} = (C^{-1}_{\psi\psi} + H^T C^{-1}_{\epsilon\epsilon} H)^{-1}, \\ \psi^a &= \psi^f - C^a_{\psi\psi} H^T C^{-1}_{\epsilon\epsilon}(H\psi^f - y_{obs}), \end{aligned}$$

and this can be then rewritten for the system which contains both forecast and analysis

$$(C^a_{\psi\psi})^{-1} = (C^f_{\psi\psi})^{-1} + H^T C^{-1}_{\epsilon\epsilon} H, \quad (4.40)$$

$$\psi^a = \psi^f - C^a_{\psi\psi} H^T C^{-1}_{\epsilon\epsilon}(H\psi^f - y_{obs}). \quad (4.41)$$

Using the Sherman-Morrison-Woodbury matrix inversion formula

$$(M + U^T NU)^{-1} = M^{-1} - M^{-1}U^T\left(N^{-1} + UM^{-1}U^T\right)^{-1} UM^{-1}, \quad (4.42)$$

the analysis covariance matrix $\left((C^a_{\psi\psi})^{-1}\right)$ can be reformulated using $M = (C^f_{\psi\psi})^{-1}$ and $N = C^{-1}_{\epsilon\epsilon}$ and $U = H$ giving

$$\begin{aligned}
C^a_{\psi\psi} &= \left((C^f_{\psi\psi})^{-1} + H^T C^{-1}_{\epsilon\epsilon} H\right)^{-1}, && (4.43)\\
&= C^f_{\psi\psi} - C^f_{\psi\psi} H^T (H C^f_{\psi\psi} H^T + C_{\epsilon\epsilon})^{-1} H C^f_{\psi\psi}, && (4.44)\\
&= C^f_{\psi\psi} - K H C^f_{\psi\psi}, && (4.45)
\end{aligned}$$

where the Kalman gain (K) matrix was defined as

$$K = C^f_{\psi\psi} H^T (C_{\epsilon\epsilon} + H C^f_{\psi\psi} H^T)^{-1}. \tag{4.46}$$

Now a complete equation for DA can be formulated. First just looking at the covariance section in equation (4.41), the value of $C^a_{\psi\psi}$ from equation (4.45) can be substituted giving,

$$\begin{aligned}
C^a_{\psi\psi} H^T C^{-1}_{\epsilon\epsilon} &= (C^f_{\psi\psi} + K H C^f_{\psi\psi}) H^T C^{-1}_{\epsilon\epsilon}\\
&= (C^f_{\psi\psi} + (C^f_{\psi\psi} H^T (C_{\epsilon\epsilon} + H C^f_{\psi\psi} H^T)^{-1}) H C^f_{\psi\psi}) H^T C^{-1}_{\epsilon\epsilon}\\
&= C^f_{\psi\psi} H^T (I - (H C^f_{\psi\psi} H^T + C_{\epsilon\epsilon})^{-1} H C^f_{\psi\psi} H^T) C^{-1}_{\epsilon\epsilon},\\
&= C^f_{\psi\psi} H^T (C_{\epsilon\epsilon} + H C^f_{\psi\psi} H^T)^{-1},\\
\therefore C^a_{\psi\psi} H^T C^{-1}_{\epsilon\epsilon} &= K
\end{aligned}$$

hence the analysis equation can finally be written in terms of the Kalman gain giving the usual equation

$$\psi^a = \psi^f - K(H\psi^f - y_{obs}) \tag{4.47}$$

where the Kalman gain is calculated as

$$K = C^f_{\psi\psi} H^T (C_{\epsilon\epsilon} + H C^f_{\psi\psi} H^T)^{-1}. \tag{4.48}$$

Again note the resemblance to equations to (4.6) and (4.7).

4.6 EnKF analysis scheme

Now that the formulation for EnKF has been described lets look at how it can be implemented practically. The implementation of the EnKF algorithm can be slightly tricky as the data read into the system to calculate the analysis can become quite large, especially when multidimensional models are used. So it is key to manipulate the data in such a way that when used in a code the RAM usage is kept low. A way to implement the scheme was given in Evensen (2003). This section looks at how EnKF was implemented in this study.

Before going through the scheme a few variables need to be defined which will be used throughout the study

- n - total number of vector points in the model
- N - ensemble size
- p - total number of observed points

At the beginning of the analysis scheme, the forecast X of shape $n \times N$ is read into the system. The observation Y of shape p is read into the system and changed to a shape of $p \times N$ by adding noise. The ensemble is subjected to the observation matrix H which is of shape $p \times n$. The observation matrix maps the ensemble to create a matrix HX of shape $p \times N$. The average of both these ensembles are then calculated and stored as $\overline{X}$ and $\overline{HX}$ to be used to calculate statistics. The average in both the matrices are repeated to create a matrix which correspond to their respective original ensemble, i.e. $n \times N$ for $\overline{X}$ and $p \times N$ for $\overline{HX}$. Using this the two variance are calculated

$$\begin{aligned} A &= X - \overline{X}, \\ HA &= HX - \overline{HX}, \end{aligned} \tag{4.49}$$

here A is the variance in the whole model and HA is the variance in the mapped model. Now the background error matrix can be calculated

$$\begin{aligned} HPHT &= (HA \cdot HA)/(N-1), \\ PHT &= (A \cdot HA)/(N-1), \end{aligned} \tag{4.50}$$

where $HPHT$ is of shape $p \times p$ and is the background error of the observed points and PHT is of shape $n \times p$ and is the error in the whole model. Using the observational error R of shape $p \times p$ the inner part of the Kalman gain can be calculated and the matrix can then be inverted

$$Ka = (HPHT + R)^{-1}, \tag{4.51}$$

where the shape of the matrix remains the same as before, $p \times p$. Now the full Kalman gain can be calculated

$$Kal = PHT \cdot Ka, \tag{4.52}$$

where the Kalman gain is of shape $n \times p$. Next the difference between the observation and ensemble is calculated

$$dy = Y - HX, \tag{4.53}$$

Name	Symbol	Shape
Observation matrix	H	$p \times n$
Forecast	X^f	$n \times N$
Observation	Y	$p \times N$
Background covariance matrix	P	$n \times n$
Observational error	R	$p \times p$
Analysis	X^a	$n \times N$
Model variance	A	$n \times N$
Mapped model variance	HA	$p \times N$
Mapped background error	$HPHT$	$p \times p$
Background error	PHT	$n \times p$
Inner kalman gain	Ka	$p \times p$
Kalman gain	Kal	$n \times p$
Observation and model difference	dy	$p \times N$
Inner analysis	dx	$n \times N$

Table 4.1: Shape of matrix for variables used during analysis

this difference is multiplied by the Kalman gain to get the weighted correction for the ensemble

$$dx = Kal \cdot dy, \tag{4.54}$$

where the shape of dx is $n \times N$. The analysis can now be finally calculated using the weight

$$X^a = X + dx, \tag{4.55}$$

where Xa is of shape $n \times N$ and is the new initial state of the ensemble. This ensemble can now be used to reinitialise the model. Table (4.1) gives a summary of the shapes of the different matrices that are used in the analysis step.

4.6.1 Structure of the matrix

The previous section showed how the analysis for EnKF is done and what the shape of the matrices used for analysis are. In this section examples are given of what the structure of the matrices look like when these calculations are done. This is important as the storing of data in the matrices can be tricky especially when dealing with multidimensional data.

In many coding languages such as Python, there are built in commands to multiply 2D matrices. Hence for multidimensional data, matrix reduction needs to be used to create a matrix when can be used for calculations. In the following sections, a simple examples are given followed by how matrix reduction can be used to create a 2D matrix for multidimensional data.

4.6.1.1 Simple example

First lets look at a very simple example where there are n variables, p observed variables and N ensembles. Given a model $\mathbf{X}$ with variables $\mathbf{x_n}$ a 2D matrix can be created to store the data for the ensemble,

$$\mathbf{X} = (\mathbf{x_1}\ \mathbf{x_2}\ \ldots\ \mathbf{x_n}) = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \ldots & x_{1N} \\ x_{21} & x_{22} & x_{23} & \ldots & x_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \ldots & x_{nN} \end{bmatrix} \tag{4.56}$$

The same can be done for an observation matrix $\mathbf{Y}$ with variables variables $\mathbf{Y_n}$ as seen below

$$Y_{obs} = (\mathbf{Y_1}\ \mathbf{Y_2}\ \ldots\ \mathbf{Y_p}) = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & \ldots & Y_{1N} \\ Y_{21} & Y_{22} & Y_{23} & \ldots & Y_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_{p1} & Y_{p2} & Y_{p3} & \ldots & Y_{pN} \end{bmatrix} \tag{4.57}$$

The average can now be easily calculated in the matrix form and kept in a 2D matrix form which will be helpful later on. This can be done using the equation below along with special commands that might be specific for the coding language being used (eg. *numpy.mean* for python),

$$\overline{\mathbf{X}} = \frac{1}{N} \begin{bmatrix} \sum_{a=1}^{N} x_{1a} \\ \sum_{a=1}^{N} x_{2a} \\ \vdots \\ \sum_{a=1}^{N} x_{na} \end{bmatrix} \mathbf{1} = \begin{bmatrix} \overline{x}_1 & \overline{x}_1 & \overline{x}_1 & \ldots & \overline{x}_1 \\ \overline{x}_2 & \overline{x}_2 & \overline{x}_2 & \ldots & \overline{x}_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \overline{x}_n & \overline{x}_n & \overline{x}_n & \ldots & \overline{x}_n \end{bmatrix} \tag{4.58}$$

Using the average matrix the variance can be calculated by simple subtraction

$$\mathbf{X}' = \mathbf{X} - \overline{\mathbf{X}} = (x_1'\ x_2'\ \ldots x_n') = \begin{bmatrix} x_{11} - \overline{x}_1 & x_{12} - \overline{x}_1 & \ldots & x_{1N} - \overline{x}_1 \\ x_{21} - \overline{x}_2 & x_{22} - \overline{x}_2 & \ldots & x_{2N} - \overline{x}_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} - \overline{x}_n & x_{i2} - \overline{x}_n & \ldots & x_{nN} - \overline{x}_n \end{bmatrix} \tag{4.59}$$

The covariance can be calculated using dot products and a transpose of the variance matrices

$$\begin{aligned} \mathbf{P} &= \frac{1}{N-1}(x-\overline{x})\cdot(x-\overline{x})^T, \\ \mathbf{P} &= \frac{1}{N-1}\mathbf{X}'\cdot(\mathbf{X}'), \end{aligned} \tag{4.60}$$

The creation of the observation matrix H which is a $p \times n$ matrix can be done easily

$$H = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \tag{4.61}$$

The observation error matrix R which is a $p \times n$ matrix is also created the same way,

$$R = \begin{bmatrix} r & 0 & \dots & 0 \\ 0 & r & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r \end{bmatrix} \tag{4.62}$$

Remember the observation errors are uncorrelated to error of one variable has to relationship with another. Using the 2D matrices defined above the rest of the analysis can be easily calculated as described in the previous section.

4.6.1.2 2D Grid example

Having a multidimensional model can quickly create confusion on how to create the 2D matrices need for analysis. The multidimensional model has multiple grid points and variables, and multiple variables could be stored at each grid point. To overcome this problem a matrix reduction method needs to be used. Lets define i, j to be the positions on a 2D grid, the number of variables as n and N as the ensemble number. As before, given a model $\mathbf{X}$ with variables $\mathbf{x_n}$, the stored matrix data can be a bit hierarchical,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_{11}} & \mathbf{x_{12}} & \mathbf{x_{13}} & \dots & \mathbf{x_{1j}} \\ \mathbf{x_{21}} & \mathbf{x_{22}} & \mathbf{x_{23}} & \dots & \mathbf{x_{2j}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x_{i1}} & \mathbf{x_{i2}} & \mathbf{x_{i3}} & \dots & \mathbf{x_{ij}} \end{bmatrix} = \begin{bmatrix} [x]_{11}^{nN} & [x]_{12}^{nN} & \dots & [x]_{1j}^{nN} \\ [x]_{21}^{nN} & [x]_{22}^{nN} & \dots & [x]_{2j}^{nN} \\ \vdots & \vdots & \ddots & \vdots \\ [x]_{i1}^{nN} & [x]_{i2}^{nN} & \dots & [x]_{ij}^{nN} \end{bmatrix} \tag{4.63}$$

where

$$[X]_{11}^{nN} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1N} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{nN} \end{bmatrix} \tag{4.64}$$

here each position has its own matrix which contains data for each variable for the ensemble. This multidimensional matrix needs to be reduced to get a 2D matrix.

Lets work with a simple example where there is a 2×2 grid with variables U,V,W at each grid point and ensemble of 3. So the total number of vectors are $n_{tot} = n{*}i{*}j = 12$. For this system the matrix is reduced to a 2D system where each row corresponds to each of the grid points and each column corresponds to an ensemble. Further, each variable needs to be stored in the matrix sequentially, with all the values of U being stored first followed by V and finally W. This ends up giving a 2D matrix that can be used in the EnKF scheme. First lets look at how the data is stored in the ensemble and the average matrix which are of shape $ntot \times N$. Notice that all that has been done in essence is a redefinition of n being n_{tot},

$$\mathbf{X} = \begin{bmatrix} U_{11}^1 & U_{11}^2 & U_{11}^3 \\ U_{12}^1 & U_{12}^2 & U_{12}^3 \\ U_{21}^1 & U_{21}^2 & U_{21}^3 \\ U_{21}^1 & U_{22}^2 & U_{22}^3 \\ V_{11}^1 & V_{11}^2 & V_{11}^3 \\ V_{12}^1 & V_{12}^2 & V_{12}^3 \\ V_{21}^1 & V_{21}^2 & V_{21}^3 \\ V_{22}^1 & V_{22}^2 & V_{22}^3 \\ W_{11}^1 & W_{11}^2 & W_{11}^3 \\ W_{12}^1 & W_{12}^2 & W_{12}^3 \\ W_{21}^1 & W_{21}^2 & W_{21}^3 \\ W_{22}^1 & W_{22}^2 & W_{22}^3 \end{bmatrix}, \overline{\mathbf{X}} = \begin{bmatrix} \overline{U}_{11} & \overline{U}_{11} & \overline{U}_{11} \\ \overline{U}_{12} & \overline{U}_{12} & \overline{U}_{12} \\ \overline{U}_{21} & \overline{U}_{21} & \overline{U}_{21} \\ \overline{U}_{21} & \overline{U}_{22} & \overline{U}_{22} \\ \overline{V}_{11} & \overline{V}_{11} & \overline{V}_{11} \\ \overline{V}_{12} & \overline{V}_{12} & \overline{V}_{12} \\ \overline{V}_{21} & \overline{V}_{21} & \overline{V}_{21} \\ \overline{V}_{22} & \overline{V}_{22} & \overline{V}_{22} \\ \overline{W}_{11} & \overline{W}_{11} & \overline{W}_{11} \\ \overline{W}_{12} & \overline{W}_{12} & \overline{W}_{12} \\ \overline{W}_{21} & \overline{W}_{21} & \overline{W}_{21} \\ \overline{W}_{22} & \overline{W}_{22} & \overline{W}_{22} \end{bmatrix} \tag{4.65}$$

Using these the variance and covariance can be easily calculate using a dot product

$$\mathbf{X}' = \mathbf{X} - \overline{\mathbf{X}}$$

$$\mathbf{P} = \frac{1}{N-1}\mathbf{X}'(\mathbf{X}')^T$$

$$\mathbf{P} = \begin{bmatrix} U^{1\prime}_{11} & U^{2\prime}_{11} & U^{3\prime}_{11} \\ U^{1\prime}_{12} & U^{2\prime}_{12} & U^{3\prime}_{12} \\ U^{1\prime}_{21} & U^{2\prime}_{21} & U^{3\prime}_{21} \\ U^{1\prime}_{21} & U^{2\prime}_{22} & U^{3\prime}_{22} \\ V^{1\prime}_{11} & V^{2\prime}_{11} & V^{3\prime}_{11} \\ V^{1\prime}_{12} & V^{2\prime}_{12} & V^{3\prime}_{12} \\ V^{1\prime}_{21} & V^{2\prime}_{21} & V^{3\prime}_{21} \\ V^{1\prime}_{22} & V^{2\prime}_{22} & V^{3\prime}_{22} \\ W^{1\prime}_{11} & W^{2\prime}_{11} & W^{3\prime}_{11} \\ W^{1\prime}_{12} & W^{2\prime}_{12} & W^{3\prime}_{12} \\ W^{1\prime}_{21} & W^{2\prime}_{21} & W^{3\prime}_{21} \\ W^{1\prime}_{22} & W^{2\prime}_{22} & W^{3\prime}_{22} \end{bmatrix} \cdot \begin{bmatrix} U^{1\prime}_{11} & \dots & U^{1\prime}_{22} & V^{1\prime}_{11} & \dots & V^{1\prime}_{22} & \dots & W^{1\prime}_{22} \\ U^{2\prime}_{11} & \dots & U^{2\prime}_{22} & V^{2\prime}_{11} & \dots & V^{1\prime}_{22} & \dots & W^{2\prime}_{22} \\ U^{3\prime}_{11} & \dots & U^{3\prime}_{22} & V^{3\prime}_{11} & \dots & V^{1\prime}_{22} & \dots & W^{3\prime}_{22} \end{bmatrix}$$

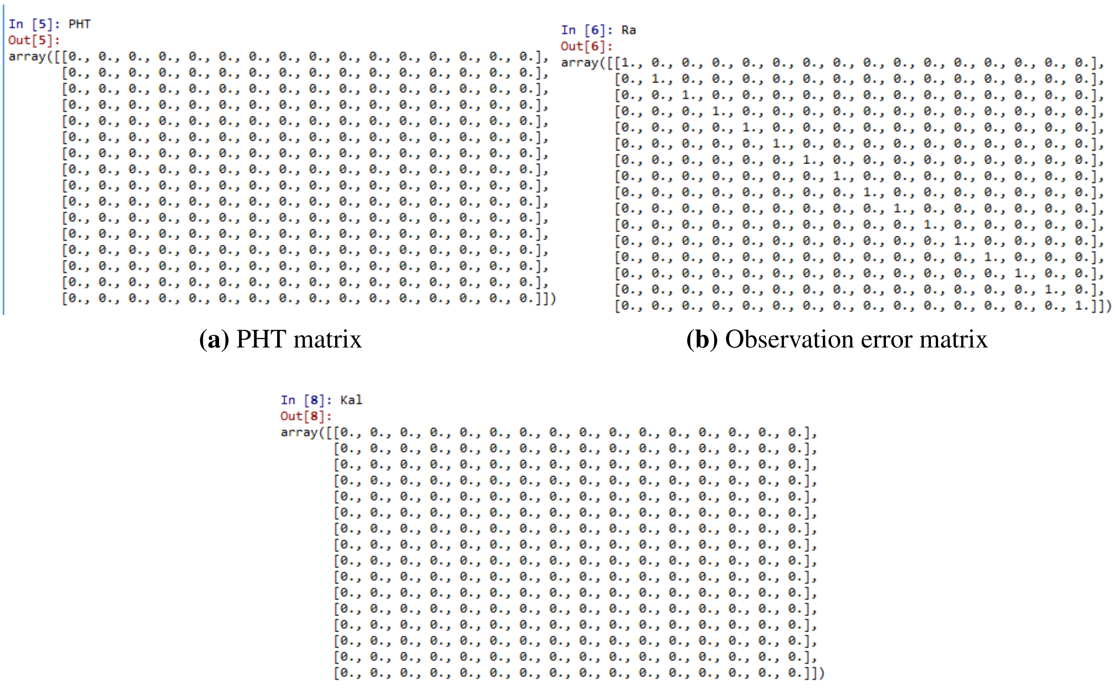
Here each row gives the covariance for a variable at the point with all the other points in the system. Now H becomes a big matrix

$$H = \begin{bmatrix} 1 & 0 & 0 & \dots & & & & \\ 0 & 1 & 0 & 0 & \dots & & & \\ 0 & 0 & 1 & 0 & \dots & & & \\ & \dots & 0 & 1 & 0 & \dots & & \\ & & \dots & 0 & 1 & 0 & \dots & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & \dots & 0 & 0 & 0 \\ & & & & & & 0 & 0 \end{bmatrix}$$

Same idea with R

$$R = \begin{bmatrix} r & 0 & 0 & \dots & & & & \\ 0 & r & 0 & 0 & \dots & & & \\ 0 & 0 & r & 0 & \dots & & & \\ & \dots & 0 & r & 0 & \dots & & \\ & & \dots & 0 & r & 0 & \dots & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & \dots & 0 & r & 0 \\ & & & & & & 0 & r \end{bmatrix}$$

The EnKF scheme can now be used to solve for the analysis. This implementation can be used for all multidimensional models.



```
In [5]: PHT
Out[5]:
array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

(a) PHT matrix

```
In [6]: Ra
Out[6]:
array([[1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.]])
```

(b) Observation error matrix

```
In [8]: Kal
Out[8]:
array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

(c) Kalman gain matrix

Figure 4.2: Test for Kalman Gain

4.6.2 Testing the analysis code

An easy way to make sure that the scheme is working properly when coded would be to use an ensemble which should result in the Kalman gain having a value of zero and hence no corrections are done, that means eq (4.7) is just $X^a = X^f$. For this test an ensemble of ones is created and when the variance for the ensemble is calculated it is just a matrix of zeros. It must be taken into account that there is a matrix inversion when calculating the Kalman gain (eq. 4.6), so not everything can be zero otherwise the code will crash. To over come this problem the observation error matrix was chosen as a diagonals of ones.

The calculation using the above mentioned matrices are shown in figure (4.2), where values for PHT, Ra and Kal matrices can be seen. As can be observed the PHT matrix is zero and the observation error matrix is a diagonals of one. Finally the calculated Kalman gain matrix is zero leading to no correction being done showing that the code works as it is suppose to.

4.7 Limitations of EnKF and Tuning Parameters

As data obtained in many cases from both the models and observations are imperfect, the data assimilation methods need to be able to adapt to these imperfections. In the case of EnKF, it is assumed that all distributions are Gaussian, but this assumption only holds for an infinite number of ensembles and observations (Evensen, 1994). This not feasible in real life, hence over the years since its introduction, a few tuning parameters have been developed for ENKF along with a few variations of the method which can be used depending on the situation. Some of these limitations of ENKF are described now along with methods that have been developed to overcome them.

4.7.1 Localisation

The ensemble size has a big impact on the time taken to get a prediction. The bigger the ensemble the longer it will take to run the model. Hence many studies try and keep the ensemble size small, for example in atmospheric applications the ensemble size is around $N = 100$, while the number of vectors can be around $O(10^8)$ (Houtekamer and Zhang, 2016). As the ensembles are used to calculate the background error it is very important to have a good sample size. When vectors in the system $n_{ens} << n_{model}$ and $n_{ens} << p_{obs}$, this is commonly known as the rank problem. Here the under representation of the model can lead to a spurious correlation in the covariance matrix and hence inaccurate analysis.

A unique solution to solve this rank problem is to split the data into independent local problems. This method is called localisation. The idea is that points far away from an observed grid point do not influence behaviour, hence when finding covariance they should be ignored (for example in NWP the weather in Australia does not immediately impact the weather in Europe). The implementation of localisation is considerably ad hoc and it is largely believed that the bigger the ensemble the less severe the covariance localisation needs to be (Houtekamer and Mitchell, 1998).

For the localisation correlation function ρ, it has become common to use a fifth order piecewise rational function given in equation (4.10) in Gaspari and Cohn (1999). However, some experimentation is necessary to find out an optimal length scale for the localization function. The correlation function can then be implemented in the Kalman gain equation (4.6) to give

$$K = [\rho \circ (PH^T)] \cdot [R + \rho \circ (HPH^T)]^{-1}, \tag{4.66}$$

here the localisation correlation function ρ is multiplied element-wise (Schur product) with the covariance matrix. This essentially creates a sparce matrix for the Kalman gain.

4.7.2 Inflation - multiplicative and additive

As mentioned previously, in EnKF it is assumed that the distribution is Gaussian but this is only true if an infinite number of ensembles are used ($N \to \infty$). So when only a limited number of ensembles are used the spread can become a problem. A well spread ensemble is needed to sample enough of the state space to generate an accurate error for the background. But with small ensembles it is hard to have well spread data. This can lead to under-correction of the ensemble leading them to converge. This convergence can lead to the ensemble occupying the same state space and never being able to spread out and hence not being able to give accurate predictions. There are two main ways to increase the spread of the ensemble or inflate the ensemble. The first being additive perturbation/inflation and the second being multiplicative perturbation/inflation.

Additive inflation essentially leads to adding a perturbation to the ensemble after analysis. Using a simple static background error covariance a simple random field can be created which can be added on to the model. Houtekamer et al. (1999) used a tunable parameter (α) when creating the perturbation matrix.

$$Q = \alpha P^f \qquad 0 < \alpha < 1, \tag{4.67}$$

$$q \sim n(0, Q), \tag{4.68}$$

$$X^f(t+1) = M(x^a(t)) + q, \tag{4.69}$$

this method has been competitive with other methods used in EnKF. In the studies presented in this thesis however just a simple random field was used to create a perturbation and added to the analysis.

Multiplicative inflation works by increasing the magnitude of the error covariance matrix. The background error covariance is simply multiplied by the tunable parameter γ. When no inflation is needed this tunable factor will be equal to 1. But with smaller ensembles, a bigger value of γ will be needed

$$P^f_{inflated} = \gamma P^f \qquad \gamma \geq 1 \tag{4.70}$$

Over inflation can lead to a covariance growth in data sparse areas and to overcome these issues adaptive inflation algorithms have also been developed as reported by Anderson (2009).

Other methods include splitting the ensemble into sub ensembles during the analysis as reported in Houtekamer and Mitchell (1998); Mitchell and Houtekamer (2009). Here the ensemble is divided in k sub-ensembles of equal size and the Kalman gain from one sub-ensemble is used for the analysis of another sub ensemble.

$$\begin{aligned} X_1 &= X_1 + K_2(Y - HX_1) \\ X_2 &= X_2 + K_1(Y - HX_2) \end{aligned} \tag{4.71}$$

4.8 Conclusion

This chapter presented a general overview of data assimilation and the Ensemble Kalman filter. Different types of data assimilation methods that have been developed over the years were introduced including the big two currently is use, variational methods and ensemble methods. The pros and cons of both these methods were presented when it comes to dealing with observations and error.

A brief introduction to Kalman filtering was given with examples of how EnKF can be formulated. The implementation of the analysis step of EnKF was shown along with the shapes of the matrices used. This was followed by examples of what the structures of these matrices are and how they are used for multidimensional problems. Lastly some of the limitations of EnKF were discussed along with methods that have been developed to help. Houtekamer and Zhang (2016) published a recent review about EnKF in Atmospheric Data Assimilation which provides a lot more details about different aspects of EnKF.

Chapter 5

EnKF and the Lorenz Model

5.1 Introduction

The effectiveness of any Data Assimilation (DA) method can only be ascertained when tested against a model that pushes its limits to give accurate predictions. In many cases, the Lorenz model has been used to fill this niche as its nonlinear behaviour at particular parameters provides a robust testbed to look at the different aspects and limitations of a DA method (Evensen, 1997, 2009; Annan and Hargreaves, 2004). DA can also be used for parameter estimation where, as the name suggests, the parameters become part of the variables in the system and DA is used to predict the values of these parameters (Evensen, 2009). This method is relevant in cases where the parameters for a system are not accurately known (e.g. in experiments).

In the study presented in this chapter, the Ensemble Kalman Filter (EnKF) is tested using the Lorenz model. EnKF, as the name suggests, is an ensemble based Kalman filter DA method which uses a Bayesian approach to obtain an accurate prediction. EnKF has been used in a wide variety of applications from ocean modelling to atmospheric circulation studies. Evensen (2003) reports a review of the wide variety of research done using EnKF.

A sample EnKF code written in MATLAB (see section 5.3.2 for more details) is used for an initial test and to write a Python version of EnKF to test the Lorenz Model. The techniques learned in this chapter were used as a basis for how to apply EnKF to the MORALS code. This code solves the Navier-Stokes equation for a fluid rotating in a thermal annulus in axisymmetric 2D and full 3D. More information about the MORALS code is given in chapter (3) with chapter (7) presenting the results of the twin experiment and chapter (8) presenting the results for EnKF with experimental data.

In this chapter the Lorenz model is presented in section (5.2) followed by the Runge-Kutta method in section (5.3.1), which was used to solve the Lorenz Model. This is followed by the MATLAB setup and the python setup in section (5.3.2) and (5.3.3) respectively. This also outlines how the code works and how it was used. Section (5.3.3) describes the creation and acquisition of observations for the study. Section (5.3.4) explains the use of inflation with section (5.3.5) describing the setup for parameter estimation. The results of the study are presented in section (5.4) for MATLAB and section (5.5) for Python. The Python results are divided into multiple subsections for the different types of testing that was done to ”push” EnKF. This included looking at how the ensemble size plays a role in prediction in section (5.5.2), followed by how inflation can change the accuracy of EnKF in section (5.5.3). Changing the time between assimilation is tested in section (5.5.4), and finally results for parameter estimation are presented in section (5.5.5).

5.2 The Lorenz model

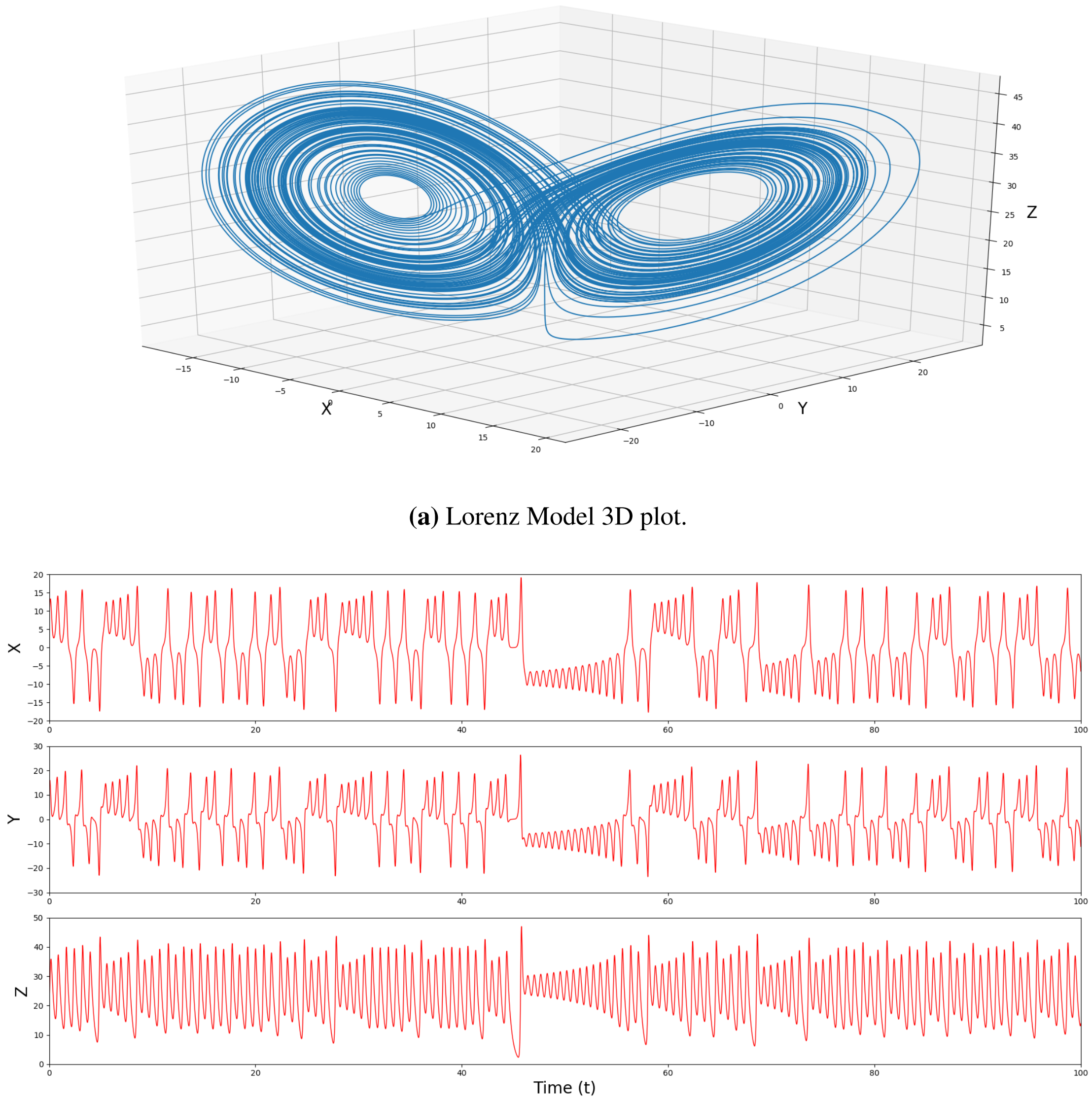
The Lorenz model (Lorenz, 1963) is a set of three ordinary differential equations which give a strongly nonlinear system. The model was conceived to describe the flow in a thermal convection experiment but has since been used in numerous studies related to nonlinear systems. The system is known for its non-periodic/chaotic behaviour and unpredictability for certain parameter values. The system is described in equation (5.1) below,

$$\begin{aligned} \dot{x} &= \sigma(y - x), \\ \dot{y} &= rx - y - xz, \\ \dot{z} &= xy - bz, \end{aligned} \tag{5.1}$$

Here x, y and z are the position variables and $\dot{x}$, $\dot{y}$ and $\dot{z}$ are the first derivatives of the variables with respect to time t. The constants in the equation are given the values of $\sigma = 10$, $r = 28$ and $b = 8/3$. These values were used in all the studies presented in the chapter unless specified otherwise. These values are commonly used to get chaotic behaviour and have been extensively studied and have served as a test-bed for many data assimilation techniques from variational methods to ensemble methods (Evensen, 1997; Miller, 1994). The Lorenz system has two nontrivial fixed points at $z = r - 1 = 27$, $x = y = \pm\sqrt{b(r-1)} = \pm\sqrt{72}$ which are unstable at $r = 28$, but the system has a relatively slow growth rate in their neighbourhood. Figure (5.1) shows a typical example of what the system looks like with these parameter values.

5.3 Method

DA is a complicated undertaking, even when done for a “simple” model such as the Lorenz model. Hence it is understandable that many pieces are needed to bring the



(a) Lorenz Model 3D plot.

(b) Lorenz Model plot showing the behaviour of x, y and z variables.

Figure 5.1: A typical Lorenz model trajectory using $\sigma = 10$, $r = 28$ and $b = 8/3$.

assimilation to fruition. This section tries to explain a few of the parts that were used when applying DA on the Lorenz Model.

First, the Runge-Kutta method is described that was used to solve the Lorenz Model. This is followed by a description of the working of the MATLAB code. This code was used as a test to understand how EnKF can be implemented in code for different models. The Python code is then described which was written using the MATLAB code as a framework. The importance of inflation in EnKF is summarised in section (5.3.4) along with what values were used in the studies. Finally, a description of parameter estimation is given along with how it was implemented in the Python code.

5.3.1 Runge-Kutta Methods

To solve the Lorenz model the Runge-Kutta 4th order method was used. Previous studies used lower order finite difference schemes when solving for the Lorenz model (Evensen, 1997; Miller, 1994). This led to quite a large number of ensembles being needed when trying to find accurate stable solutions using EnKF. With the use of the more stable RK4 method, this number should be much smaller, thus saving on computational cost and time, while still leading to a good understanding of how EnKF works under different conditions.

When solving for a system with equation $\dot{y} = f(t, y)$ with $y(t_0) = y_0$ the 2nd order Runge-Kutta system is given as

$$\begin{aligned} k1 &= f(y_n, t_n) \\ k2 &= f(y_n + k1\frac{dt}{2}, t_n + dt/2) \\ y_{n+1} &= y_n + k2 \times dt \\ t_{n+1} &= t_n + dt \end{aligned} \tag{5.2}$$

while the 4th order is given as

$$\begin{aligned} k1 &= f(y_n, t_n) \\ k2 &= f(y_n + k1 \times \frac{dt}{2}, t_n + dt/2) \\ k3 &= f(y_n + k2 \times \frac{dt}{2}, t_n + dt/2) \\ k4 &= f(y_n + k3 \times dt, t_n + dt/2) \\ y_{n+1} &= y_n + \frac{dt}{6}(k1 + (2 \times k2) + (2 \times k3) + k4) \\ t_{n+1} &= t_n + dt \end{aligned} \tag{5.3}$$

As 2nd order method is less accurate then the 4th order method and the later was used for all the studies presented in this chapter.

5.3.2 MATLAB setup

Before writing an EnKF code in Python, a better understanding of how the DA method works was needed. An EnKF code written in MATLAB was obtained from the Nansen Environmental and Remote Sensing Center (NERSC) website (http://enkf.nersc.no/). The code was originally written by Geir Evensen and is maintained by Pavel Sakov. The code follows Sakov et al. (2010) for asynchronous data assimilation and contains a few test models including the Lorenz model that can be used to better understand the functioning of EnKF.

For this study, the most important files in the package are the parameter (eg. prm/prm-L3-m=2.txt) file and the main ($main.mat$) file. The parameter file can be edited for different settings which tell the package which model to use (Lorenz 65, Lorenz 96, linear advection model, etc), the ensemble size (m), and the time step (dt), etc. The main file contains code which connects all the various files in the package. These files contain instructions such as how to obtain initial conditions, finding the Kalman gain, which assimilation method to use, how to plot data, etc. These files are called/executed as and when required by $main$ depending on the settings in the parameter file.

For the Lorenz model, the user has a choice of solving the equations using Runge-Kutta 2nd order or Runge-Kutta 4th order methods. The user can also choose what type of DA method to use, such as Ensemble Kalman filter (EnKF), Ensemble Transform Kalman Filter (ETKF), deterministic EnKF, etc. For the results presented in this chapter, RK4 and EnKF were used for all studies done using the MATLAB code.

Running of the code

To run the simulation, the main file is executed from the MATLAB command window using the command $[x, x\_true, E, stats] = fmain(prmfname)$. This starts the run using a freshly generated ensemble as specified by the parameter file. Both the truth and ensemble are solved sequentially using the same time step and solver settings. Initial conditions for the ensemble and truth are obtained using a MATLAB sample file ($.mat$ format) provided by the package. This file consists of 10,000 sample positions, with three rows each row containing values for x, y, and z respectively. Using the shuffle function in MATLAB, the columns are mixed and the first column is used as the initial value for the truth/observation and the following columns as initial values for the ensemble members.

The simulation is then run which sequentially solves for the observation, followed by the ensemble at each time step. The simulations (truth and ensemble) are run from initialisation to the assimilation step. Here the ensemble is corrected using EnKF. Deterministic additive inflation is added to these corrections to obtain the new ensemble and the simulations are reinitialised. The simulation then continues from this point.

5.3.3 Python Setup

The Python code does not use the Sakov et al. (2010) method as used in the matlab version to solve for assimilation. A more stochastic version of EnKF was created here to be used for this study. This was done to better understand EnKF's behaviour under different conditions. The method used is described in chapter (4). When writing an EnKF code in Python, a lot of the ideas on how to structure the code were taken from the MATLAB package (section 5.3.2) described in the previous section.

As observational data will already be available when working with MORALS there is no need to simultaneously simulate observations and models as in the MATLAB package. This allows testing of EnKF using different time steps when simulating the ensemble to find how low resolution models can be used to successfully predict the observation. As in MORALS, where a low resolution model ensemble is used against a high resolution truth, a similar approach was decided upon when testing EnKF here.

The general structure of the EnKF Python code is divided just like the MATLAB package into a parameter file, which defines all the different settings for the model/system, and the main file, which will link all the different parts of code (solving the Lorenz model, assimilating data, reading observations, etc) and call them as and when needed. The information contained in the different sections of the code can be summarised as:

1. Parameters - gives the ensemble number, variable number, observed variable number, observation error, simulation end time, inflation, time step, steps before assimilation and file path for observation.

2. Initial condition - can either use the sample file from the MATLAB package (*L3_samples.mat*) or create our own using the random function in Python.

3. Model - uses Runge-Kutta 4th order to solve the Lorenz model equations.

4. Data assimilation - carries out the EnKF data assimilation.

5. Plotting - reads and plots the observation data and the ensemble data.

6. Main - Contains commands to call the different parts of the code to carry out the simulation.

When a study is started, the parameter file is checked by the main file and using the values present (ensemble number, time step, etc) the simulation is started. All simulations of the ensemble were run with a time step, $\Delta t = 0.01$, and the results were stored in text files to be used later when plotting data. Most of the studies were done with data being assimilated at 50 time steps, i.e. at 0.5 time intervals. This value is much larger than the one used in the MATLAB package.

Observation
As the Python setup does not simulate the observation and model simultaneously like the MATLAB package, the observational data needs to be created before data assimilation

can be done. To create observations, simulations were run at a time step, $\Delta t = 0.001$ for 200 seconds using the Runge-Kutta 4th order to solve the Lorenz model. The results were stored in a text file with each row representing a time step. Each row in the file stores the time (t) followed by values of x, y and z respectively. This same method was used for the ensemble as well. During parameter estimation values of σ, r and b are also stored in the files on the same row following the values of the normal variables.

An observational error of 2.0 was used for all three variables during DA. When obtaining the observation values for data assimilation, the code checks the first column in the data file looking for the time that corresponds to the current ensemble simulation time. This row is then read into the system to be used in the data assimilation. Each variable in the observation is perturbed using a random number generated using the observational error, between -2.0 and 2.0 in this case, as stated in the parameter file. This was the same error that was used in the MATLAB package.

5.3.4 Inflation

Inflation is essentially used for the better spread of the ensemble and to make the system more Gaussian. This leads to better sampling of the state space, which leads to a better covariance matrix, which in turn leads to a better Kalman gain and hence a better prediction. There are two types of inflation, multiplicative and additive. Chapter (4) presents more details about the use of inflation in ensemble studies.

As the MATLAB package is based on Sakov et al. (2010), it uses deterministic method for inflation. Here the covariance is used to determine the inflation for different variables. For the Python code, both deterministic inflation and stochastic inflation was coded into the system. For most of the results presented in the chapter deterministic additive inflation was used on the results before reinitialising the code.

Stochastic inflation was only used during parameter estimation as the covariance of the parameters are very small and don't change during the simulation. This narrow spread of the parameters during the simulation leads to a small covariance, which in turn leads to a small Kalman gain. The small Kalman gain leads to small deterministic inflation on the parameters. This, in turn, leads to an even smaller spread and covariance when the system comes to the next DA. This cycle continues making the Kalman gain smaller and smaller for the parameters which eventually leads to the system breaking when numbers come close to zero. Using a Gaussian random perturbation to create the new ensemble leads to a spread of values for the variables and parameters irrespective of their covariance/Kalman values. This may not be ideal but leads to a more stable system in the long run, especially for parameter estimation.

The Random module in Python was used to create the Gaussian perturbations. The standard deviation of the perturbation can be tuned but that has not been done in this study to keep the number of variables to a minimum. Hence a standard deviation of ± 1 about a mean of 0 was used to create the inflation which is the default setting of the Python normal distribution function.

5.3.5 Parameter estimation

Mathematical models are commonly used to model real world problems. However, the parameters under which these real world problems occur are never accurately known. In these cases, the DA can be used to estimate the values of these parameters. During parameter estimation the model parameters are also assumed to be variables and are observed and estimated by the data assimilation system just like any other variable of the model would be.

For the study presented in this chapter, the normal variables (x, y, z) of the Lorenz model were given random initial values just as before. But as the regime in which the system is run is known, a reasonable guess at what the initial values of the parameters are can be made. Hence the parameters were given random initial values as shown in equation (5.4).

$$\begin{aligned} \sigma &= 10 \pm 4, \\ r &= 28 \pm 5, \\ b &= 8/3 \pm 0.5. \end{aligned} \tag{5.4}$$

Using these settings, the experiments done before without parameter estimation can be repeated with an emphasis on looking at what the system tells us about the parameters. Annan and Hargreaves (2004) used the same values.

5.4 MATLAB Results

Initial studies using the MATLAB code were preformed using a time step of $dt = 0.01$ for 10000 steps giving 100 seconds of data. Studies were done for varying ensemble sizes with DA being applied at every 5 time steps. These were the default settings in the MATLAB package. As these are just initial tests no multiplicative inflation was used in the system.

Figure (5.2a) shows the first results using an ensemble of 3. The plot clearly shows that the ensemble average (blue line) does not correctly predict the observations (red line), the black dots show where the system observed the truth. The ensemble quickly starts having problems tracking the observations after the simulation starts. The simulations get stuck around the Lorenz fixed points and takes a lot of time to free themselves so they can track the observations again. Due to the small ensemble size used in this study not much of the state space is covered. Hence it is hard for the model to track the observation for a long period of time, especially when the observation moves from one fixed point to another.

Increasing the ensemble to 5 does not give much of an improvement when tracking the observations as seen in figure (5.2b). The ensemble tracks the truth correctly for slightly longer than before but quickly moves away from it when the truth moves from the neighbourhood of one fixed point to that of another.

Increasing the ensemble to 7 does make the system track the observations for a much longer period as seen in figure (5.3). But the system eventually loses track of the observations. It is clear from the figure that as the ensemble members oscillate between the neighbourhoods of the two fixed points, it becomes very hard for EnKF to correct the ensemble, especially in cases where the observations and some of the ensemble members are circling the opposite fixed points.

5.5 Python Results

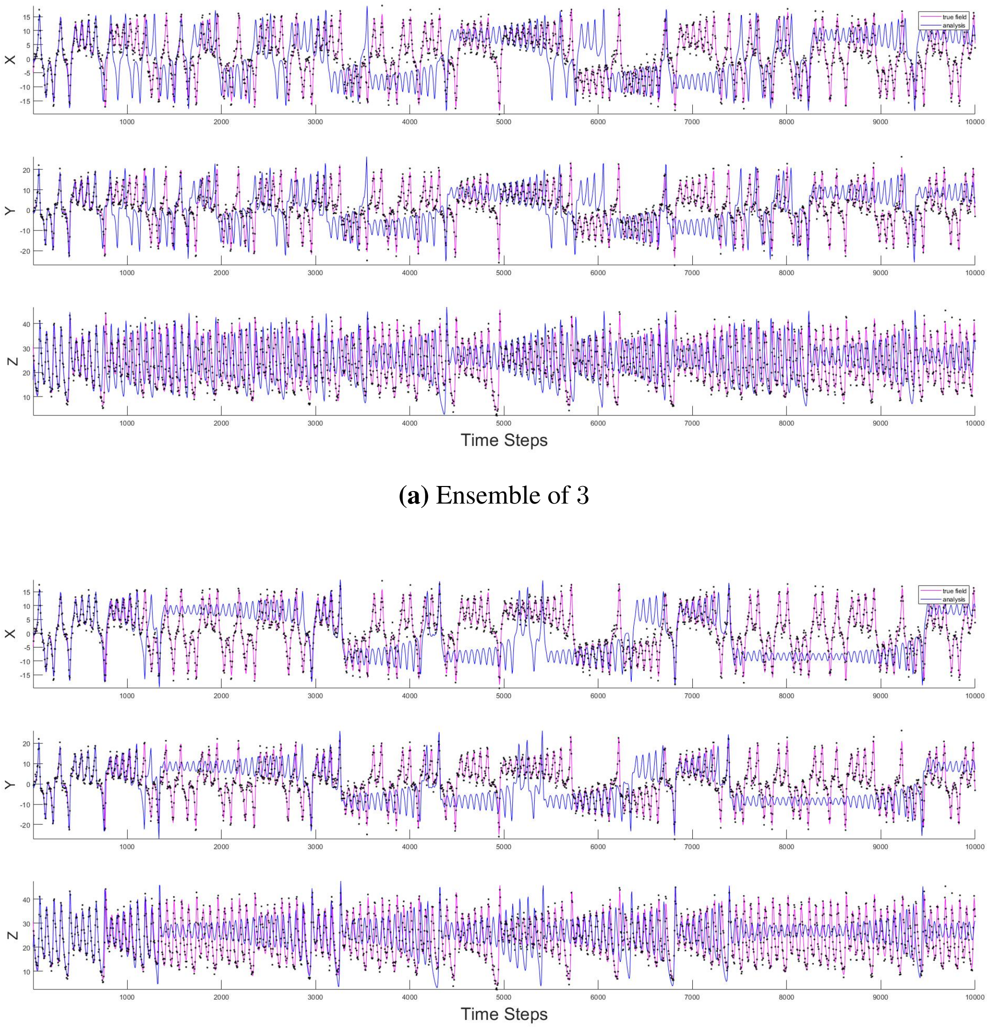
After initial tests in MATLAB to understand how EnKF works and how to code it, a more stochastic version of EnKF was created in Python. This version was used as a testbed to learn about its behaviour in certain conditions and to eventually modify the code so that it can be used with MORALS, which is a 3D model.

In this section, results are presented of tests done on EnKF using the Lorenz model. The study starts with a focus on understanding how EnKF handles ensemble sizes before looking at how inflation and data assimilation gaps affect the simulations. The was then extended into testing if EnKF can be used for parameter estimation. For most of the results presented in this chapter, the simulations were done with a time step of $dt = 0.01$ with DA at every 50 time steps, though in section (5.5.4) below the number of timesteps between DA was varied. In most cases, deterministic inflation was used for additive inflation apart from section (5.5.5) where Gaussian inflation was used.

5.5.1 Testing DA scheme

To test if the EnKF scheme is working correctly a similar test as done in section (4.6.2) was conducted. As before the aim is to make the Kalman gain is zero so no correction is done on the system. But unlike before, the values in the Lorenz model change with time. Hence to test the system the covariance matrix was manually made zero while the observation error was kept as an identity matrix during the assimilation step.

The results of the test are plotted in figure (5.4) which shows the twenty seconds of the study for an ensemble size of three where DA was done every 0.5 seconds. The red line the is truth and the blue lines are the individual ensemble. As observed, the ensemble does not converge and each member follows its own path clearly indicating that no correction is being done to move the ensemble to more closely follow the truth. The results are the same as if no DA have been done and each ensemble member evolves independently of the other.



(a) Ensemble of 3

(b) Ensemble of 5

Figure 5.2: MATLAB simulation using the original code with data assimilation at every 5 time steps and a $dt = 0.01$. The blue line is the ensemble average and the red line is the truth. The black dots are the observed truth used in DA

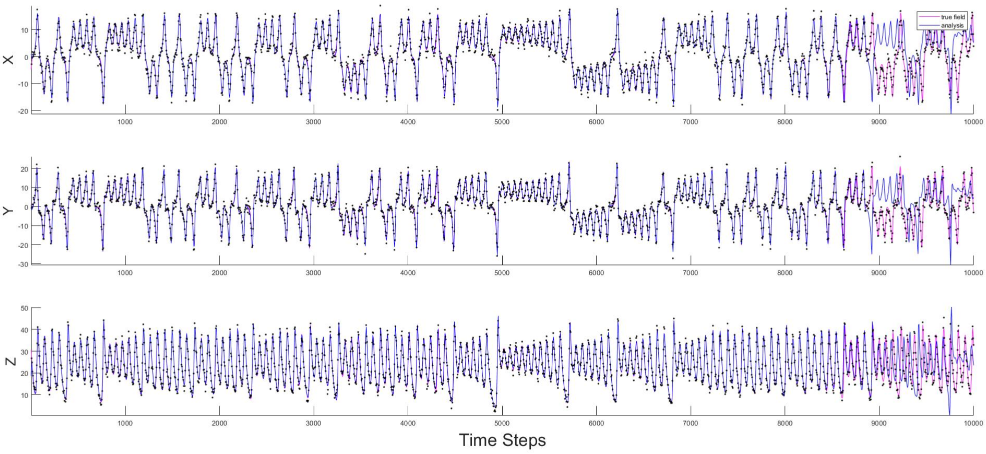


Figure 5.3: MATLAB simulation with an ensemble of 7 using the original code with data assimilation at every 5 time steps and a $dt = 0.01$. The blue line is the ensemble average and the red line is the truth. The black dots are the observed truth used in DA

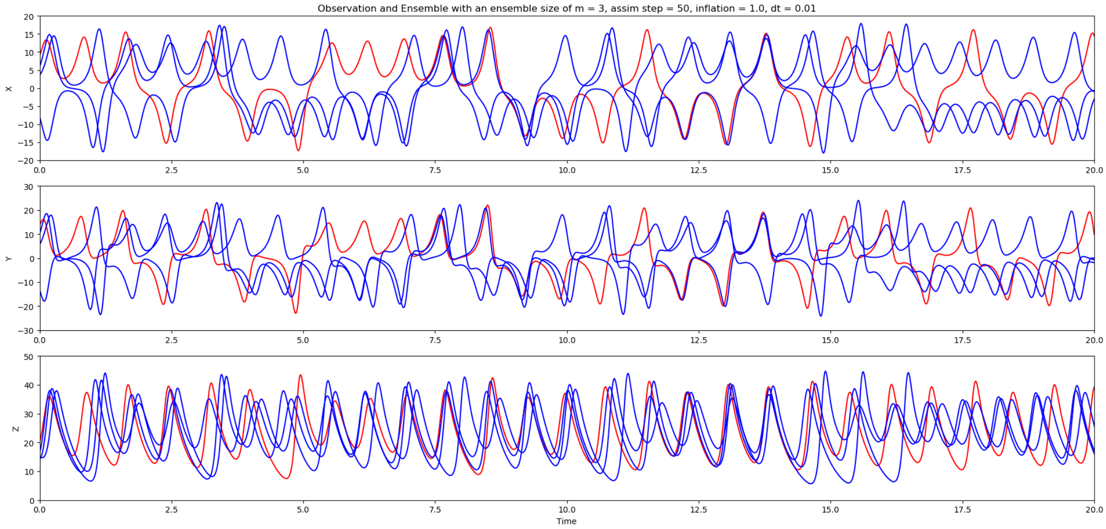


Figure 5.4: Testing DA scheme with m=3 and DA every 0.5 seconds. The red line is the truth and the blue line is the ensembles

5.5.2 Changing ensemble sizes

In the first set of studies, only the ensemble numbers are changed. Tests are done for an ensemble number of 3, 5, 7 and 10. It should be expected that as the ensemble size is increased the accuracy of the ensemble will also increase with the spread sampling. This is because with larger ensemble sizes more of the state space is being sampled leading to a more accurate covariance and Kalman gain.

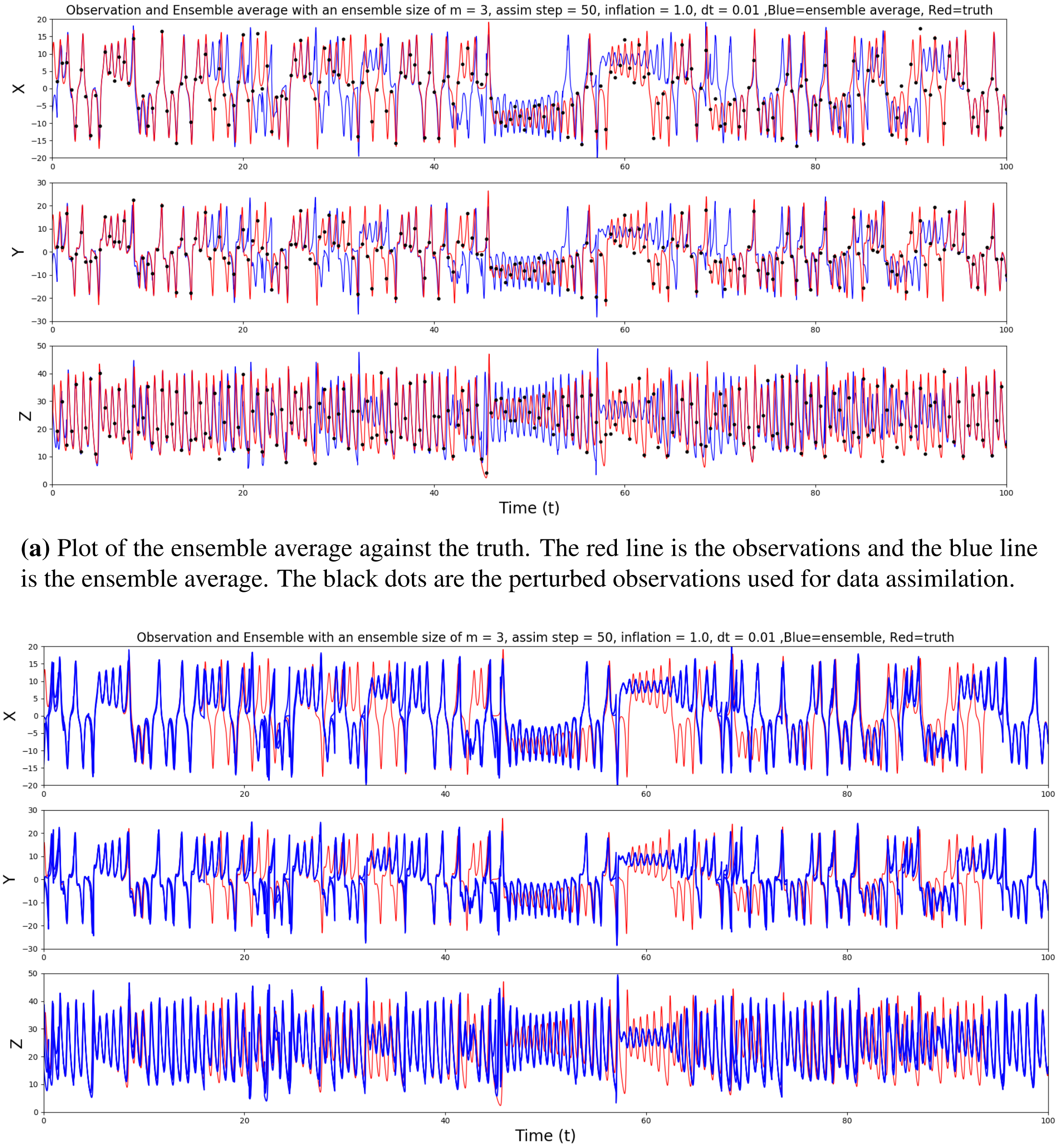
Ensemble of 3

The first study was done using a small ensemble of just 3, with the results plotted in figure (5.5). Figure (5.5a) shows the ensemble average against the truth. The system can be seen sporadically tracking the truth. It regularly tracks and then diverges from the truth. When the truth is steady, going around a single fixed point (e.g. at the $t = 50.0$ mark), the system comes close to tracking the truth but is never able to. A similar problem can be observed at the $t = 60.0$ mark where the ensemble is initially corrected too close to the fixed point and now cannot correct to the much larger position of the truth. This is due to the nature of the fixed points in the Lorenz model. Particles are trapped around the neighbourhood of a fixed point and have to orbit it until they can now escape to the other point. So here if the difference in predicted position between and ensemble and truth is large then the system will find it hard to correct the ensemble as the system is trapped in the neighbourhood. The same behaviour can be observed when the truth quickly switches between fixed points (e.g. $t = 20.0$), here the ensemble is trapped in a fixed point neighbourhood and takes longer time to correct itself.

The individual ensembles are plotted in figure (5.5b) along with the truth. The ensembles clearly stick close together and rarely diverge (they are mostly on top of each other in the plot). Due to the small ensemble, this non-divergence leads to the system not sampling enough of the state space. This makes it harder to track the observations especially when the ensembles get locked on one fixed point and cannot follow the truth to the other fixed point (e.g. $t = 20.0$). An easy way to solve this problem might be to inflate the ensemble covariance during the assimilation. Given the chaotic nature of Lorenz and the large gap between assimilation, there is no guarantee that even after corrections the simulation will continue to track the observations correctly and not diverge. The only way to solve this problem is probably to sample more of the state space with more ensembles leading to a better Kalman gain and correction.

Ensemble of 5

The ensemble size for the next study was increased to 5 with the same study being run again. The results for the simulation are plotted in figure (5.6), with the ensemble average plotted against the truth in figure (5.6a). The system does a much better job of tracking the observation compared to the smaller ensemble used before, with the average diverging less. The system does struggle around the $t = 60.0$ mark with the ensemble getting stuck around the fixed point for too long. It seems that the system trying to correct the ensemble prolongs the problem.



(a) Plot of the ensemble average against the truth. The red line is the observations and the blue line is the ensemble average. The black dots are the perturbed observations used for data assimilation.

(b) Plot of the individual ensembles against the truth. The red line is the observations and the blue lines are the ensemble members, plotted on top of each other.

Figure 5.5: Python simulation with data assimilation at every 50 time steps, a $dt = 0.01$, an ensemble of 3 members and no inflation.

Looking at the ensemble plotted against the truth in figure (5.6b) clearly shows that this struggle of the system to track the truth can again be due to a lack of spread in the state space. For the most part, the ensembles stay together and don't diverge. They also stick together around the $t = 60.0$ mark with all the ensembles being close to a fixed point. It would have been very helpful for EnKF if the ensembles would have diverged, this would have led to EnKF knowing which ensemble member was more accurate and correct the other ensembles towards that solution.

Ensemble of 7

The ensemble size is increased further to 7 and the results of the simulation are plotted in figure (5.7). The plot of ensemble average against the truth in figure (5.7a) shows that the results look better but it is still not perfect. The model does diverge at a few places but does correct itself back on track quickly. Early on in the simulation the ensemble get stuck around a fixed point and takes so long to the correct itself that the truth is back the on same fixed point neighbourhood as the ensemble.

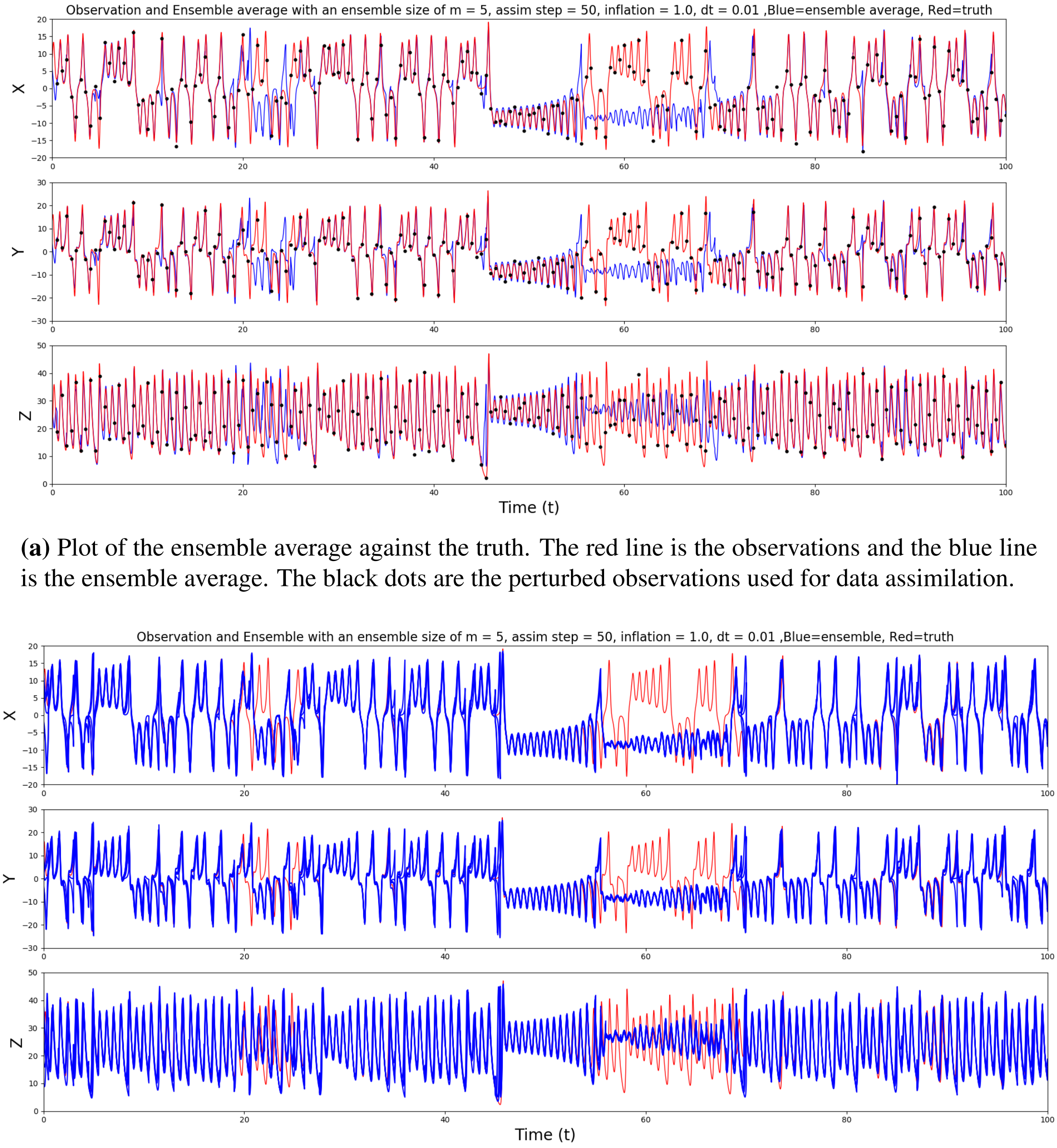
Looking at the ensembles in figure (5.6b), it can be seen that the ensembles do cover a wider state space. This is one of the main reasons why the system does better at tracking the observation. The ensembles are locked together around the same fixed point early on in the simulation this is the main reason why the system cannot correct itself to follow the truth. Later on, even when the ensembles diverge from one another the system quickly corrects the ensemble and bring them back to track the truth.

Ensemble of 10

The ensemble size was again increased to 10 and now the system gives a near perfect result with the system tracking the observations as observed in figure (5.8). The ensemble average along with the truth is plotted in figure (5.8a) with the ensemble against the truth being plotted in figure (5.8b). As observed the ensemble does a very good job of tracking the truth. The ensemble does not get stuck around a fixed point like it has done in other studies and is also able to switch between the neighbourhood of the two fixed points quite easily. The ensemble seems to diverge often especially around when the truth is switching between the two fixed points. But due to the wide sampling of the state space, it is easier for EnKF to quickly correct the system when it diverges from the truth.

5.5.2.1 Discussion

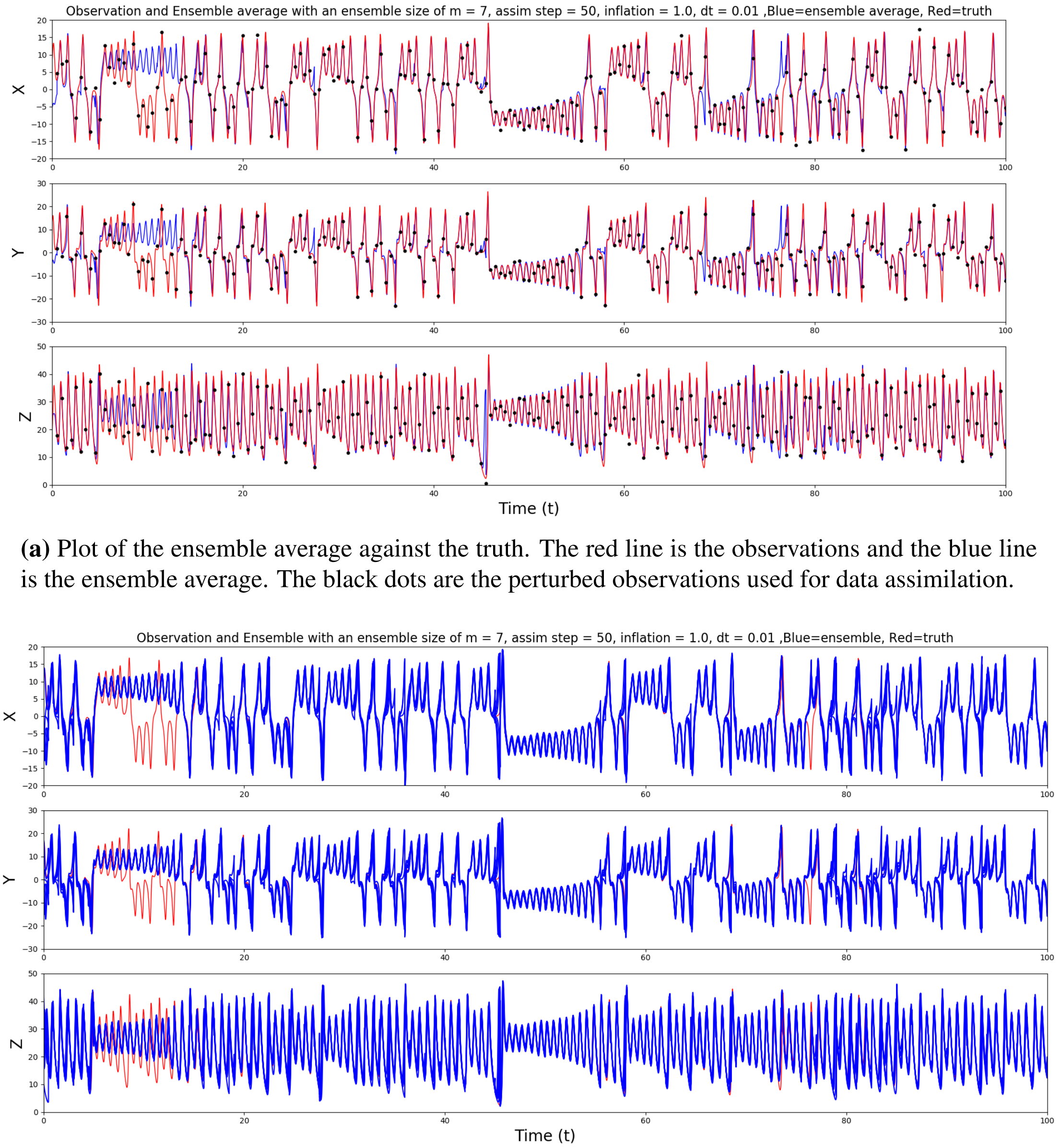
It can be seen that increasing the ensemble size can dramatically increase the accuracy of the model. In general, the transitions between the neighbourhood of the two fixed points are still very tricky to get right, especially if the observation changes from one fixed point to another, multiple times quickly. Getting the correct value when around a fixed point is also important as the ensemble might get trapped around the neighbourhood of a single fixed point for a long period of time.



(a) Plot of the ensemble average against the truth. The red line is the observations and the blue line is the ensemble average. The black dots are the perturbed observations used for data assimilation.

(b) Plot of the individual ensembles against the truth. The red line is the observations and the blue lines are the ensemble members, plotted on top of each other.

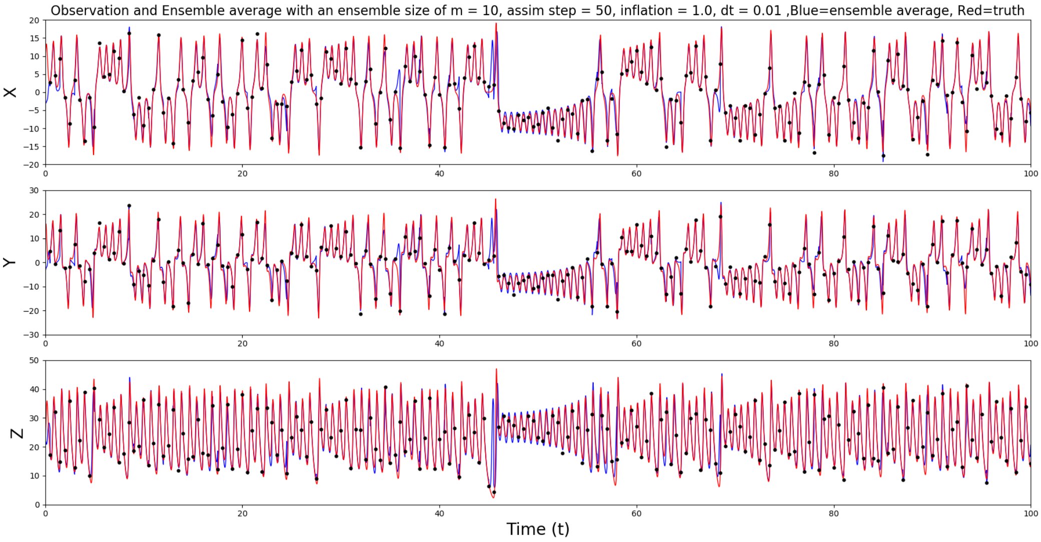
Figure 5.6: Python simulation with data assimilation at every 50 time steps, a $dt = 0.01$, an ensemble of 5 members and no inflation.



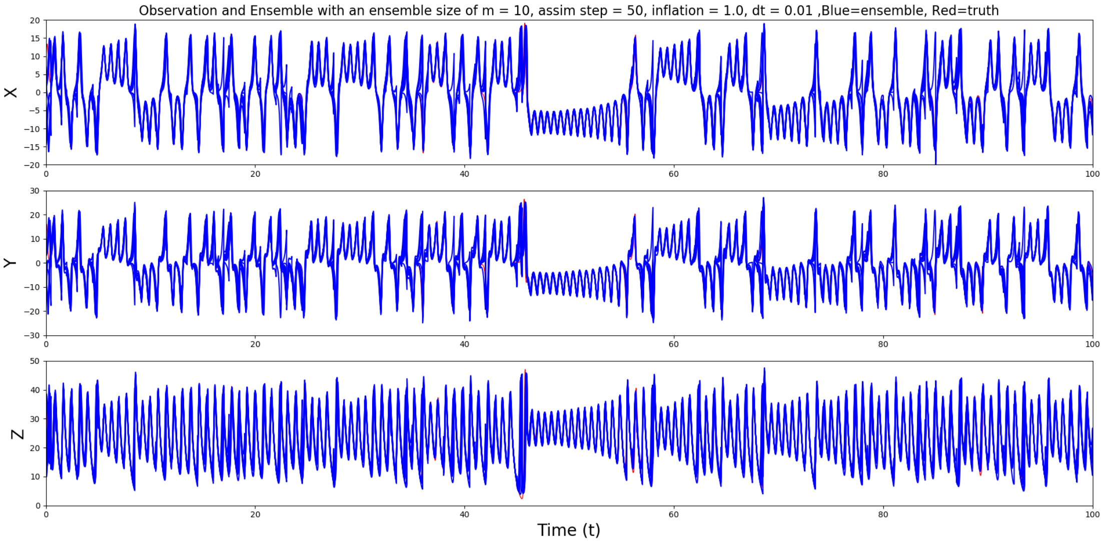
(a) Plot of the ensemble average against the truth. The red line is the observations and the blue line is the ensemble average. The black dots are the perturbed observations used for data assimilation.

(b) Plot of the individual ensembles against the truth. The red line is the observations and the blue lines are the ensemble members, plotted on top of each other.

Figure 5.7: Python simulation with data assimilation at every 50 time steps, a $dt = 0.01$, an ensemble of 7 members and no inflation.



(a) Plot of the ensemble average against the truth.The red line is the observations and the blue line is the ensemble average. The black dots are the perturbed observations used for data assimilation.



(b) Plot of the individual ensembles against the truth. The red line is the observations and the blue lines are the ensemble members, plotted on top of each other.

Figure 5.8: Python simulation with data assimilation at every 50 time steps, a $dt = 0.01$, an ensemble of 10 members and no inflation.

Due to the chaotic nature of the Lorenz system, the ensembles will always diverge from the observations regardless of the ensemble size. But the likelihood of the average of the ensemble diverging does depend on the ensemble size with smaller ensembles having a higher likelihood of diverging compared larger ones. The likelihood of the system correcting itself after diverging also increases with ensemble size.

When the Python results are compared to the MATLAB results it can be seen that they show very similar behaviour when increasing the ensemble size. Even with just the use of Gaussian perturbation, the system does a good job of tracking the observations even with this larger gap between DA.

5.5.3 Inflation

As seen in the previous experiment, the ensemble size affects the accuracy of EnKF. In this section, results are presented for the different ensembles sizes used in the previous section but with multiplicative inflation. As larger ensemble sizes lead to better accuracy a smaller inflation should be required for these studies and vice versa for smaller ensemble studies.

Ensemble of 3

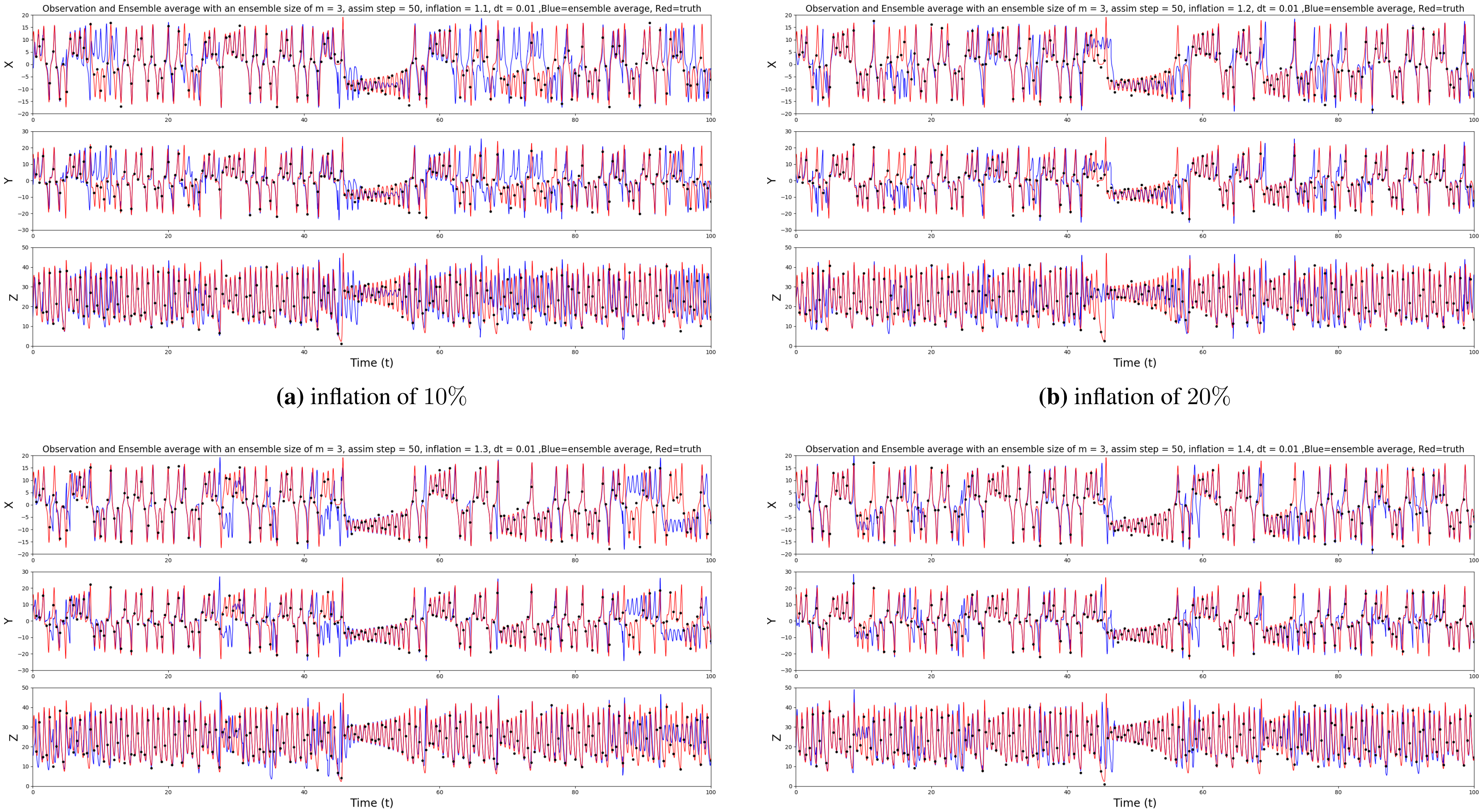
The study starts with an ensemble of 3 being used with an multiplicative inflation of 10%. The results for this are plotted in figure (5.9a), and show a slight improvement when compared to figure (5.5) where no inflation was done. There is less divergence between the ensemble and the truth. But problems persist with the system finding it tricky when the truth regularly moving between the two fixed points later on in the simulation.

The inflation was increased to 20% with the results plotted in figure (5.9b). The results are an improvement on the smaller inflation, with the ensemble being able to track more of the observations. Even when the ensemble diverges due to the inflation the system can correct itself and get close to the truth. This can be seen at $t = 50.0$ mark, where the ensemble successfully switches to the correct fixed point and then tracks the truth reasonably closely for some time.

The inflation was increased further to 30% and the results plotted in figure (5.9c), with another improvement being observed to the tracking of the truth. Even when the ensemble diverges it can correct itself and start tracking the truth. Increasing the inflation even further to 40% with the results are plotted in figure (5.9d). Another small improvement is observed here when it comes to tracking the truth. The results now seem comparable to using larger ensemble sizes.

The inflation is increased again to 50% with the results plotted in figure (5.10a). Surprisingly more improvement is observed in tracking the truth. And just like what was observed at higher ensemble studies before when the system does diverge from the truth it quickly corrects itself and can track the truth again after a few assimilation steps.

The inflation is increased to 60% with the results plotted in figure (5.10b). The



(a) inflation of 10%

(b) inflation of 20%

(c) inflation of 30%

(d) inflation of 40%

Figure 5.9: Python simulation with data assimilation at every 50 time steps, a $dt = 0.01$, an ensemble of 3 and differing inflation. The red line is the observations and the blue line is the ensemble average. The black dots are perturbed observations used for data assimilation.

results are again very good, with the ensemble almost tracking the truth through the simulation. The results are comparable to results from an ensemble of 7.

Ensemble of 5

As described previously, with the increase of the ensemble size to 5 a lot of improvements to the tracking were observed. Hence the amount of inflation required to get accurate tracking of the truth should be less. The study was started with multiplicative inflation of 10% with the results plotted in figure (5.11a). The results improve upon what was observed without any inflation, especially around the $t = 60$ mark, which caused problems previously. The system follows the truth quite well throughout the simulation with only a few deviations which are quickly corrected during DA. The only major deviation from the truth is observed around the $t = 80$ mark with the truth switching between the neighbourhood of the two fixed points while the ensemble is stuck around just one fixed point.

The multiplicative inflation was increased to 20% with the results plotted in figure (5.11b). The results seem on par with the lower inflation, with deviation at the $t = 70$ mark where the system switches between the neighbourhood of the two fixed points. Increasing the inflation further to 30%, the results were plotted in figure (5.11c). Improvement is observed here with the ensemble tracking the truth nearly perfectly with little divergence.

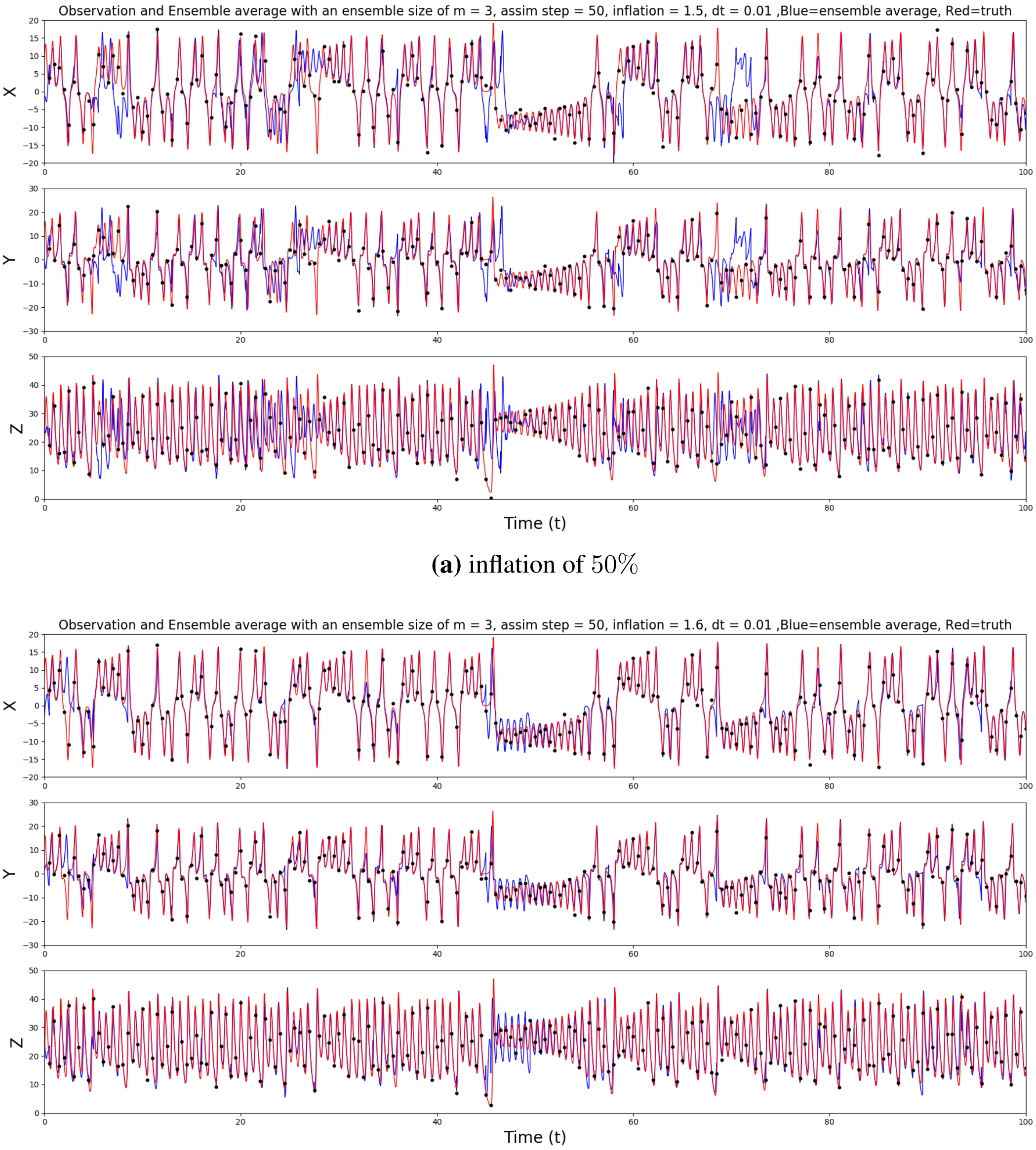
Finally, the inflation was increased to 40% with the results plotted in figure (5.11d). Now the accuracy of the system seems to decrease with more deviation being observed at various places which seem to take longer for the system to correct. This includes an areas around $t = 30$ mark where the ensemble is stuck near a fixed point. This seems to indicate that there might be a limit to the inflation value that can be used in the system before the system starts to spread/diverge too much.

Ensemble of 7

Increasing the ensemble size further to 7, the amount of inflation needed to accurately track the observation should be even lower than before. In the first test, inflation of 5% was used with the results being plotted in figure (5.12a). With this small inflation, a small improvement is observed when compared to no inflation as seen in figure (5.7). The ensemble system tracks the truth well through the simulation with a few places where it deviates, just as it did in the normal simulation with no inflation.

Increasing the inflation to 10%, the results are plotted in figure (5.12b). The system is on par with what was observed before, tracking most of the simulation, but it still deviates at places. The ensemble has issues around the $t = 60$ mark but quickly corrects itself. This quick correction is observed every time the system deviates from the observation.

The inflation was increased to 15% with the results plotted in figure (5.12c). The improvement are again just marginal when compared to the previous results. The system does well around the $t = 60$ mark and does not deviate much throughout the run with any deviations quickly corrected.



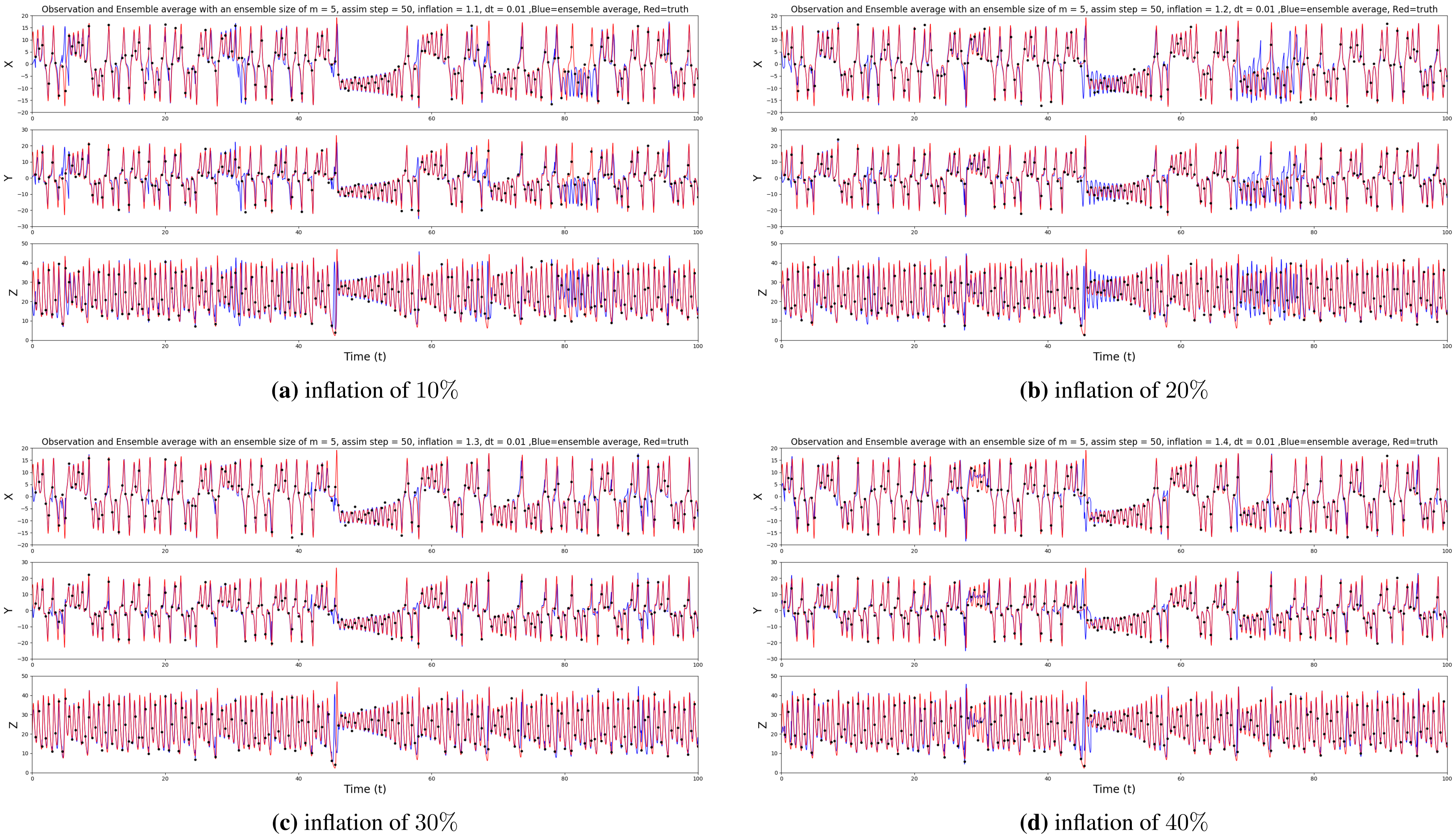
Observation and Ensemble average with an ensemble size of m = 3, assim step = 50, inflation = 1.5, dt = 0.01 ,Blue=ensemble average, Red=truth

(a) inflation of 50%

Observation and Ensemble average with an ensemble size of m = 3, assim step = 50, inflation = 1.6, dt = 0.01 ,Blue=ensemble average, Red=truth

(b) inflation of 60%

Figure 5.10: Python simulation with data assimilation at every 50 time steps, a $dt = 0.01$, an ensemble of 3 and differing inflation. The red line is the observations and the blue line is the ensemble average. The black dots are perturbed observations used for data assimilation.



(a) inflation of 10%

(b) inflation of 20%

(c) inflation of 30%

(d) inflation of 40%

Figure 5.11: Python simulation with data assimilation at every 50 time steps, a $dt = 0.01$, an ensemble of 5 and differing inflation. The red line is the observations and the blue line is the ensemble average. The black dots are perturbed observations used for data assimilation.

Increasing the inflation further to 20%, the results were plotted in figure (5.12d). The simulation here gives similar results to before, with no improvements being observed. This seems to indicate that inflation is no longer effective at correcting the ensemble. As before the ensemble tracks the truth well with a few deviations from which the system corrects quickly.

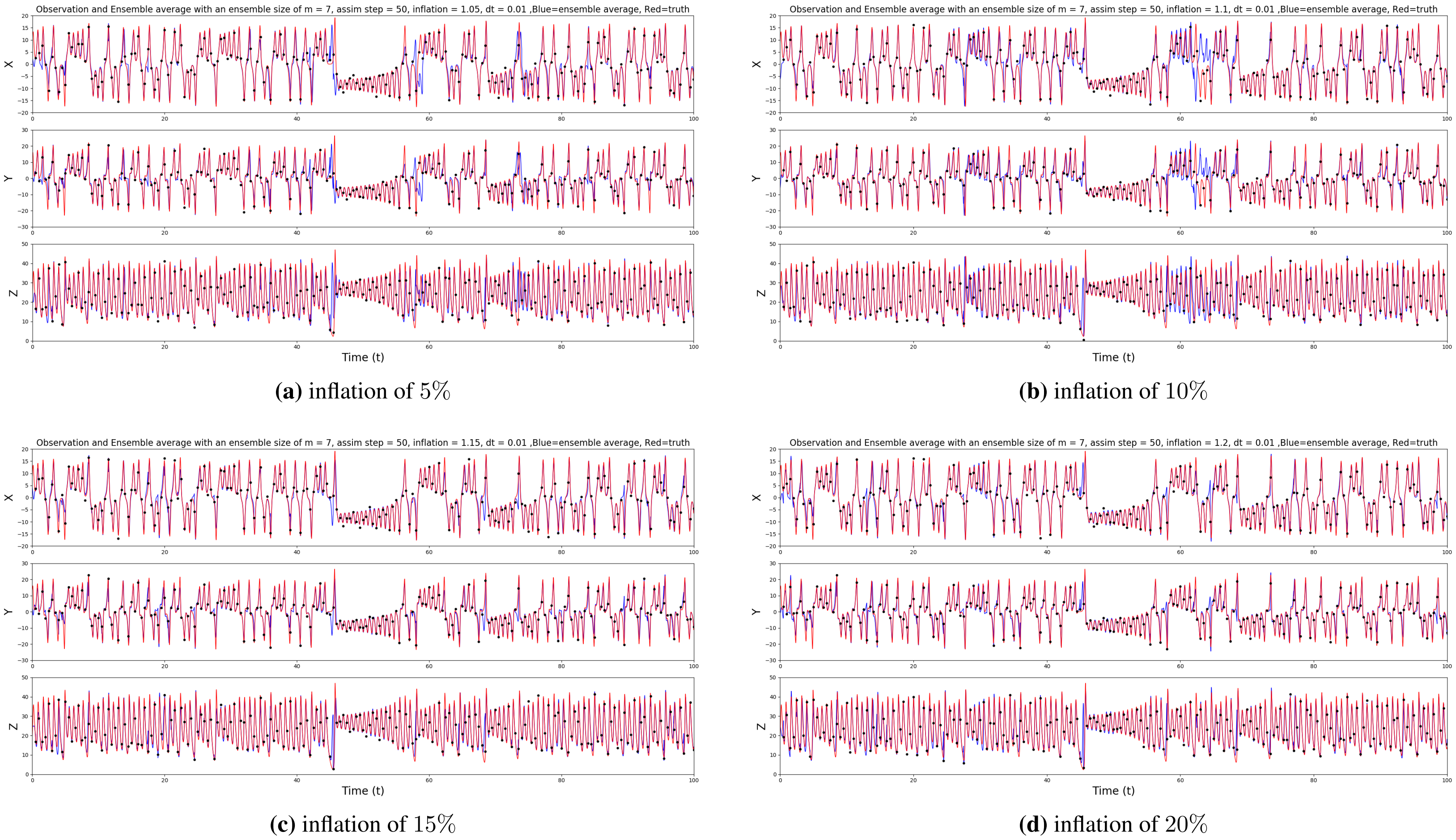
5.5.3.1 Discussion

Inflation greatly helps in spreading the ensembles around the state space helping to better track the observation. Generally the smaller the ensemble the higher the inflation needed for the system to track the observations accurately. This makes sense as inflation just spreads the ensemble in state space leading to a better chance of covering the observations even with low ensemble numbers. As bigger ensembles should be well spread in state space they don't need large inflation. Even with the correct inflation, the system might show some places of deviation from the truth due to the chaotic nature of the Lorenz model. But the real test is how quickly the system can correct itself and get back to tracking the truth.

The usefulness of inflation can be observed when using small ensembles like an ensemble of 3. As the inflation in the system is increased, the system becomes better at tracking the observations. At areas where previously the system provided poor prediction, the tracking of the observation is better. As discussed before, as the size of the ensemble increases, the size of the state space sampled also increases, hence with an ensemble of 5 and 7 a smaller inflation of around 30% and 15% respectively does a better job of tracking the observations. Over inflating the ensembles seems to not lead to any improvements in the accuracy of the system and in some cases can lead the system to be more inaccurate as observed at the inflation of 40% with an ensemble of 5. This is due to the Kalman gain giving higher than needed weights to the ensembles. This leads to an overcorrection which makes the ensemble members diverge. One way to solve this problem in a higher dimensional system would be to use localisation to stop false correlation, which might lead to unnecessary overcorrection of the system.

5.5.4 Changing Assimilation Gap

As a good understanding has been developed on how EnKF copes with ensemble sizes and inflation, in this section the time between assimilations was changed to see how the system copes with longer assimilation intervals. When trying to track a chaotic model such as the Lorenz system, given enough time between assimilations the individual ensemble members will always diverge. A way to overcome this problem would be to use a larger ensemble, but this will increase the computational resource required. Another way to tackle this is to tune the multiplicative inflation. As this is just a simple factor which is used during DA to increase the covariance and is computationally much cheaper.



(a) inflation of 5%

(b) inflation of 10%

(c) inflation of 15%

(d) inflation of 20%

Figure 5.12: Python simulation with data assimilation at every 50 time steps, a $dt = 0.01$, an ensemble of 7 and differing inflation. The red line is the observations and the blue line is the ensemble average. The black dots are perturbed observations used for data assimilation.

An ensemble of 7 was used for all the studies presented in this section, as previously observed produced accurate predictions. It is logical to assume that a higher inflation will be needed in all cases to get the same accuracy that was observed when DA was done every 50 time steps.

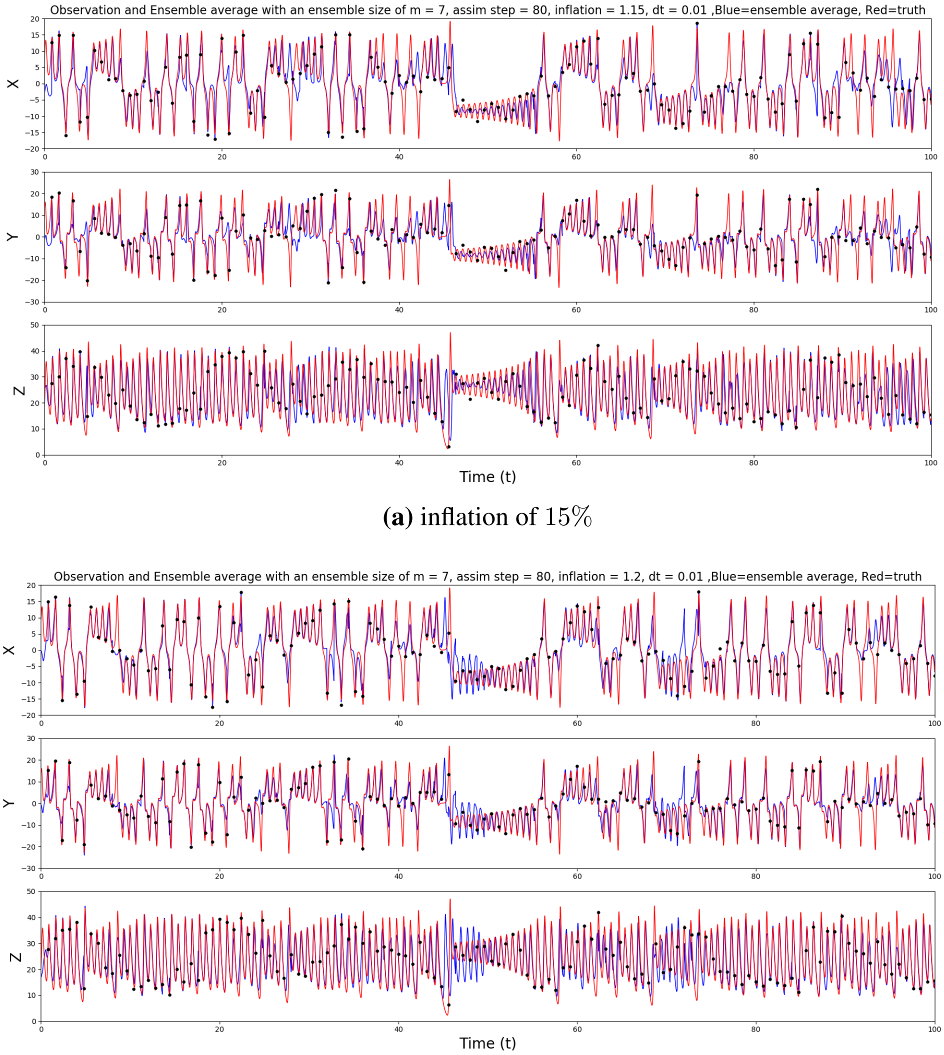
60 **time steps**
For the first series of tests, DA is done every 60 time steps. With no multiplicative inflation, the results are plotted in figure (5.15a). As expected with a larger gap between assimilations the system now diverges more then it did previously with a shorter gap as observed in figure (5.7) and (5.12). For the most part, the ensemble does a good job of tracking the truth. The only major point during the simulation where the ensemble diverges is early on in the simulation where the ensemble seem to be stuck in the neighbourhood of the single fixed point, around the $t = 10$ mark.

In the second test multiplicative inflation 15% was used in the study. Figure (5.15b) shows the results of the study. As expected, with the inflation the system tracks the truth more accurately, especially in the first half of the simulation when compared to the previous study when no multiplicative inflation was used. The system only diverges towards the end of the study around the $t = 80$ mark and does not fully recover until nearly the end of the simulation.

80 **time steps**
For the second series of tests, DA is done every 80 time steps. Using inflation of 15%, the results are plotted in figure (5.13a). With the long gap and with multiplicative inflation the results here seem good when it comes to tracking the truth. But this is a bit deceptive, with the longer interval between assimilation the system underestimates the truth a lot. The problem might be due to the longer assimilation interval, as this gives more time for deviation for the individual ensemble which leads to a lower average. Other points where the ensemble does diverge it is quickly corrected to come back to the correct path.

Increasing the inflation to 20%, the results are plotted in figure (5.13b). There is no visible improvement in the tracking of the truth with this higher inflation. The ensemble average deviates less and tracks the truth far better. The ensemble at $t = 50$ is in the same neighbourhood as the truth but due to the characteristics of the fixed point, is never able to track the truth accurately. The inflation is increased further to 25%, and the results are plotted in figure (5.14). With this higher inflation, the ensemble does very well now to accurately track the truth and only deviates a few times. When deviations do happen the system can quickly correct the ensemble and track the truth again.

100 **time steps**
For the next set of studies, the interval between DA was increased to 100 time steps, i.e. 1 time unit. With this longer interval between successive DA, larger inflation will be required to accurately track the truth. As before an ensemble of 7 was used and for the first test, multiplicative inflation of 20% is applied. The results of this study are plotted in figure (5.16a) and shows very close results, with the ensemble tracking the truth. One



(a) inflation of 15%

(b) inflation of 20%

Figure 5.13: Python simulation with data assimilation at every 80 time steps, a $dt = 0.01$, an ensemble of 7 and differing inflation. The red line is the observations and the blue line is the ensemble average. The black dots are perturbed observations used for data assimilation.

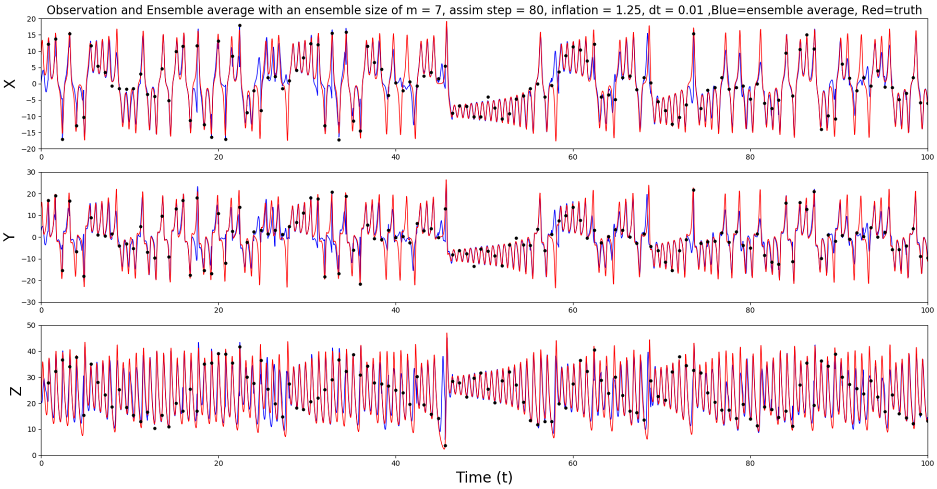


Figure 5.14: Python simulation with an inflation of 25% data assimilation at every 80 time steps, a $dt = 0.01$, an ensemble of 7 and differing inflation. The red line is the observations and the blue line is the ensemble average. The black dots are perturbed observations used for data assimilation.

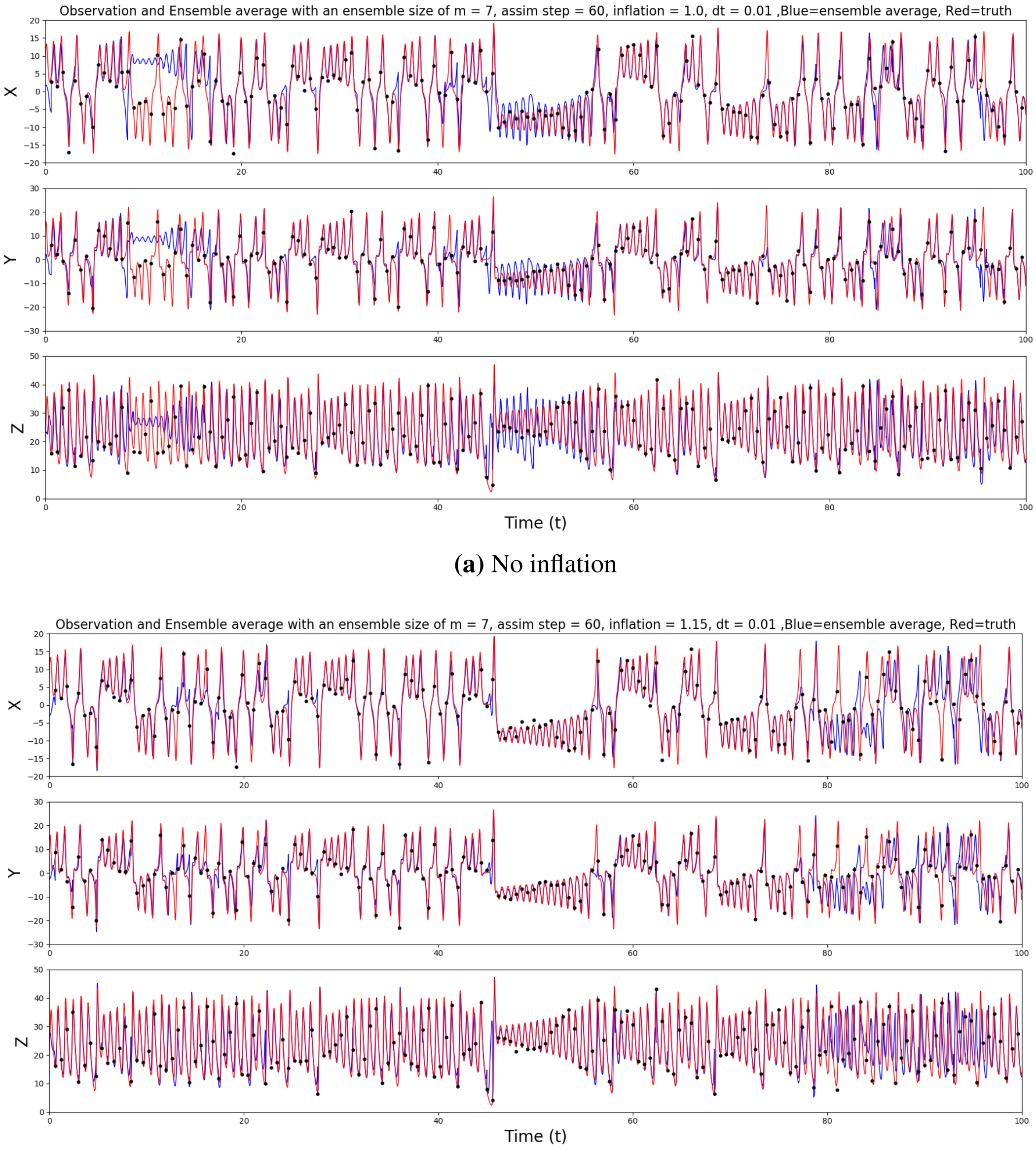
thing to observe is that the estimations in many cases are lower than the truth. This is just due to the divergence of the ensemble system due to the gap.

For the next set of studies, the inflation was increased to 25% and 30% with the results being plotted in figure (5.16b) and (5.17) respectively. In both cases, no significant changes are observed when compared to lower inflation of 20%. This seems to indicate that the long assimilation gap does limit to how accurate the system can be even with the addition of multiplicative inflation. This is because with the long gap ensembles will diverge. This will lead to a sampling of bigger state space which should lead to a better correction by EnKF but with a long gap, there is a high chance of the ensemble diverging before it can be corrected.

Ensemble of 14

With the large gap between assimilation, one way the system can now track the truth is to use a larger ensemble size. With this even when the individual ensembles diverge from the truth on average the results should still be able to track the truth. To test this idea the ensemble size is increased to 14, figure (5.18a) shows the results when no multiplicative inflation is used. The results here are on par what is observed with the lower ensemble of 7 with higher multiplicative inflation. There are places where the system diverges from the truth but is quickly corrected brought back.

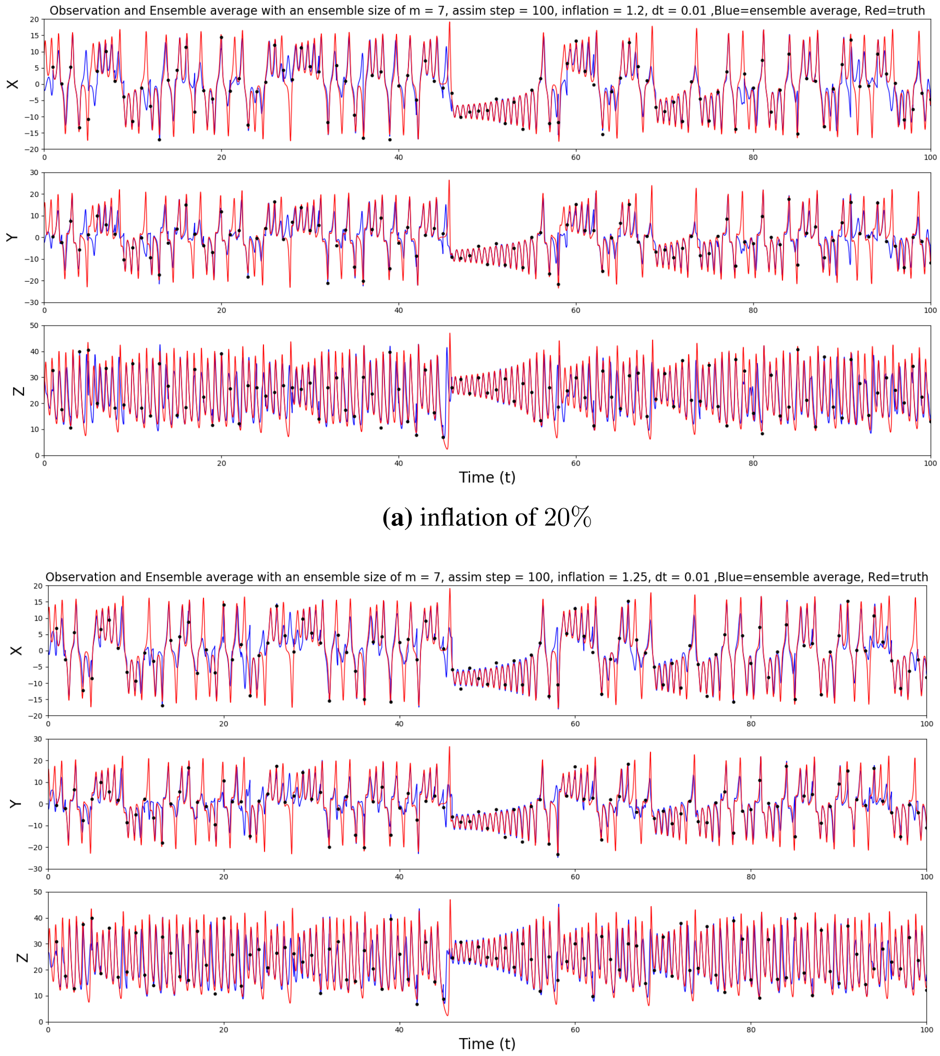
Using inflation of 10%, the results plotted in figure (5.18b). The results are an improvement when compared to when no multiplicative inflation was used. There are fewer deviations from the truth but the values are still underestimated. Two more studies are done with a multiplicative inflation of 15% and 20% with the results plotted in figure (5.18c) and (5.18d) respectively. In both cases, only marginal improvements are



(a) No inflation

(b) inflation of 15%

Figure 5.15: Python simulation with data assimilation at every 60 time steps, a $dt = 0.01$, an ensemble of 7 and differing inflation. The red line is the observations and the blue line is the ensemble average. The black dots are perturbed observations used for data assimilation.



(a) inflation of 20%

(b) inflation of 25%

Figure 5.16: Python simulation with data assimilation at every 100 time steps, a $dt = 0.01$, an ensemble of 7 and differing inflation. The red line is the observations and the blue line is the ensemble average. The black dots are perturbed observations used for data assimilation.

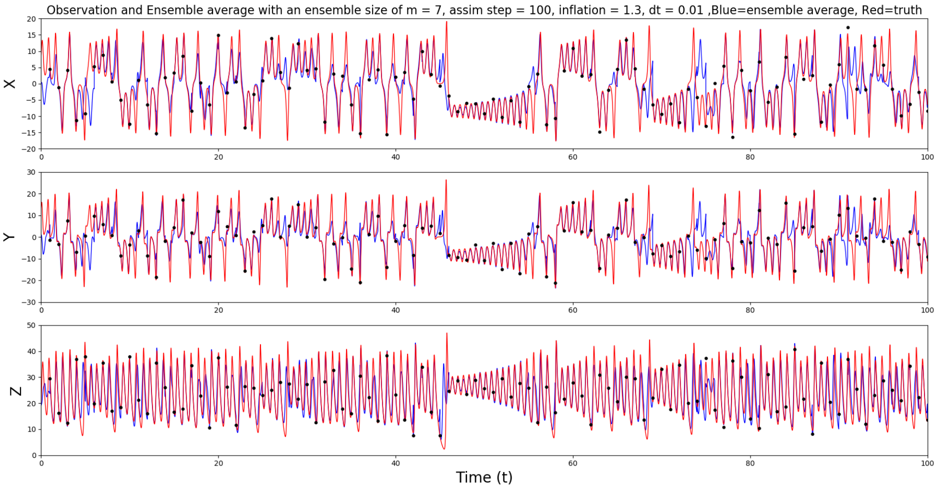


Figure 5.17: Python simulation with an inflation of 30%, data assimilation at every 100 time steps, a $dt = 0.01$, an ensemble of 7 and differing inflation. The red line is the observations and the blue line is the ensemble average. The black dots are perturbed observations used for data assimilation.

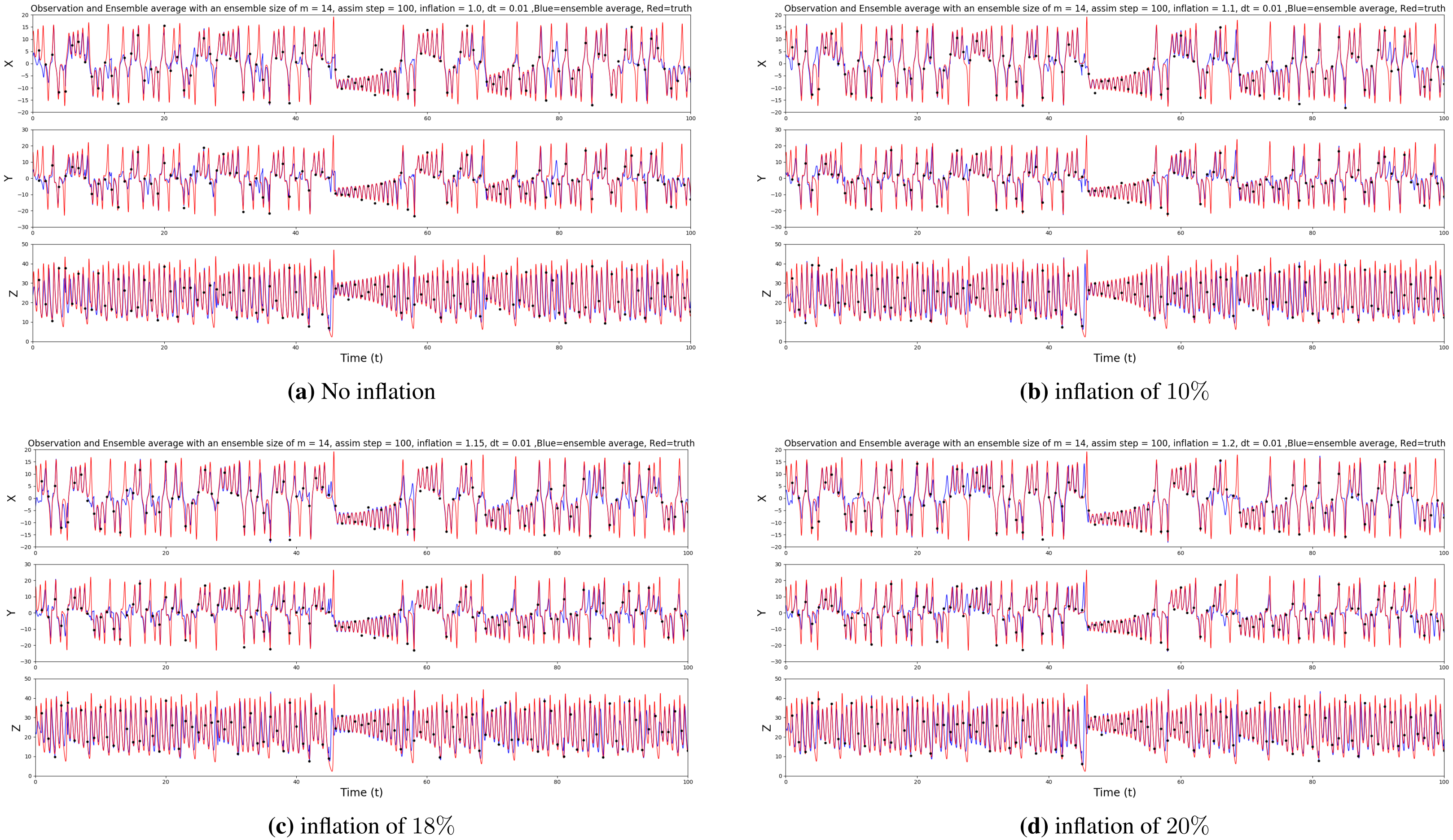
observed in the tracking of the truth. The system still deviates from the truth at moments and the values are still underestimated at places. All this is due to the ensemble diverging between one data assimilation step and the next. Even tho this larger ensemble of 14 does improve upon the results when only an ensemble of 7 was used, an even larger ensemble might be needed to get perfect tracking of the truth. This large gap between assimilation steps just provides too much time for the ensemble to diverge.

5.5.4.1 Discussion

As theorised, with the larger interval between DA the system finds it harder to track the truth. This is because the ensemble members have a higher probability of diverging. But in most cases, multiplicative inflation can help with limiting the deviation from the truth. The longer interval the higher the amount of inflation needed to accurately track the truth.

With a time interval of 60 time steps, the ensemble of 7 can still track the truth accurately as this is not much of a difference from the normal system of 50 time steps. Smaller multiplicative inflation of 15% does a good job giving good results here as seen in figure (5.15). Increasing the gap to 80 time steps leads to needing a higher inflation of around 25% to get accurate results, even then the system diverges a few times from the truth as seen figure (5.13), but is quickly corrected.

Increasing the DA interval further to 100 time steps causes the system to start underestimating the values of the ensemble. Even with an inflation of 30% the system does not fully track the truth as seen in figure (5.16). This seems to indicate that the long assimilation gap does limit to how accurate the system can be even with the addition of



(a) No inflation

(b) inflation of 10%

(c) inflation of 18%

(d) inflation of 20%

Figure 5.18: Python simulation with data assimilation at every 100 time steps, a $dt = 0.01$, an ensemble of 14 and differing inflation. The red line is the observations and the blue line is the ensemble average. The black dots are perturbed observations used for data assimilation.

multiplicative inflation. This is because with the long gap ensembles will diverge. This will lead to a sampling of bigger state space which should lead to a better correction by EnKF but with a long gap, there is a high chance of the ensemble diverging before it can be corrected.

To overcome this issue the ensemble size was increased to 14. This does give better results when compared to an ensemble of 7 but the results are still not perfect as observed in figure (5.18). Even with an inflation of 20%, there was no significant improvement in the accuracy of the system. An even larger ensemble might be needed to get perfect tracking of the truth. This large gap between assimilation steps just provides too much time for the ensemble to diverge.

It is clear that for a system like the Lorenz model there is a range of time intervals between DA where EnKF does well in tracking the truth. With larger time intervals, inflation can be used to reduce the deviation of the ensemble from the truth, but this still does not lead to a perfect result. Given the chaotic nature of the Lorenz model, there seems to be a limit on the time between DA, after which even increasing the ensemble number does not lead to perfect results. This was demonstrated with an ensemble of 14 where, even with inflation, the system did not yield perfect results.

5.5.5 Parameter estimation Results

In all previous studies, parameter values of $\sigma = 10$, $r = 28$ and $b = 8/3$ were used for both the truth and the ensemble. This gave the system the best chance to track the truth as both cases used the same parameters. But in the real world case (e.g. in experiments) there will be an amount of uncertainty about the exact values for the parameters. In this case, DA can be used to estimate the values of these parameters for the system. This is called parameter estimation.

In this set of studies, the parameters for the ensemble are now part of the variables that are predicted by EnKF. At the beginning of the study the ensemble randomly chooses the value for the parameter in each ensemble member using $\sigma = 10 \pm 4.0$, $r = 28 \pm 5.0$ and $b = 8/3 \pm 0.5$. In all studies presented, DA was done every 50 time steps with Gaussian additive inflation used in all studies as discussed in section (5.3.4). The parameter values for the truth are kept the same as the original, $\sigma = 10$, $r = 28$ and $b = 8/3$. Results are presented for an ensemble size of 7, 10, 15 and 20. Table (5.1) provides a summary of the results presented in this section.

No Inflation

In the first study using an ensemble 7 and no inflation, a study was done on the system and the results are plotted in figure (5.19). With the parameters being changed every DA step the system can remain in the same neighbourhood as the truth, even though the values of the ensemble average don't always seem correct. The ensemble collapses to around zero for the x and y variables at the $t = 50$ mark. This is due to the combination of parameters the system has predicted for the system. In the DA steps that follow,

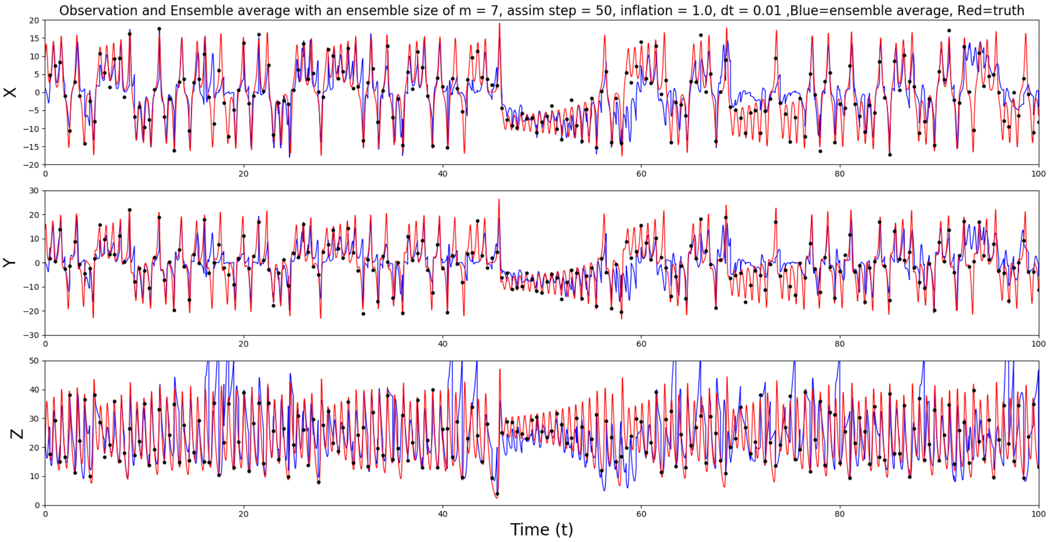


Figure 5.19: Python simulation with parameter estimation active, an ensemble of 7, data assimilation at every 50 time steps, a $dt = 0.01$ and no inflation. The red line is the observations and the blue line is the ensemble average. The black dots are perturbed observations used for data assimilation.

the system corrects the ensemble and follows the truth reasonably well until the system collapses again near the end of the simulation.

The estimation for the different parameters predicted by EnKF is plotted in figure (5.21). The predictions of the different parameters as the system evolves is plotted in figure (5.21a). Looking at the σ graph, the values predicted diverges a lot but surprisingly the system favours predicting values larger than the actual values (i.e. over 10), than predicting small values. The r values also vary quite a lot and the pattern of prediction here is the opposite of what was observed before with a majority of the predicted values being smaller than the actual values, and only a small set being higher than the actual values. For b the values predicted seem to be quite close to the actual value used for the truth.

The rest of the plots in figure (5.21) show a histogram for the parameters. Using this, a mean and standard deviation for the parameters can be calculated to be compared to the truth. Figure (5.21b) plots the results for σ. There is a large distribution of predicted values by EnKF as mentioned in the previous paragraph. The mean predicted by the system is $\sigma = 13.27 \pm 6.856$, while the true value used was $\sigma_T = 10$. So the system overpredicts the actual value of the parameter. The large variation of predicted values leads to a large standard deviation. With this, the true value does lie within the standard deviation bars predicted by the ensemble.

Figure (5.21c) shows the histogram for r, here similar distributional behaviour is observed as with σ. The mean predicted by the system is $r = 24.931 \pm 2.9$, while the value used for the truth was $r_T = 28.0$. As observed, the distribution of the prediction is large as reflected by the standard deviation. In this case, the system underpredicts the

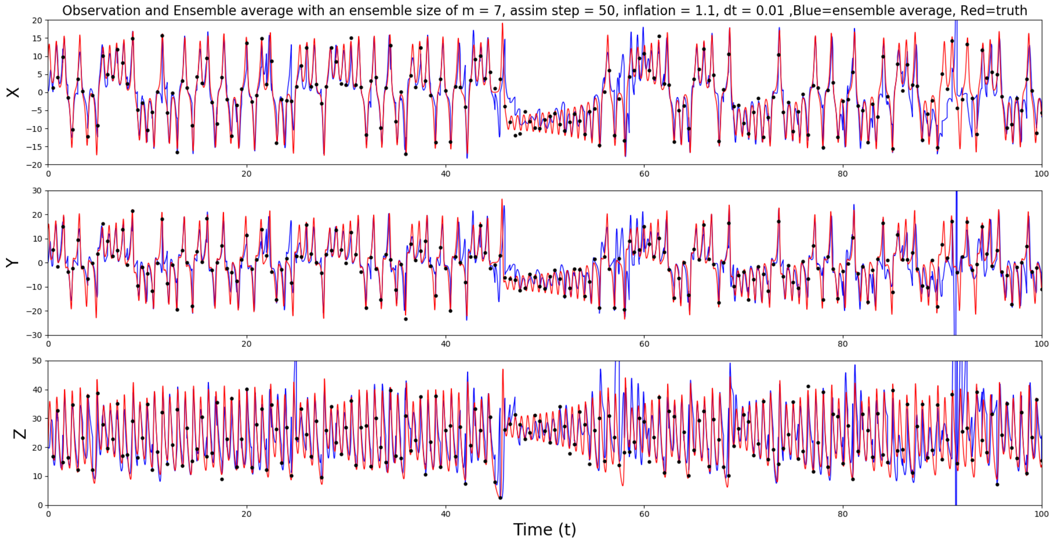


Figure 5.20: Python simulation with parameter estimation active, an ensemble of 7, data assimilation at every 50 time steps, a $dt = 0.01$ and 10% inflation. The red line is the observations and the blue line is the ensemble average. The black dots are perturbed observation used for data assimilation.

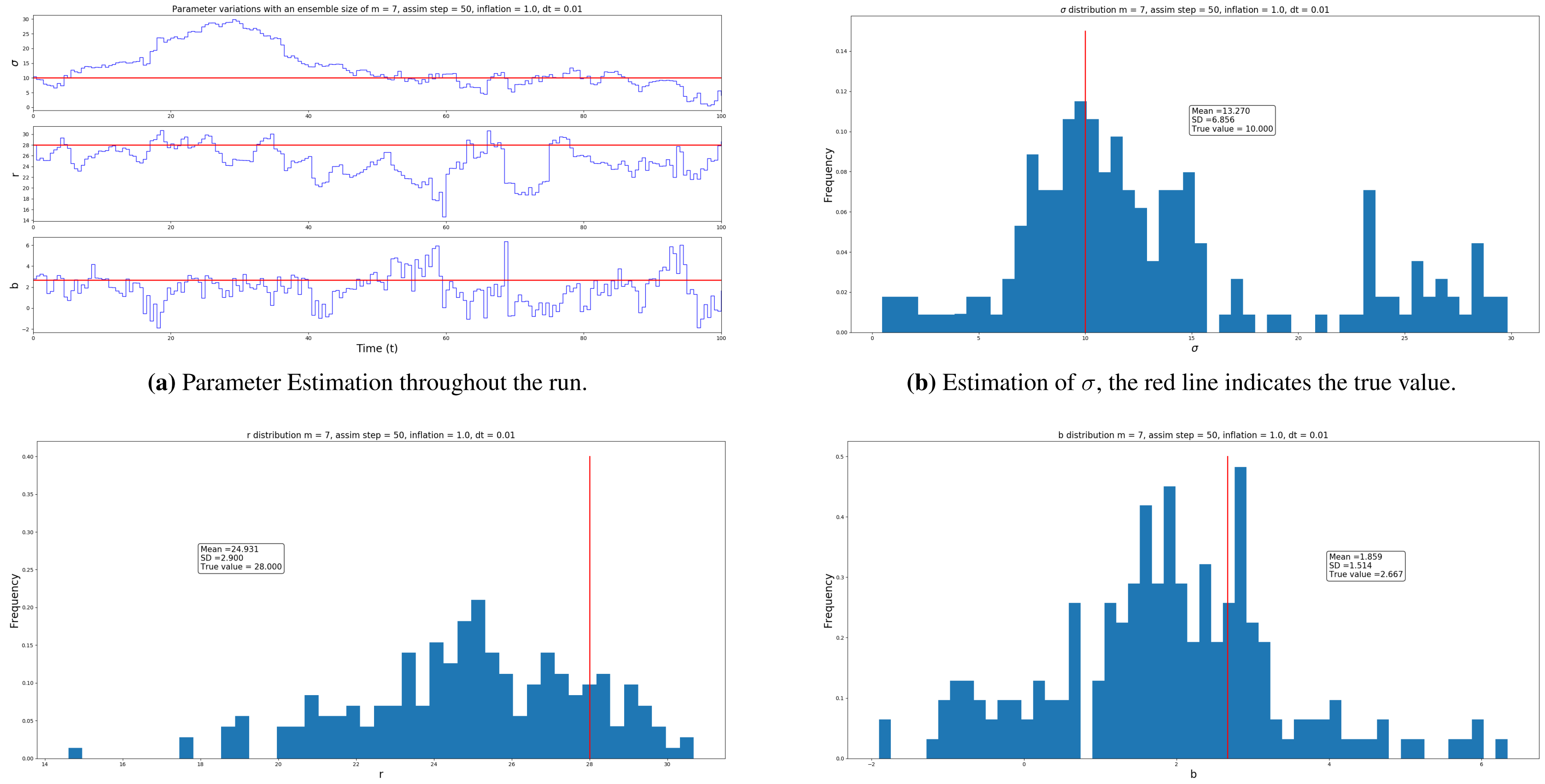
values of the parameter. The standard deviation means that the true value of r just about lie within the standard deviation bars of the estimate for r.

Lastly figure (5.21d) shows the histogram for b. The mean predicted by the system is $b = 1.859 \pm 1.514$, while the true value used was $b_T = 2.667$. As observed here the estimates are smaller than the true value. The system underpredicts the value for b, but with the standard deviation, the truth value is within the standard deviation estimates coming from the predicted values.

10% Inflation

For the next experiment, the ensemble of 7 was kept, but the multiplicative inflation used during DA was increased to 10%. Figure (5.20) shows the plots for the x,y and z variables. The results here are an improvement on when no inflation was used. The system does not collapse to zero as it did previously, but it does deviate from the truth at times.

The results for the different parameters predicted by EnKF are plotted in figure (5.23). The predictions for the different parameters as the system evolves is plotted in figure (5.23a). Looking at the σ graph, the distribution of the values was narrower than before with the predictions staying close to the truth during the rum. There are a few places where the value predicted goes to zero, but the system seems to fix the problem and get back to the correct area of the state space. The r value distribution also seems narrower than before and closer to the truth. There is a section where the prediction deviates from the truth significantly but the system does correct the values and the prediction, and then does stay close to the truth. The b values also seen closer



(a) Parameter Estimation throughout the run.

(b) Estimation of σ, the red line indicates the true value.

(c) Estimation of r, the red line indicates the true value.

(d) Estimation of b, the red line indicates the true value.

Figure 5.21: Python simulation with an ensemble of 7, data assimilation at every 50 time steps, a $dt = 0.01$ and no inflation. In (a), the blue lines are the ensemble prediction and the red lines are the true values. In (b), (c) and (d) histograms of the estimated parameters are shown, with a red vertical line for the truth value.

to the true value. When comparing the three graphs all the predictions move away from the truth together around at $t = 90$ mark. As expected, when this happened the values for x, y and z also diverge from the truth.

The histogram for σ is plotted in figure (5.23b). With the higher inflation, the spread of the prediction is smaller than before. The mean predicted by the system is $\sigma = 9.724 \pm 3.288$, while the true value used was $\sigma_T = 10$. The mean predicted values of this parameter in the run is much closer than before, with the truth being within the standard deviation range. Most of the predicted values seem to be lower than the truth and this fact is also reflected in the mean calculated which is smaller than the truth.

The histogram for r is plotted in figure (5.23c). The mean predicted by the system is $r = 27.442 \pm 1.931$, while the true value used was $r_T = 28.0$. Here the distribution is much closer to the truth than observed in the previous study, as reflected by the mean and standard deviation. It is very impressive to see that the system can predict values so close to the truth but this might just be the system getting lucky on the run with the initial conditions which have to lead the system towards the right path.

Finally the histogram for b is plotted in figure (5.23d). The mean predicted by the system is $b = 2.364 \pm 1.252$, while the true value used was $b_T = 2.67$. The predicted mean is very close to the true value of the parameter with the standard deviation also being smaller than before.

It seems that on this run the system was very well when predicting the values of the parameters with all being very close to the true values. This might be due to the initial conditions used by the system which lead to the system not deviating too much from the true values.

20% Inflation

Keeping an ensemble 7 the multiplicative inflation used during DA is increased further to 20%. Figure (5.22) shows the plots for the x,y and z variables. The results stay close to the truth but the ensemble still collapses to zero from time to time throughout the simulation. There are no major deviations from the truth, only the z variable has some large values at times.

The results for the different parameters predicted by EnKF are plotted in figure (5.24). The predictions of the different parameters as the system evolves is plotted in figure (5.24a). While the value from b remains close to the true value, but the value of σ and r are more chaotic then what was observers before. They both deviate from the truth with the predicted value of σ being larger then the truth and the prediction of r being much smaller than the truth. With the larger inflation, the one would expect the system to be more stable and more accurate. But since the combination of the three variables gave stable answers is seems like EnKF was never able to get in the right neighbourhood to get accurate values, especially for σ.

The histogram for σ is plotted in figure (5.24b). The mean predicted by the system is $\sigma = 17.252 \pm 4.702$, while the true value used was $\sigma_T = 10$. As described before the prediction here quite a bit larger than the true value. Even with the large standard deviation, the system is not close to the truth.

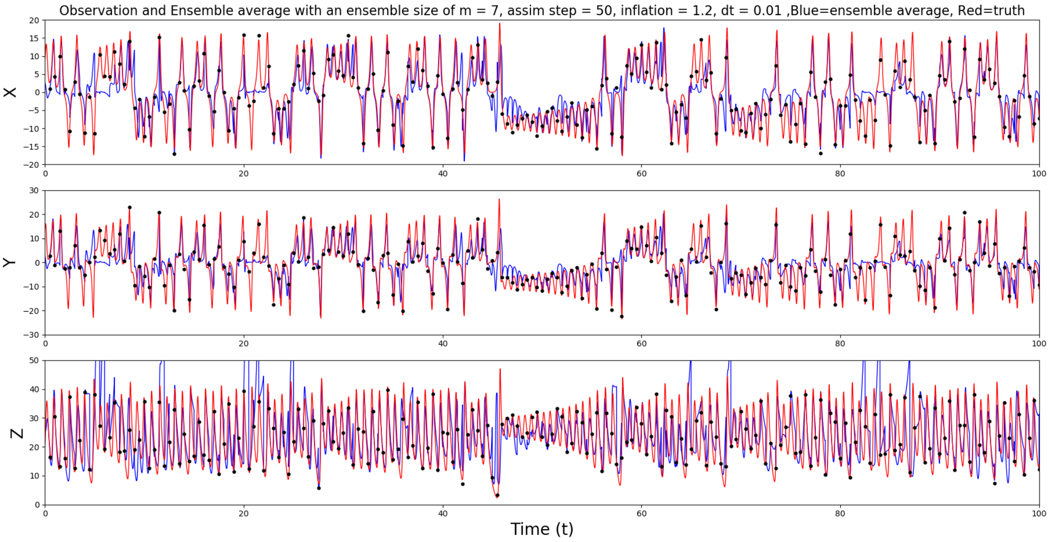
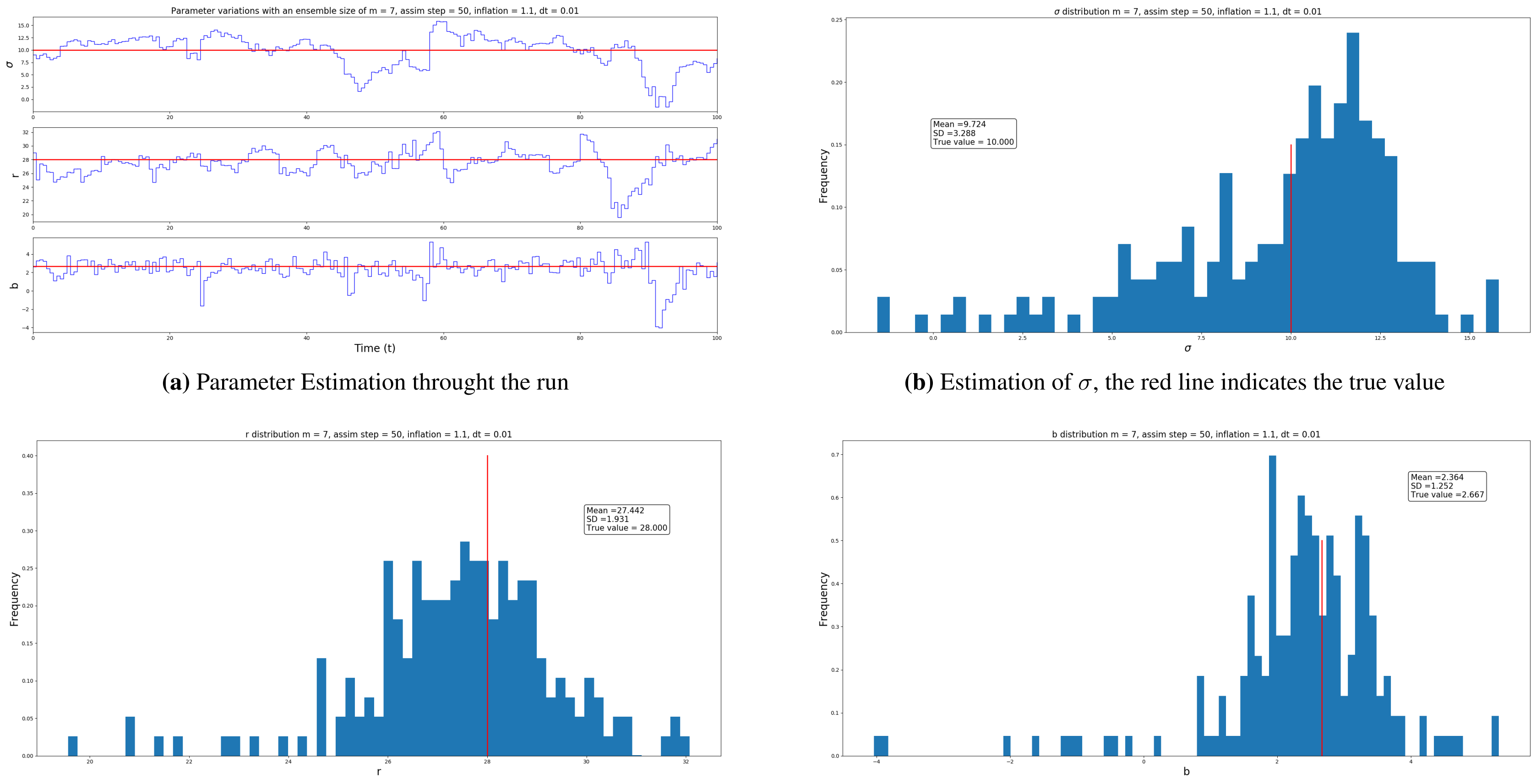


Figure 5.22: Plot of the ensemble average against the truth. Python simulation with ensemble of $m = 7$ data assimilation at every 50 time steps, a $dt = 0.01$ and no inflation. The red line is the observations and the blue is the ensemble average. The black dots are perturbed observation take during data assimilation



(a) Parameter Estimation throught the run

(b) Estimation of σ, the red line indicates the true value

(c) Estimation of r, the red line indicates the true value

(d) Estimation of b, the red line indicates the true value

Figure 5.23: Python simulation with an ensemble of 7, data assimilation at every 50 time steps, a $dt = 0.01$ and 10% inflation. In (a), the blue lines are the ensemble prediction and the red lines are the true values. In (b), (c) and (d) histograms of the estimated parameters are shown, with a red vertical line for the truth value.

The histogram for r is plotted in figure (5.24c). The mean predicted by the system is $r = 25.338 \pm 2.148$, while the true values used was $r_T = 28.0$. As observed previously, a large section of the prediction is lower than the true value. With the smaller standard deviation, the truth does not lie within the predicted mean value.

The histogram for b is plotted in figure (5.24d). The mean predicted by the system is $b = 1.909 \pm 1.442$, while the true values used was $b_T = 2.67$. The predicted mean here again close to the truth but interestingly the standard deviation is larger than the predicted mean. This large standard deviation might be due to the predictions given by EnKF when the system deviation from the truth during the 100 seconds and 150 seconds mark.

Ensemble of 10

The ensemble size of the system was now increased to 10 to see if a larger ensemble can lead to better results. Figure (5.25) plots she results for the different parameters predicted by EnKF for this simulation. The predictions of the different parameters as the system evolves are plotted in figure (5.25a). With the higher ensemble size, the predictions for r and b are close to actual values used but the system predicts higher values for σ for much of the run.

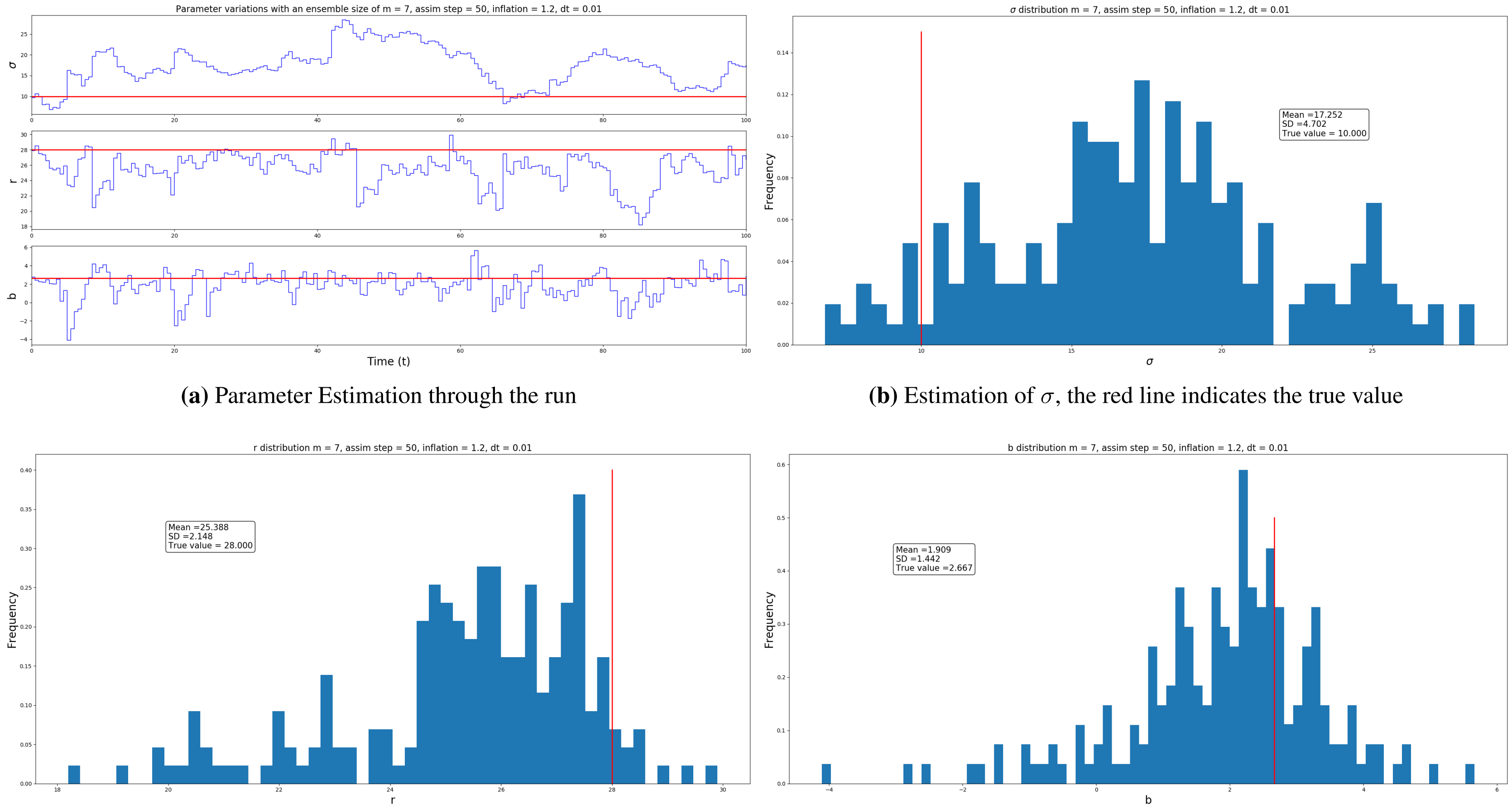
The histogram for σ is plotted in figure (5.25b). The mean predicted by the system is $\sigma = 113.645 \pm 2.678$, while the true value used was $\sigma_T = 10$. As observed in the previous plot the higher prediction of σ has to lead to a higher average value. The standard deviation is smaller than before as most of the ensemble skewed towards the high values.

The histogram for r is plotted in figure (5.25c). The mean predicted by the system is $r = 27.609 \pm 1.434$, while the true value used was $r_T = 28.0$. The mean prediction here is closer than what was predicted before, with a smaller standard deviation. The histogram shows a double peak with the second peak being closer to the truth.

The histogram for b is plotted in figure (5.25d). The mean predicted by the system is $b = 2.553 \pm 0.748$, while the true value used was $b_T = 2.667$. The mean prediction is close to the true value and the standard deviation is smaller than what was observed with inflation in previous studies. The distribution of the prediction is more like a Gaussian than seen for the other parameters indicating that EnKF has done a good job of predicting the values around the truth.

Ensemble of 10 with 10$\%$ Inflation

The results for the different parameters predicted by EnKF are plotted in figure (5.26). The predictions of the different parameters as the system evolves are plotted in figure (5.26a). The predictions here are less chaotic than previously observed when no inflation was used. Unlike previously observed the predictions for σ, r and b are very close to the truth throughout the run. The values for σ diverge during the early stages of the simulation around the $t = 20$ mark, while the values r and b deviate towards the end of the run around the $t = 90$ mark where values for both are under predicted. These deviations will affect the mean predictions for the parameters.



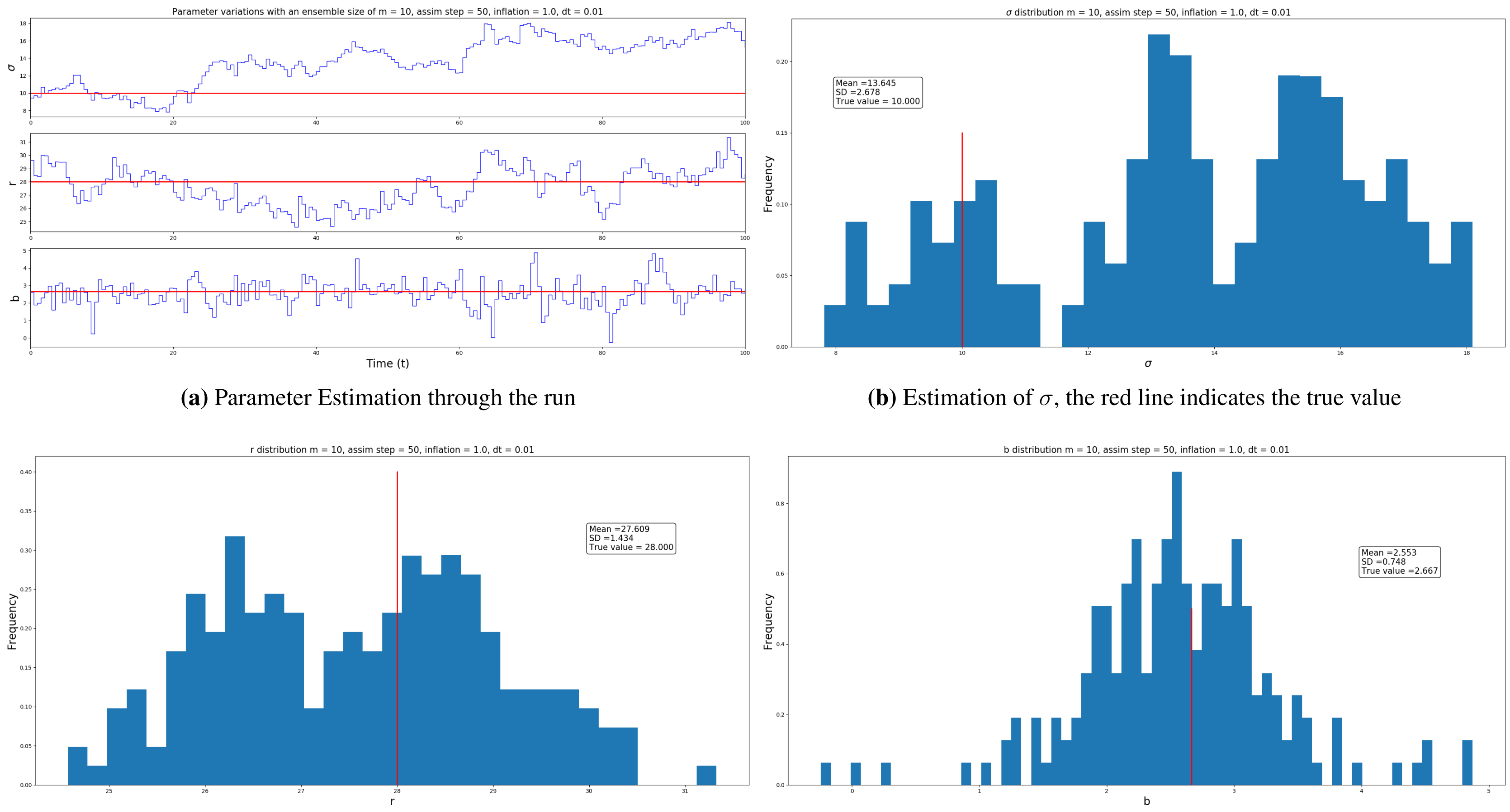
(a) Parameter Estimation through the run

(b) Estimation of σ, the red line indicates the true value

(c) Estimation of r, the red line indicates the true value

(d) Estimation of b, the red line indicates the true value

Figure 5.24: Python simulation with an ensemble of 7, data assimilation at every 50 time steps, a $dt = 0.01$ and 20% inflation. In (a), the blue lines are the ensemble prediction and the red lines are the true values. In (b), (c) and (d) histograms of the estimated parameters are shown, with a red vertical line for the truth value.



(a) Parameter Estimation through the run

(b) Estimation of σ, the red line indicates the true value

(c) Estimation of r, the red line indicates the true value

(d) Estimation of b, the red line indicates the true value

Figure 5.25: Python simulation with an ensemble of 10, data assimilation at every 50 time steps, a $dt = 0.01$ and no inflation. In (a), the blue lines are the ensemble prediction and the red lines are the true values. In (b), (c) and (d) histograms of the estimated parameters are shown, with a red vertical line for the truth value.

The histogram for σ is plotted in figure (5.26b). The mean predicted by the system is $\sigma = 10.342 \pm 1.851$, while the true value used was $\sigma_T = 10$. With the inflation, the mean predicted value is closer to the truth than before and the standard deviation is lower than before.

The histogram for r is plotted in figure (5.26c). The mean predicted by the system is $r = 26.627 \pm 1.841$, while the true value used was $r_T = 28.0$. The mean predicted here is smaller than what was predicted when no inflation was used. This difference might be due to the deviation that was observed towards the end of the simulation where the predicted value of r well below the truth.

The histogram for b is plotted in figure (5.26d). The mean predicted by the system is $b = 2.469 \pm 0.992$, while the true value used was $b_T = 2.67$. The predicted mean here is close to the true value but the values is lower then what was predicted before. The standard deviation observed is also larger than what was observed before. Both of these are because of the deviation that was observed near the end of the simulation.

Ensemble of 15

In the light of the improvement resulting from increasing the ensemble size to 10, the ensemble size was now increased to 15, with the results for the different parameters predicted by EnKF being plotted in figure (5.27). The predictions of the different parameters as the system evolves are plotted in figure (5.27a). The value of σ oscillates a lot around the truth throughout the run. The values of r and b fluctuate a lot as during the simulation but the changes are less extreme than what is observed for σ.

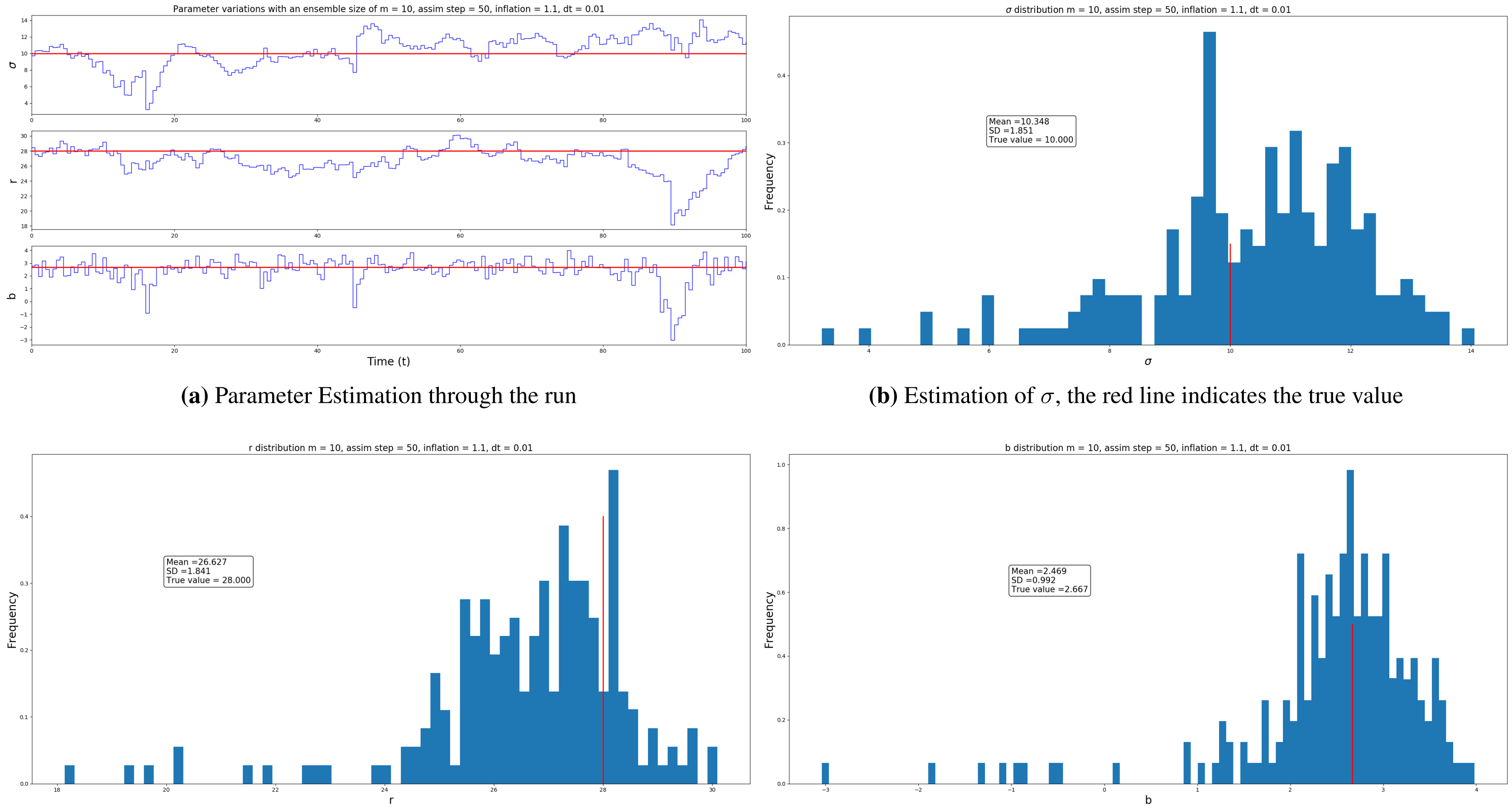
The histogram for σ is plotted in figure (5.27b). The mean predicted by the system is $\sigma = 9.893 \pm 1.526$, while the true value used was $\sigma_T = 10$. The mean here is very close to the true value and even the standard deviation is smaller then what has be observed before. As stated previously the prediction of the values of σ fluctuated a lot during the simulation but it seems on average it leads to an accurate prediction.

The histogram for r is plotted in figure (5.27c). The mean predicted by the system is $r = 27.033 \pm 1.170$, while the true value used was $r_T = 28.0$. The mean here is very close to the true value as well. The standard deviation is also small here, but this might be due to the fluctuations being small as observed in figure (5.27a).

The histogram for b is plotted in figure (5.27d). The mean predicted by the system is $b = 2.642 \pm 0.604$, while the true value used was $b_T = 2.67$. The mean predicted value is again very close to the true value with a very small standard deviation. This reflects how the prediction of b does not diverge too far from the true value.

Ensemble of 20

The ensemble number was further increased again to 20 and the results for the different parameters predicted by EnKF are plotted in figure (5.28). The predictions of the different parameters as the system evolves is plotted in figure (5.28a). The predictions for σ deviate for a while during the simulation, but not to extremely large values as observed in the previous studies. DA does a good job of pulling it back towards the true



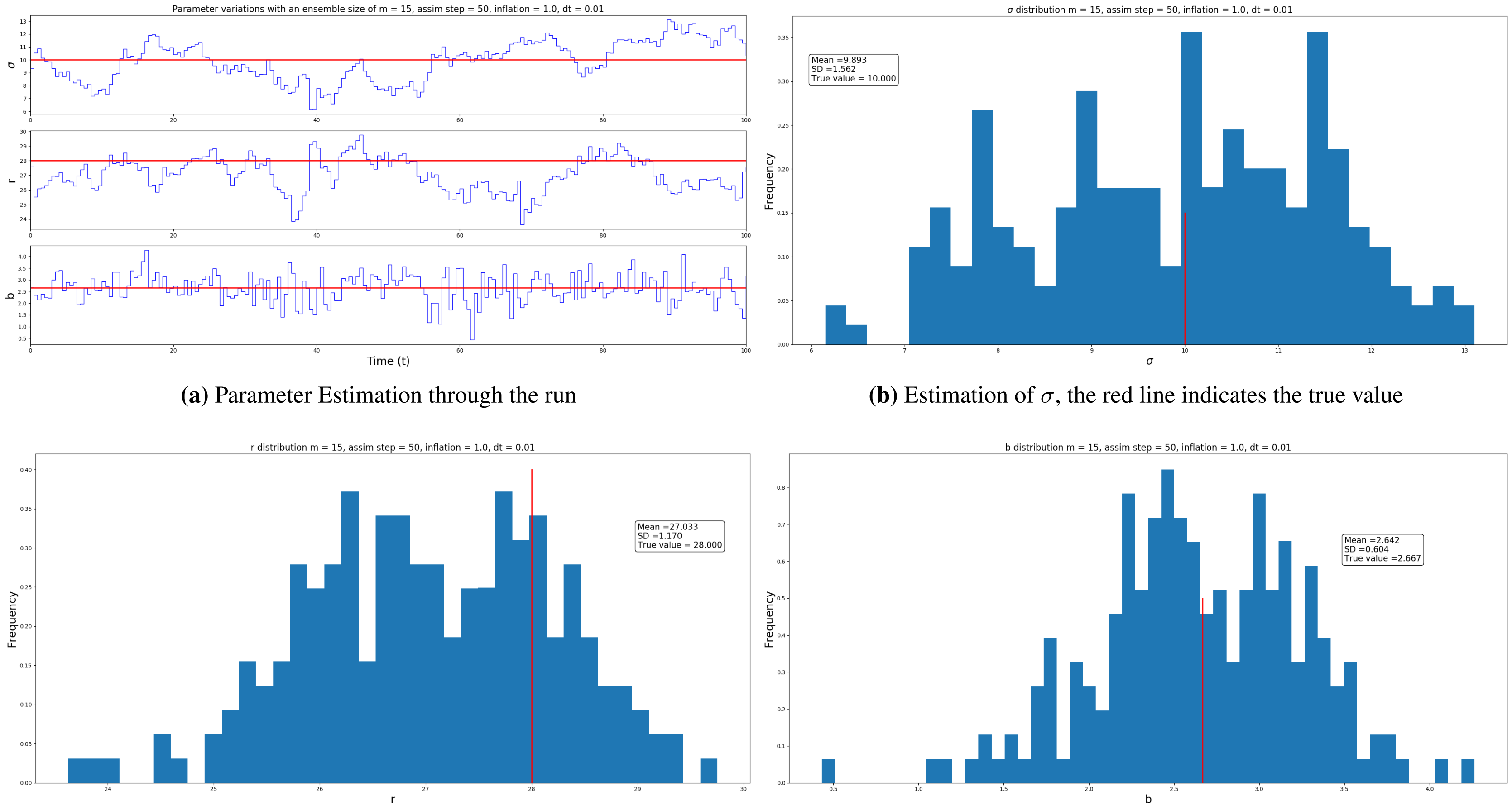
(a) Parameter Estimation through the run

(b) Estimation of σ, the red line indicates the true value

(c) Estimation of r, the red line indicates the true value

(d) Estimation of b, the red line indicates the true value

Figure 5.26: Python simulation with an ensemble of 10, data assimilation at every 50 time steps, a $dt = 0.01$ and 10% inflation. In (a), the blue lines are the ensemble prediction and the red lines are the true values. In (b), (c) and (d) histograms of the estimated parameters are shown, with a red vertical line for the truth value.



(a) Parameter Estimation through the run

(b) Estimation of σ, the red line indicates the true value

(c) Estimation of r, the red line indicates the true value

(d) Estimation of b, the red line indicates the true value

Figure 5.27: Python simulation with an ensemble of 15, data assimilation at every 50 time steps, a $dt = 0.01$ and no inflation. In (a), the blue lines are the ensemble prediction and the red lines are the true values. In (b), (c) and (d) histograms of the estimated parameters are shown, with a red vertical line for the truth value.

value. The value of r seems to remain quite close to the true value while the value of b fluctuates very close around the true value as usual.

The histogram for σ is plotted in figure (5.28b). The mean predicted by the system is $\sigma = 10.582 \pm 1.826$, while the true value used was $\sigma_T = 10$. As with the previous increase in ensemble size, the mean prediction stays close to the actual value used in the observation. The standard deviation remains small as well just as it did with an ensemble of 15.

The histogram for r is plotted in figure (5.28c). The mean predicted by the system is $r = 27.764 \pm 1.557$, while the true value used was $r_T = 28.0$. The mean prediction is very close to the true value with the standard deviation being small. The histogram for b is plotted in figure (5.28d). The mean predicted by the system is $b = 2.669 \pm 0.642$, while the true value used was $b_T = 2.67$. The mean prediction here is nearly the exact value of the one used to generate the observations.

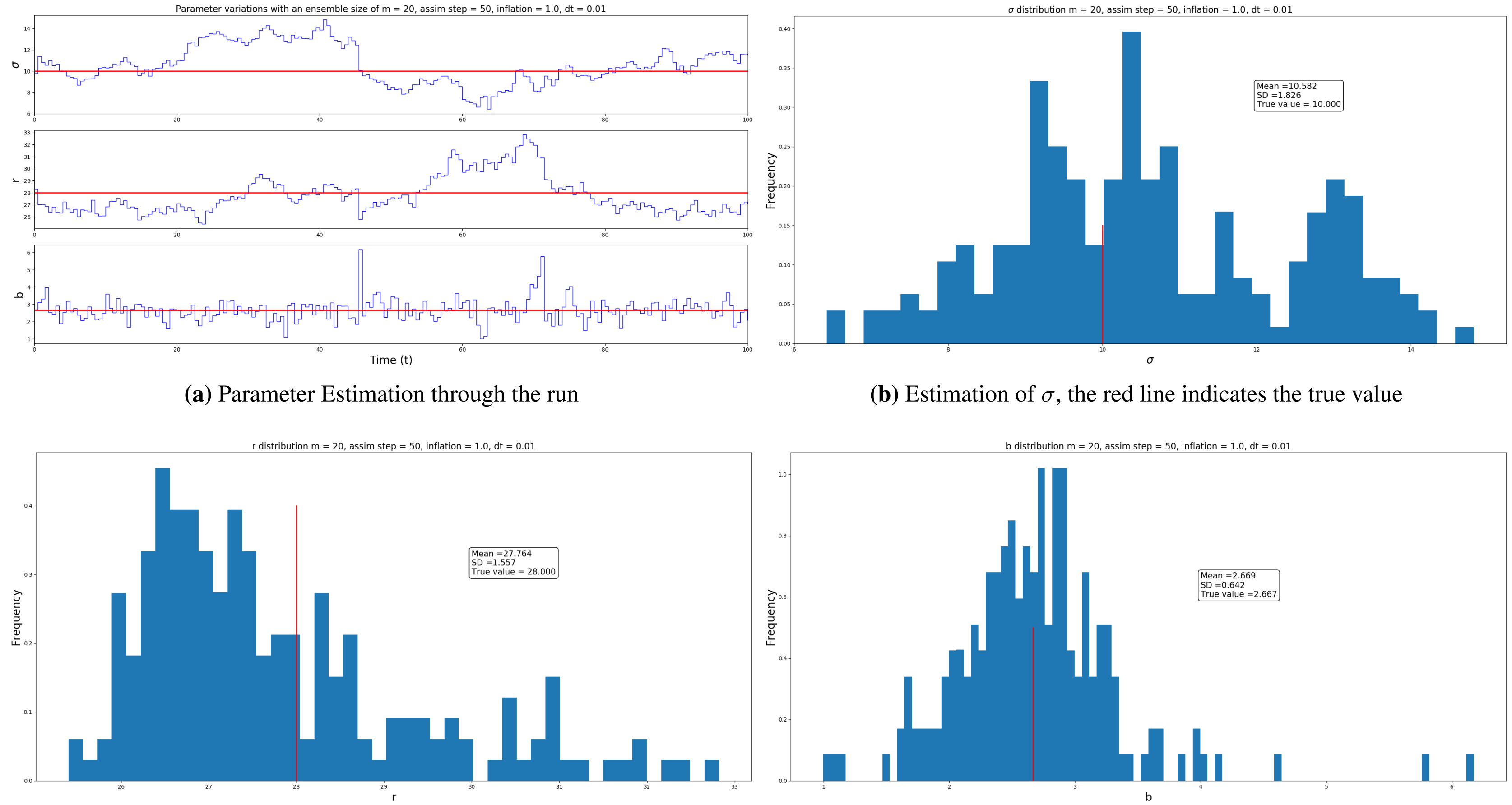
5.5.5.1 Discussion

Parameter estimation is the study of letting the DA method predict the values of the parameters in a system along with values of the normal variables. In this set of studies, the observation parameters used values of $\sigma = 10$, $r = 28$ and $b = 8/3$ as normal. When the simulation is initialised, the system randomly chooses the values for the parameters in each ensemble using $\sigma = 10 \pm 4.0$, $r = 28 \pm 5.0$ and $b = 8/3 \pm 0.5$, with all studies being done with a time step of $\delta t = 0.01$ and DA being done every 50 time steps. Table (5.1) provides a summary of the results presented in this section.

The first set of studies was done using an ensemble of 7, to study how EnKF behaves with and without multiplicative inflation. The first study was done without any inflation, with the study running for 100 time units. The results predicted for the parameters are generally very encouraging, even if the true value is not always within the standard deviation of the ensemble values. Using multiplicative inflation does make the ensemble more accurate but also lead to larger standard deviation which can lead to more chaotic results.

For the second set of experiments, the ensemble size was increased to 10. The first test was done without any inflation to see what the system does with this larger ensemble size. With the larger ensemble size, the predictions for the parameter are closer than before, with a smaller standard deviation as well. Applying a 10% inflation does not lead to a more accurate prediction overall, but this might be due to the more chaotic nature this run. Increasing the ensemble sizes to 15 and 20 results in more accurate predictions for the parameters and leads to a lower standard deviation. This is to be expected, as more of the state space is being sampled by the system. This leads to a more accurate prediction of error in the system.

It seems that increasing the ensemble sizes lead to a more accurate prediction for the parameters as expected. This can be clearly observed in figure (5.29) where the errors decrease with an increase in ensemble size and the predicted value also get closer to the true value. The addition of multiplicative inflation, however, can lead to higher standard



(a) Parameter Estimation through the run

(b) Estimation of σ, the red line indicates the true value

(c) Estimation of r, the red line indicates the true value

(d) Estimation of b, the red line indicates the true value

Figure 5.28: Python simulation with an ensemble of 20, data assimilation at every 50 time steps, a $dt = 0.01$ and no inflation. In (a), the blue lines are the ensemble prediction and the red lines are the true values. In (b), (c) and (d) histograms of the estimated parameters are shown, with a red vertical line for the truth value.

Ensemble	Inflation	σ	r	b
True	–	10.000	28.000	2.667
7	1.00	13.270 ± 6.856	24.931 ± 2.900	1.859 ± 1.514
7	1.10	9.724 ± 3.288	27.422 ± 1.931	2.364 ± 1.252
7	1.20	17.252 ± 4.702	25.338 ± 2.148	1.909 ± 1.442
10	1.00	13.645 ± 2.678	27.609 ± 1.434	2.553 ± 0.748
10	1.10	10.342 ± 1.851	26.627 ± 1.841	2.469 ± 0.992
15	1.00	9.893 ± 1.562	27.033 ± 1.170	2.642 ± 0.604
20	1.00	10.582 ± 1.826	27.764 ± 1.557	2.669 ± 0.642

Table 5.1: A summary of the results obtained during parameter estimation.

deviation but not necessarily a more accurate prediction

5.6 Conclusion

In this chapter, EnKF was applied to the Lorenz model and various experiments were conducted to learn about the behaviour of EnKF under different conditions. This behaviour will inform how EnKF should be applied to the MORALS code, which solves the Navier-Stokes equation for a fluid rotating in a thermal annulus with a fully 3D system.

An EnKF code written in MATLAB was used as a basis to write a Python version. This version was used to test EnKF's capabilities. In the first set of tests, the ensemble sizes were changed to see how the accuracy of EnKF changes. It was observed that increasing the ensemble size can dramatically increase the accuracy of the model. The Lorenz system has two fixed points, and the system transitions between the neighbourhoods of these fixed points, and generally finds it tricky to accurately predict where the observations would be. This was especially the case when the observations oscillated between the two fixed point regions. Given time, the ensemble system will always diverge, but with larger ensemble sizes the probability of this happening becomes smaller and the speed at which the system corrects this divergence also increases with larger ensemble sizes.

In the second set of experiments, multiplicative inflation was used along with different ensemble sizes. Overall, inflation helped a lot when it came to better tracking the observations. Smaller ensembles needed higher inflation and larger ensembles needed smaller inflation when it came to better tracking the observations. This is because a larger ensemble samples more of the state space, hence lead to a better covariance and Kalman gain. However, over inflating the ensembles seems to not lead to any improvements in the accuracy of the system and in some cases can lead the system to be more inaccurate, as now the Kalman gain gives a higher value leading to an overcorrection which can make the ensemble members diverge. One way to solve this problem in

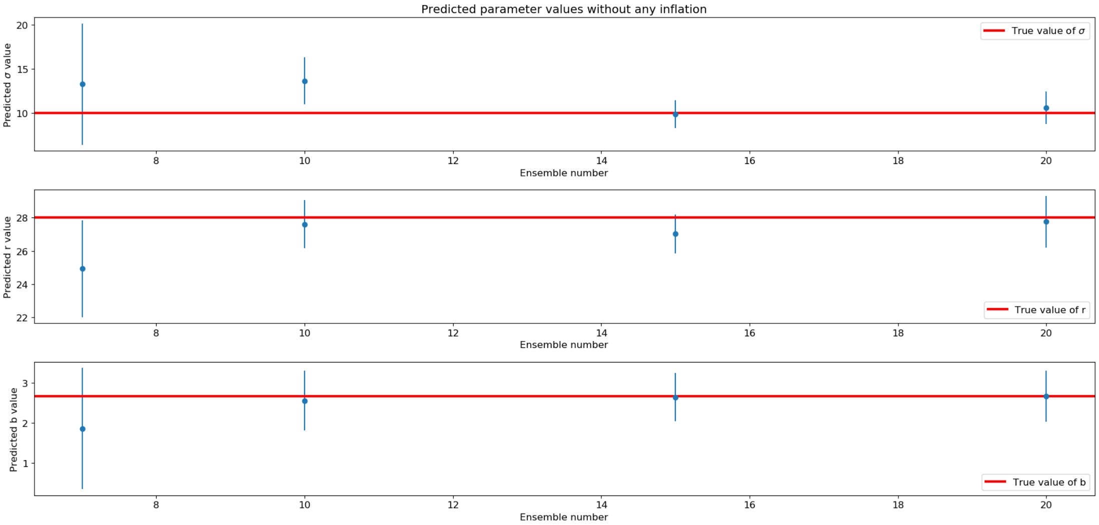


Figure 5.29: Plot of the predicted parameter values without inflation. The red line is the true value, the blue dots are the predicted average and the lines are the standard deviation.

higher dimensional systems would be to use localisation to stop false correlation between points which might lead to unnecessary overcorrection of the system. Inflation is more useful when used for smaller ensemble sizes as the user is saving on the computational cost of simulating additional ensemble members.

In the third set of experiments, the number of time steps between the DA points was changed. As theorised, with the larger interval between DA the system finds it harder to track the truth as the ensemble members have more time to diverge. But in most cases using multiplicative inflation can help stop the system from diverging from the truth. Obviously, with the longer interval between DA, the amount of inflation needed to accurately track the truth is higher than what was needed with the shorter interval.

It is clear that for a system like the Lorenz model there is a range of time intervals between DA where EnKF does well in tracking the truth. Given the chaotic nature of the Lorenz model, there also seems to be a limit on the time between DA, after which even increasing the ensemble number does not lead to perfect results. This was demonstrated with a time interval of 100 time steps, where even using an ensemble of 14 and inflation did not yield perfect results.

Finally, a study was done on the capabilities of EnKF on parameter estimation. Here the parameters of σ, r and b were predicted by EnKF along with the normal variables x, y and z. Overall EnKF does a great job in getting the predictions close to the actual values used for the parameters. With an ensemble of 7, inflation is needed to get EnKF close to the true values, but as the ensemble size is increased, less or even no inflation is required to get accurate predictions of the parameters. This result is comparable to previous studies such as Annan and Hargreaves (2004), where a much larger ensemble, up to 100 members, was used to obtain close predictions.

This chapter shows the capabilities of EnKF under different conditions to accurately correct and predicts the behaviour of the Lorenz model. In all different conditions, EnKF can be tuned to work well. These results will help shape the application of EnKF to the MORALS code, where EnKF will be applied to a 3D model. The number of points which can be observed in this system is more than 100,000 and will further push the capabilities of EnKF.

Chapter 6

Experimental Results

6.1 Introduction

With a lack of computational power in the early days of weather prediction, laboratory experiments were used to try to more closely replicate atmospheric conditions and to understand fundamental physics that underlies atmospheric behaviour (Hide, 2010). As computational methods became better, experiments also started being used as testbeds to see how well computational simulations can replicate experimental behaviour. As modern technology has made computational studies easier and accurate it has also made experiments more accurate, using modern apparatus and equipment. As such, experiments still play a very important role in understanding atmospheric behaviour and being used as a testbed for computational methods.

This chapter presents results for experiments performed at the Atmospheric, Oceanic and Planetary Physics (AOPP) Group at the University of Oxford. Previous studies performed at AOPP utilised an older experimental apparatus for a rotating thermal annulus, see Hignett (1985) for more details on the setup. In the study performed by Young and Read (2013) using this setup, the rotation rate of the annulus was changed every 20 minutes. These experiments focused on how the behaviour of the fluid changed with the rotation rate i.e. how wavenumber changes when increasing or decreasing rotation rates.

The results presented in this chapter are from experiments which were done using a newly built experimental apparatus and focused on long runs at a single rotation rate. These runs were used to test the new apparatus, along with examining the long term behaviour of the fluid and to see how stable the waves formed at these rotation rates are. Experiments were performed at 1 rad/s, 2.5 rad/s and 3 rad/s. Particle image velocimetry (PIV) was applied to the experimental data using DynamicStudio to generate two-dimensional U and V velocity data. The results obtained in this chapter will be used as observations for data assimilation using the Ensemble Kalman filter and testing its capability to track real life observations against an ensemble of low resolution numerical simulations using the MORALS code. The details of the study can be read in chapter (8).

The experimental setup is described in section (6.2) along with the values of the different parameters that were used for the experiment. This is followed by an overview of the PIV setup in section (6.3) which describes how the image was processed to obtain the U and V velocity. Finally the results of the experiment are presented in section (6.4) with results for 1 rad/s, 2.5 rad/s and 3 rad/s presented in section (6.4.2), (6.4.3) and (6.4.4) respectively.

6.2 Experimental Setup

The experimental setup uses a new thermal annulus but is essentially the same as described by Hignett (1985) in terms of design, but also contains eight temperature sensors on each of the inner and outer walls at different levels. The annulus has an inner cylinder radius, $a = 2.5$ cm, an outer cylinder radius, $b = 8.0$ cm and a depth, $d = 14.0$ cm. In all our studies the inner cylinder is cooled at $T_a = 18.0°C \pm 0.02°C$ and the outer cylinder is kept at a temperature of $T_b = 22.0°C \pm 0.02°C$. The temperature of the cylinder walls is kept constant by pumping a fluid of different temperature to their respective walls using a pump. The temperature of the liquid is regulated at the pump to obtain the temperatures specified by the user. The bottom of the annulus is insulating and a transparent lid can be placed at the top of the annulus to create a rigid top boundary.

The working fluid used in the experiment is a 17% glycerol mix by volume where glycerol is mixed with de-ionised water. The density of the mix is measured using a hydrometer to get the correct volume fraction. During the mixing of glycerol and water, non-buoyant particles are added which reflect light and are used to track the flow of the fluid. After this mix is poured into the annulus it is left to rest for an hour or so, to let any air bubbles rise to the top and escape. After this, the top lid is added to the annulus to create a rigid boundary and experiments can commence.

As mentioned before there are eight temperature sensors on the inner wall and eight on the outer wall. These are type T thermocouples used in conjunction with a Pico TC08 thermocouple logger to store temperature data. The position for thermocouples on the outer cylinder were labelled as (from top to bottom): 1, 8, 7, 2, 6, 3, 5, 4; for the inner cylinder, they are (from top to bottom): 1, 2, 3, 4, 5, 6. On the inner cylinder, sensor 7 is for common constant connection and sensors 8 and 9 are the earth. Data is logged at every level simultaneously after image capture. This comes to data capture every 6.5 seconds.

The annulus has five 5 mm slits cut at different heights of the cylinder. These slits go around the outer wall so light can be projected into the annulus. They are present at 12.4, 9.7, 7.0, 4.3 and 1.6 cm from the bottom of the annulus. Simple LED lights are attached to each of these level (the silver attachments in figure 6.1) that shine through these glass slits. This light is reflected by the seeding particles and was used to view the

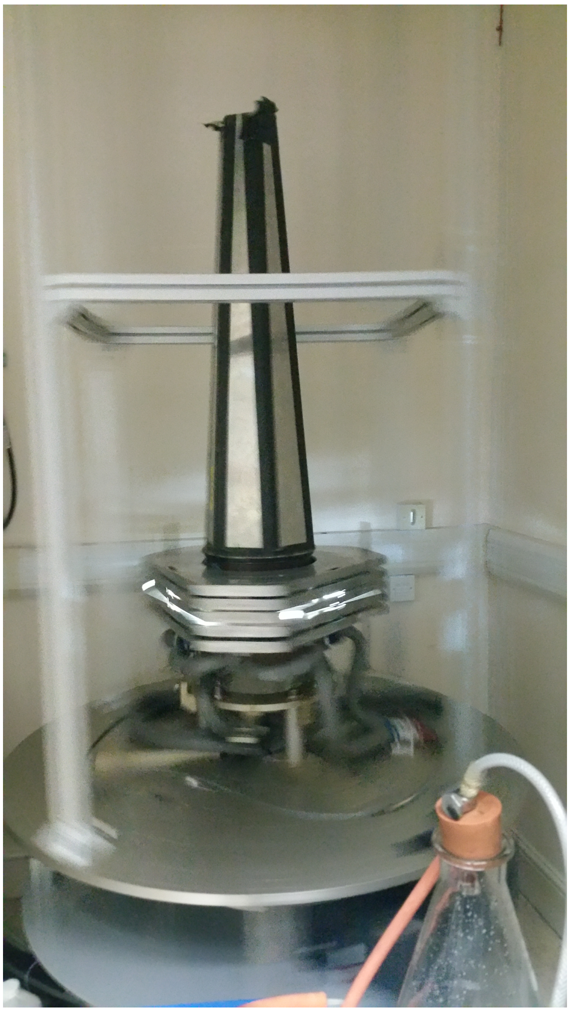


Figure 6.1: The experimental apparatus. The black shroud is used to block light from the surrounding for better image capture. The five silver attachment shine light at the five different levels of the annulus.

Annulus parameters	
Inner cylinder radius	a=2.5 cm
Outer cylinder radius	b=8.0 cm
Depth of fluid	d=14.0 cm
Inner cylinder temperature	T_a=18.0°C
Outer cylinder temperature	T_b=22.0°C
Fluid properties	
Working fluid	83% water, 17% glycerol
Density	$\rho = 1.044$g/cm^3
Slit position from the bottom of the annulus	
Level 1	1.6 cm
Level 2	4.3 cm
Level 3	7.0 cm
Level 4	9.7 cm
Level 5	12.4 cm

Table 6.1: Summary of Annulus and fluid properties used in experiments

behaviour of the fluid at that level. Light is projected through each level sequentially for 5 seconds starting from the bottom and moving towards the top. There is a 1 second delay before the next level is illuminated, this is done so there is no overlap of different levels being illuminated at the same time leading to false data. Once the top level has been illuminated the cycle starts again from the bottom level. Hence images are obtained from a level every 30 seconds.

The thermal annulus is attached to a table which rotates with a speed set by the user using an electric motor. A camera is attached 1 m above the rotating table and rotates with the table and takes an image which is 1024x768 pixels. A shroud is put on top of the annulus to block any light from the surroundings. This helps to get a better view of the light reflected from the seeding particles. In addition, the light of the room was also turned off to create a darker environment which should lead to better image capture.

The process of illumination and image capture from the camera are both controlled using a Python script created at AOPP. A series of 10 images were captured starting one second after light is shone at a particular level. The one second delay was added so that the camera only starts capturing the images once the level is properly lit. These images are taken roughly 0.16 seconds from each other. The images obtained using this process can then be used to calculate variables such as U and V velocities for that level using a PV software as described later in section (6.3).

During the initial testing, there were a lot of issues with bubbles forming in the working fluid and getting trapped by the top lid. This obstructed the view of the camera as shown in figure (6.2), and a clear image of the fluid at any level could not be captured properly. Owing to the persistence of this issue after multiple experimental attempts it

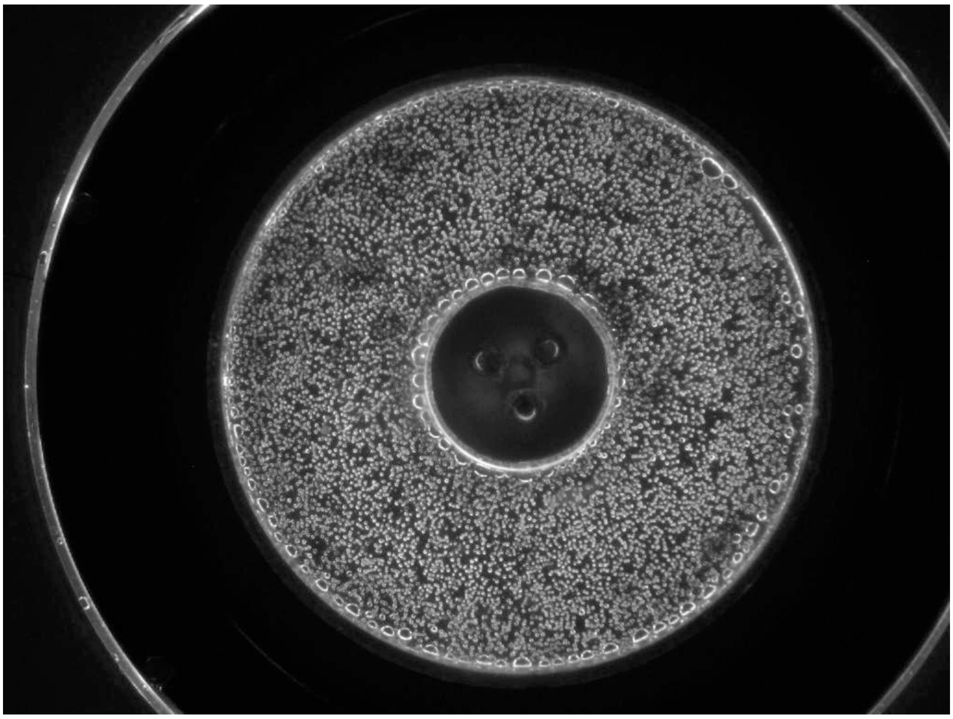


Figure 6.2: Bubbles trapped by the top lid blocking the view of the camera when capturing images for PIV

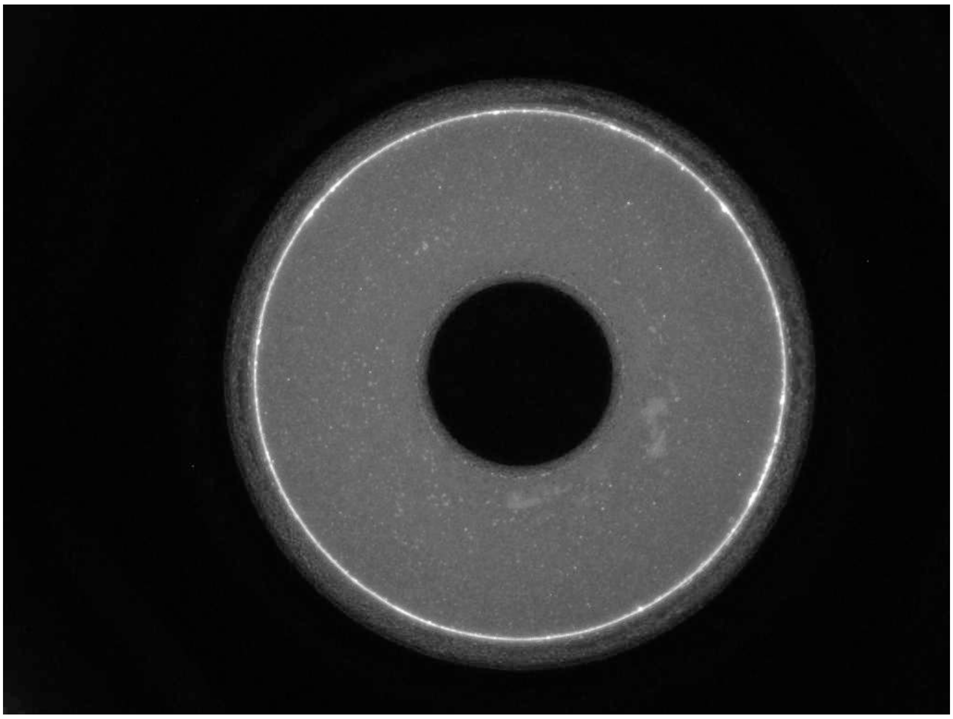


Figure 6.3: Example of an image used in PIV to create a mask. The inner and outer wall can be clearly seen and marked in DynamicStudios to create the mask.

was decided to conduct the experiments without the top lid. This will create different behaviour at the top of the working fluid as now the boundary is not rigid.

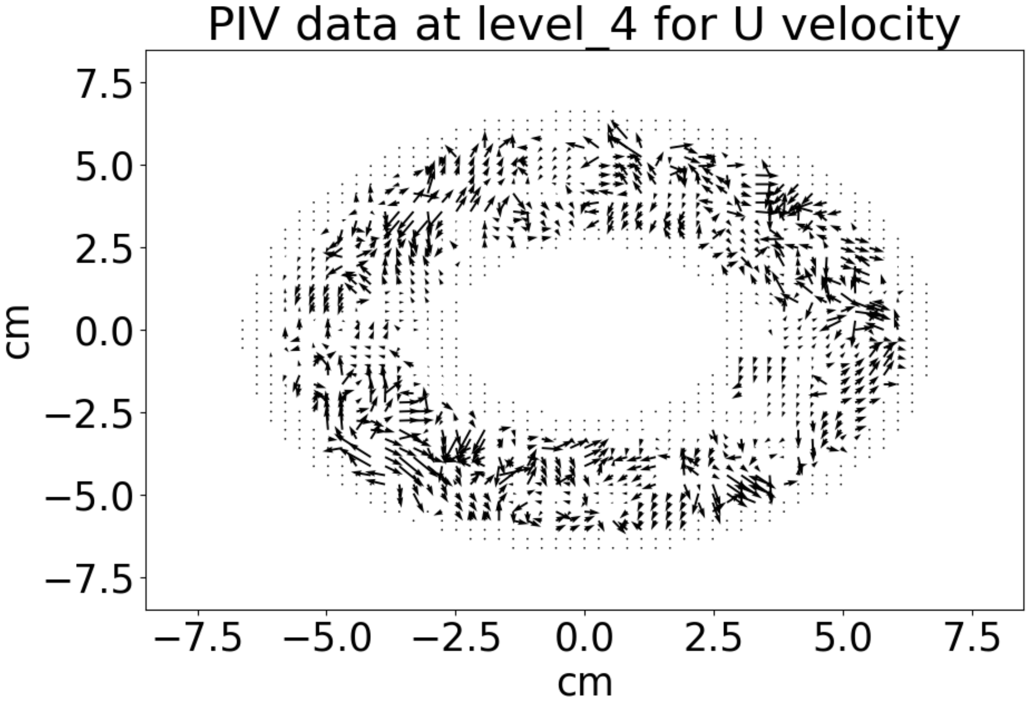
6.3 PIV

To process the image data obtained through the experiments software provided by Dantec Dynamics called DynamicStudio was used. At the beginning of the study, an image from the experiment was imported where a clear distinction between the inner and outer wall regions can be observed. For example in figure (6.3), one can see the position of the middle cylinder and where the outer wall starts, showing the edge of the annulus. This imported image was used to create a mask for the study. For this, areas of the image were highlighted to show 1) the location of the inner cylinder, 2) the working fluid region and 3) the location of the outer wall along with the region outside the annulus. This mask tells the software which parts of the image contain data and which don't. This step is very important otherwise the software will look at areas outside the working fluid and calculate erroneous data. Care also needs to be taken when calculating near the boundaries near the inner and outer wall.

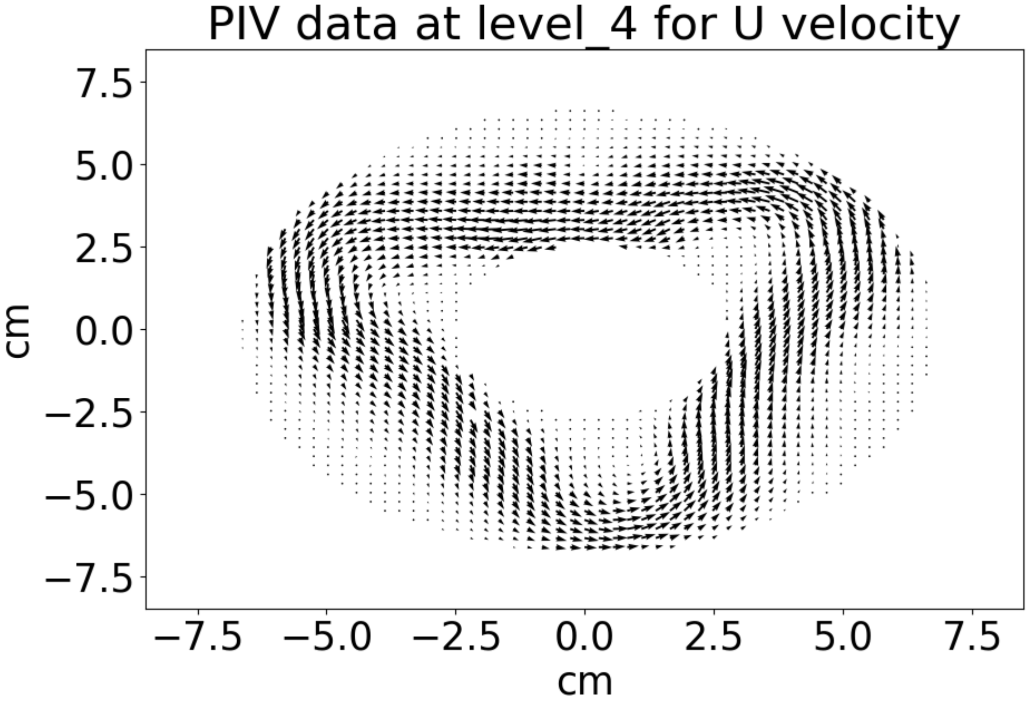
Using the mask, images from the different heights/levels for a run can now be imported to be processed. Images for each level are organised in individual branches to be processed separately. When importing the images, the system is told that the pictures are taken at 6.25Hz (i.e. every 0.16seconds). It is not possible in the software to account both for the 0.16 seconds between the 10 image set and 30 seconds difference between every 10 sets of images. Hence the data for first image in the set of 10 will be inaccurate as the system won't have accounted for the 30 second gap. Figure (6.4a) shows an example of this false data calculated by DynamicStudio. As seen here the vectors do not have any structure when compared with the correct data shown in figure (6.4b), where a clear wave is seen travelling around the annulus. When it comes to doing any further analysis of the PIV data the false calculated data are ignored.

The mask is applied to the images in each level. The image is 1024x768 pixels and an adaptive PIV option is chosen for the data processing using a grid step size of 12 pixels, with a minimum number of pixels of 24 and a maximum number of 48 for calculations. This means grid points are created every 12 pixels and the values of U and V variables were calculated using 24 to 48 pixels adaptively. After processing the data using adaptive PIV, the mask is applied again to the results to remove any invalid data points which might have been created near the boundaries.

After processing of the data is done the result for each image is then exported as a CSV file which can be read into Python to be further analysed and processed. The data file contains results for the whole image so includes data calculated outside the working fluid as well. The data points are marked by DynamicStudio to tell the user if they were calculated in a valid zone or not. A Python script was written to read in the exported files and sort out valid data which can be further analysed before being plotted. All the data calculated were in Cartesian coordinates hence the results are converted to Polar



(a) An example of PIV data for the first image after the 30 second gap showing incorrect data



(b) An example of correct PIV data for images

Figure 6.4: An example of the PIV data as calculated by DynamicStudios

coordinates. The image taken during the experiments was 1024x768 pixels while the processed data file contains 83x62 data points.

6.4 Results

This section presents the results for the three long run experiments that were performed at AOPP for 1 rad/s, 2.5 rad/s and 3 rad/s with images taken at five different levels. These different depths at which the images were taken will be referred to as level 1, level 2, level 3, level 4 and level 5 and refer to images taken at 1.6 cm, 4.3 cm, 7.0 cm, 9.7 cm, 12.4 cm respectively from the bottom of the annulus. Table (7.1) has a summary of the parameters used for the experiment.

Comparative figures are presented at each level to show what is observed in the experimental image and what is seen in the analysed PIV data. This is done to show the reader how much qualitative information can be ascertained by looking at the experimental image compared to applying PIV to it. Using the PIV data, Hovmöller (space-time) plots are also presented for each level to show how the working fluid drifts around the annulus. The plots are created from the flow by at the middle of the annulus (r=5.25cm) to ignore the effects of the boundary. This will also show if there are any structural changes during the experiment or if they are stable. Temperature data is also presented from eight temperature sensors on the outer wall of the cylinder which continuously log data around every 6.5 seconds.

Distinct behaviour is observed at each level with more recognisable behaviour being observed at higher levels. At lower levels, accurate observations are harder to obtain as the fluid and particles from higher levels obscure the camera. The density of the particles in the fluid is also very important for generating correct PIV data. With low rotation rates, the seeding particles can sink to the bottom of the annulus and with high rotation rates, particles can be pushed towards the walls of the annulus. In all cases, U velocity is plotted in the figures shown.

6.4.1 Errors and Variations in prediction

Previous study by Young and Read (2013) used a software called DigImage (Dalziel, 1994) to process their experimental data. The software error for the velocities calculated is given by equation (6.1). This error was derived assuming a uniform distribution of particles and using the standard deviation.

$$\sigma = \frac{b}{N_x t^{obs}} \sqrt{\frac{2}{3n^s}} \tag{6.1}$$

Here N_x (512) is the pixels in x direction, t_{obs} (1 seconds) is the observation time, n_s (5 images) is the number of images used for the calculation and b (8 cm) is the outer radius

Level	1 rad/s U	1 rad/s V	2.5 rad/s U	2.5 rad/s V	3 rad/s U	3 rad/s V
1	0.0771	0.0256	0.0360	0.0121	0.1543	0.0444
2	0.0292	0.0073	0.0323	0.0126	0.1197	0.0364
3	0.0353	0.0100	0.0165	0.0020	0.0475	0.0130
4	0.0311	0.0064	0.0409	0.0085	0.0409	0.0085
5	0.0259	0.0090	0.0326	0.0087	0.0527	0.0156

Table 6.2: Standard Deviation Error for rotation rates at different levels

of the cylinder. Using this method, the velocity error for results presented in Young and Read (2013) was calculated to be $0.0057\ cm/s$.

The software used to calculate velocities for our study is called DynamicStudios. This software does not function the same way as DigImage but averages over many pixels to calculate the values for the velocity, so the error equation used for DigImage does not strictly apply here. But for comparison, using the same equation gives a velocity error of $0.0564\ cm/s$ for DynamicStudios. This error is much higher compared to DigImage as the observation times are much smaller (0.16 seconds compared to 1 seconds) and only 2 images are used to calculate velocities in the PIV data.

Another study that could be done, is to see how the PIV software performs and look at its consistency when predicting velocity. A simple standard deviation can be calculated using the variations in the predicted amplitude of the baroclinic wave in the system as observed in each valid image. This variation in predicted values shows how accurately the PIV software is able to track the movement of particles in the images. And Although amplitude and structural vacillation could cause problems in calculations, this should not be a problem for steady flow observed at low rotation rate like 1 rad/s.

The results for the standard deviation at different rotation rates and at different levels are given in table (6.2) with figure (6.5) showing all the errors at the different rotation rates in one figure. At 1 rad/s and 3 rad/s is can be clearly observed that the deviation in the system decreases as the observation level increases. While at 2.5 rad/s there seems to be an increase in deviation at higher levels. It seems that vacillation might have played a role at this level when it comes to the deviation.

6.4.2 1 rad/s

The first results presented are for a run at 1 rad/s which ran for approximately three hours. At this low rotation rate, a very stable structure is produced with a dominant $m = 3$ wave travelling around the annulus. Due to this stability this rotation rate was used to benchmark the PIV software. Details of what was observed at each level is given below.

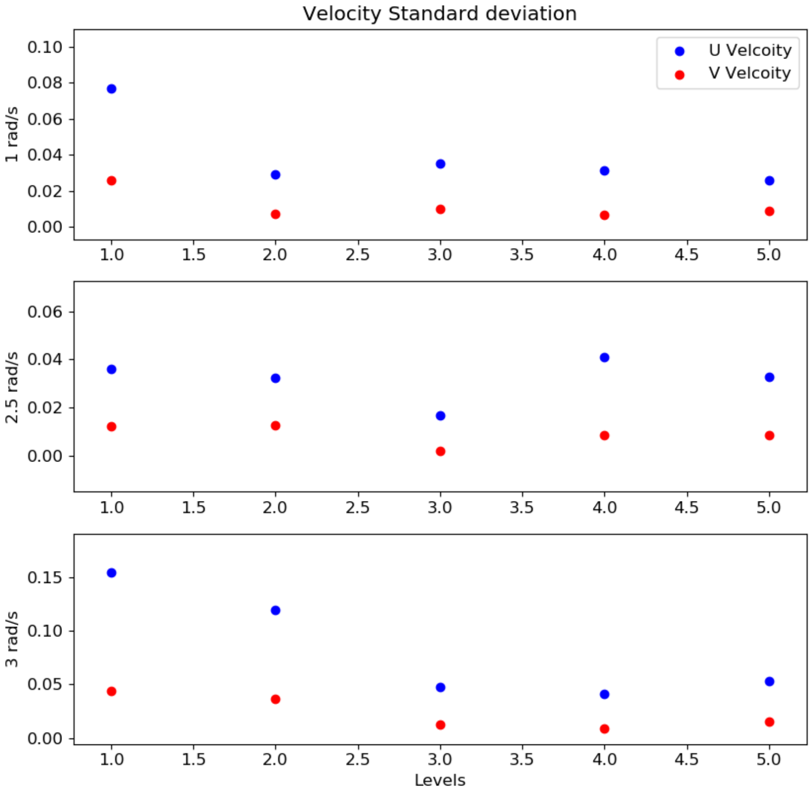


Figure 6.5: Standard Deviation Error calculated for each level and rotation rate using the amplitude of the wave in the middle of the annulus.

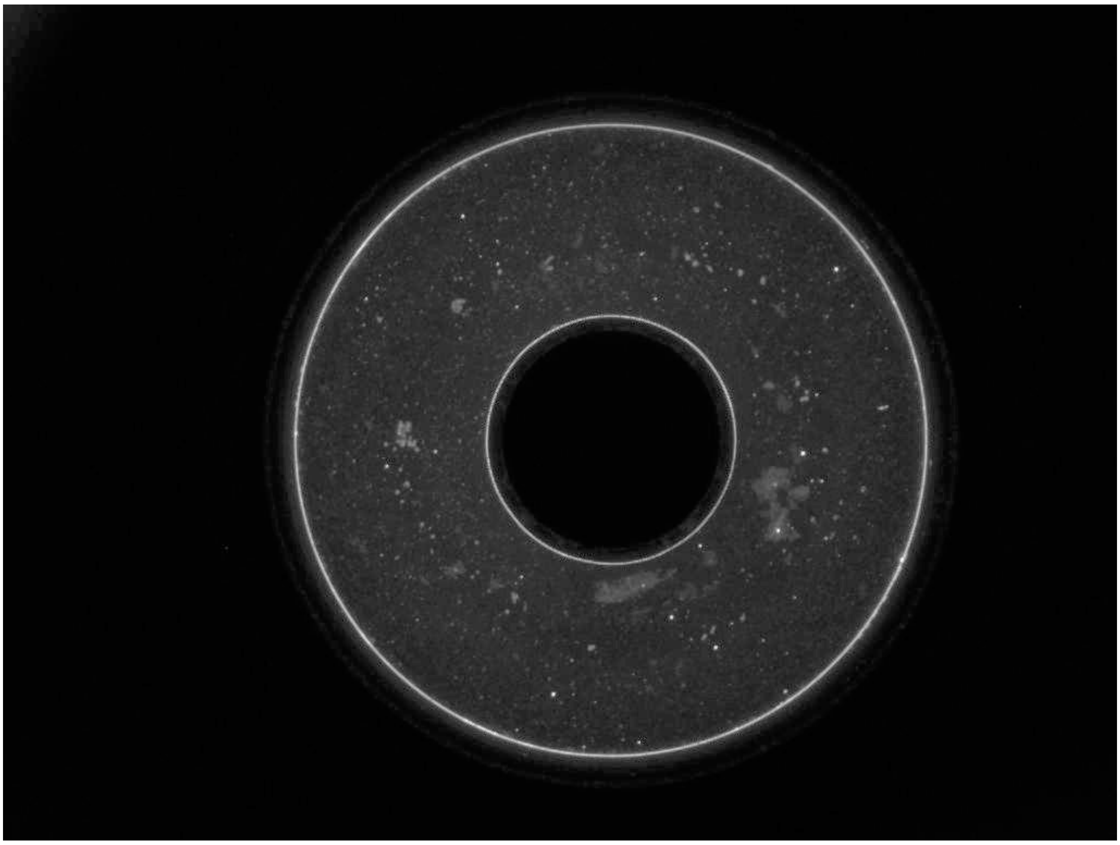


Figure 6.6: Image of the working fluid at the end of the 3hr run taken at level 1 for the 1 rad/s experiment

6.4.2.1 Level 1

Observing at this low level creates a lot of challenges as the fluid and seeding particles from above levels can obscure what is seen. The image taken at level 1 towards the end of the 3 hour experiment is presented in figure (6.6). There are particles which can be seen in the middle of the annulus and can be used to figure out the basic structure of the flow. No detail in the structure can be seen but a circular structure can be observed when following the particles. A sense of the movement in the fluid can be detected when observing sequences of images but it is difficult to pinpoint areas for structures. Hence PIV is needed to get a sense of the detail in the structure of the fluid.

The structure of the flow calculated out using the PIV software can be seen in figure (6.7). Here the U velocity of a sequence of 6 images taken at 6 continuous cycles is plotted. The messy structure observed could be attributed to a few factors such as interaction with the bottom boundary of the annulus. This might stop the fluid from forming stable waves near the bottom due to recirculation of the fluid. This recirculation could be linked to the thermal gradient between the inner and outer wall which causes the fluid to sink at the inner wall and rise at the outer wall. It could also be the case that observations from this level are not accurate enough due to the depth and the obstruction

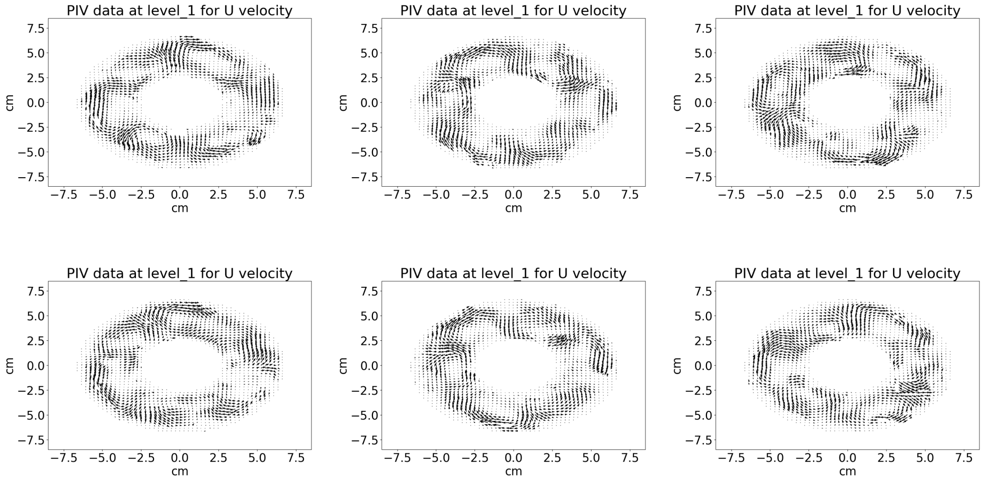


Figure 6.7: PIV result sequence towards the end of the experiment at level 1 for 1 rad/s showing U velocity for images taken 30 seconds apart.

from the non-buoyant particles and fluid at higher levels as mentioned before.

The Hovmöller plot for the U velocity at this level is shown in figure (6.8) where a 1000 second interval of data is plotted. This plot shows how the structures of the fluid move across θ, as the experiment goes on. Looking at this small section it can be observed that an $m = 3$ wave is travelling around the annulus. It also seems that the wave drifts steadily around the annulus.

6.4.2.2 Level 2

Moving up the annulus to level 2, similar behaviour to level 1 is observed from the images. An image taken from the end of the experiment is shown in figure (6.9). As before it is hard to distinguish the structures of the flow, but an idea can be garnered following a milky trail that is seen in the image. A flow can be observed travelling around the annulus in a circular pattern similar to level 1 but with one part of the flow reaching further out than the others. It seems that the depth and density of the particles might still be an issue when it comes to observing the movement of the fluid.

The plot for the PIV data for the above image plotting the U velocity is shown in figure (6.10). Moving up in depth seems to have helped in getting better observations as the structures are less messy than level 1. The structure of the fluid flow here seems to be a hybrid between the numerous vortices seen in level 1 and a wave like structure which is normally reported for this rotation rate. These relatively small scale vortices may not be real fluid flows but artefacts of the difficulty of observing the deeper levels with PIV. There are still data poor areas between vortices where structures seem to be missing from the flow and seem to reflect behaviour observed in level 1 observations. These data poor structures are visible throughout the experiment and move with the drift

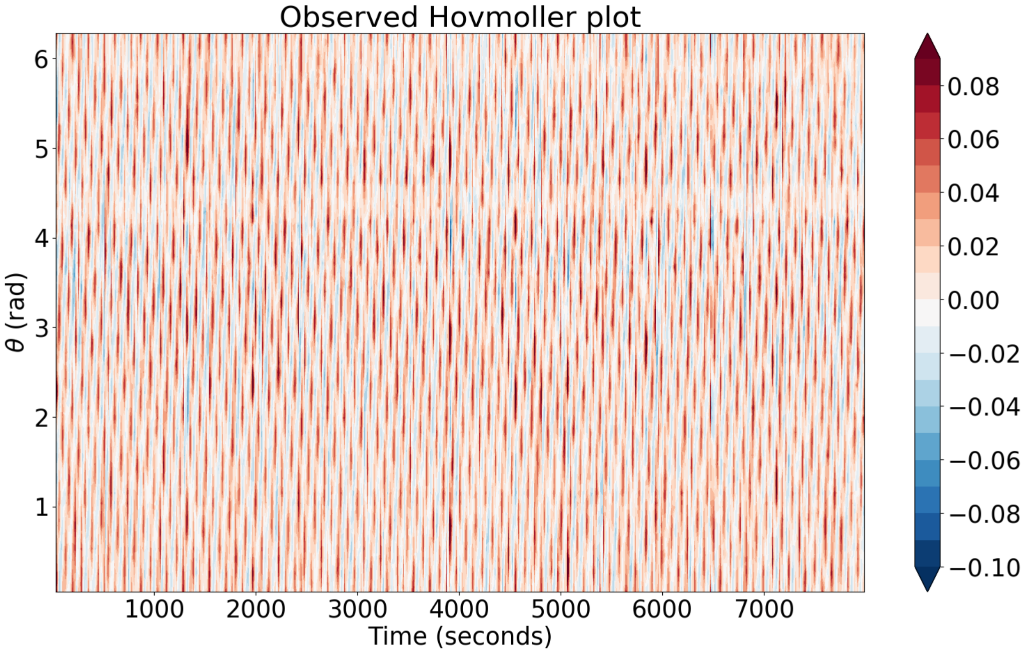


Figure 6.8: Hovmöller plot for level 1 at r=5.25cm for 1 rad/s showing U velocity. The vertical axis shows azimuthal position in degrees, and the horizontal axis is time.

of fluid. The lack of details there can be due to several reasons, from the depth of the observation to there not begin enough particles to track or that the particles are moving too slowly to be tracked, i.e. it takes more than the 10 image sequence for a particle to move.

The Hovmöller plot for level 2 is plotted in figure (6.11) and shows similar behaviour to that observed for level 1. There are a lot of bright and dimmer structures travelling around the annulus. This seems to indicate that not all structures have the same amplitude at all times. This is a good indication for amplitude vacillation as also reported by Young and Read (2013). There does not seem to be an indication of structural vacillation i.e. change of radius of the $m = 3$ structure at this level but this might just be due to the recorded observation only lasting around 1 second. A longer continuous observation might help shed more light if this behaviour occurs at this depth.

6.4.2.3 Level 3

Moving higher to level 3 the image captured at the end of the experiment is shown figure (6.12). A triangular shape in the fluid is clear at this level when following the milky trail and is a signature of an $m = 3$ wave. The travelling wave is clear to see when looking at the image sequences. The milky clump of particles observed in figure (6.12) travels around with the wave making it easier to follow. This observed level is in the middle of the annulus in terms of the height, hence observations here are the furthest from the top

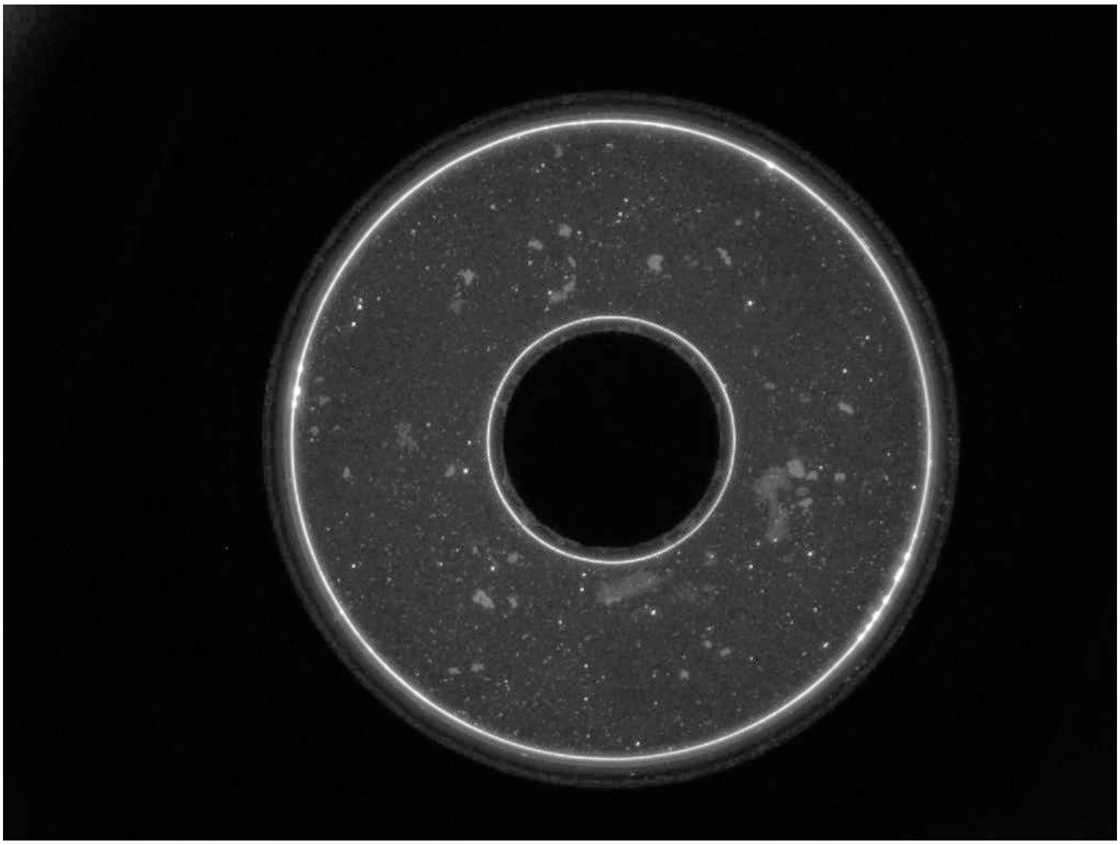


Figure 6.9: Image of the working fluid at the end of the 3hr run taken at level 2 for the 1 rad/s experiment

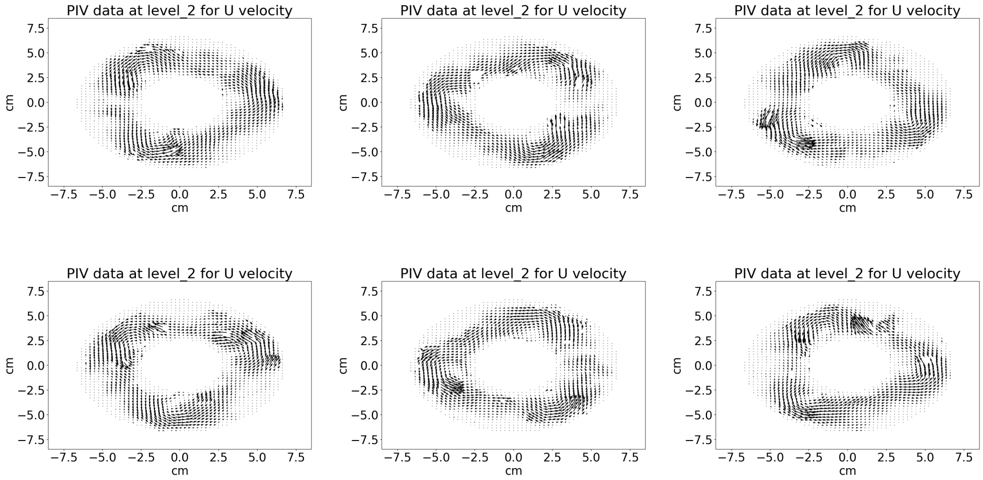


Figure 6.10: PIV result sequence towards the end of the experiment at level 2 for 1 rad/s showing U velocity

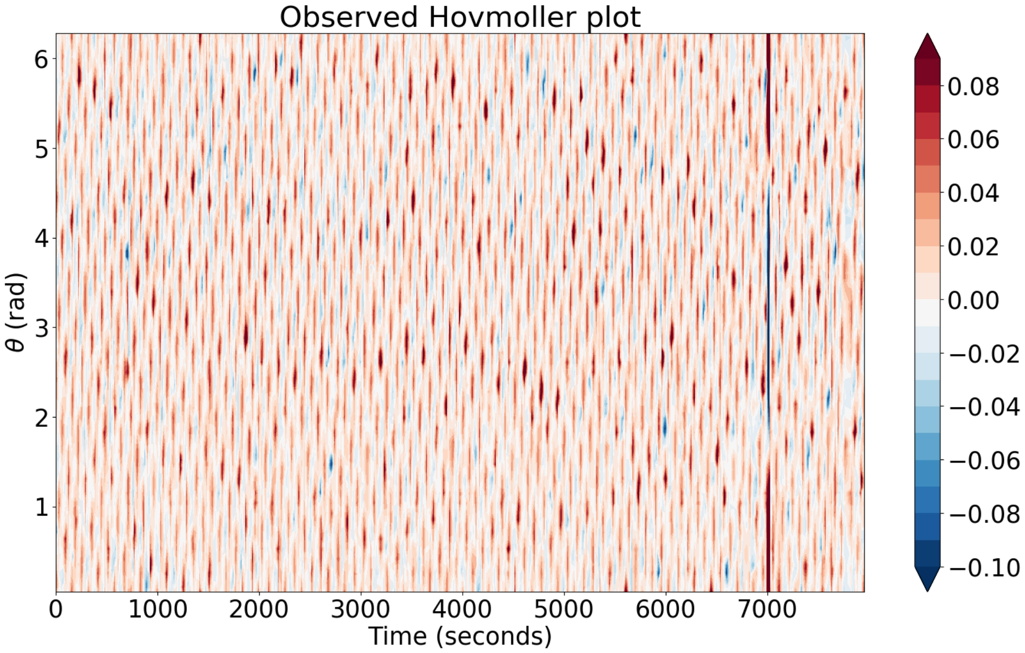


Figure 6.11: Hovmöller plot for level 2 at r=5.25cm for 1 rad/s showing the fluid between 1000 and 2000 seconds showing U velocity

and bottom boundary.

PIV data for the U velocity is shown in figure (6.13) and here the $m = 3$ wave can be observed. The shape of the wave can be matched to the milky structure in the image confirming what was discussed previously. The better structure for the wave could be attributed to its being the furthest away from both the top and bottom boundaries, hence giving the best conditions for the fluid to develop structures. Some of the data-poor areas observed in the previous level are not observed here, which seems to indicate that observation is not impeded much from the fluid above. The travelling wave is clear to see here, and the small vortices seen at lower levels are absent.

The Hovmöller plot showing the U velocity is shown in figure (6.14). The movement of the fluid is more clearly visible at this level, which matches what is observed in the PIV plot. There are still signs of amplitude vacillation as observed previously, with a mix of brighter and dimmer patches in the rotation.

6.4.2.4 Level 4

Moving to level 4, the image of the experiment is shown in figure (6.15) with the vector plot in figure (6.16). From the image, the structures of the fluid can be easily distinguished. Again the milky trail can be followed to work out that an $m = 3$ wave has formed. Following the sequence of the images, it is seen that the $m = 3$ wave is very stable and once formed does not break down and just drifts around the annulus. There

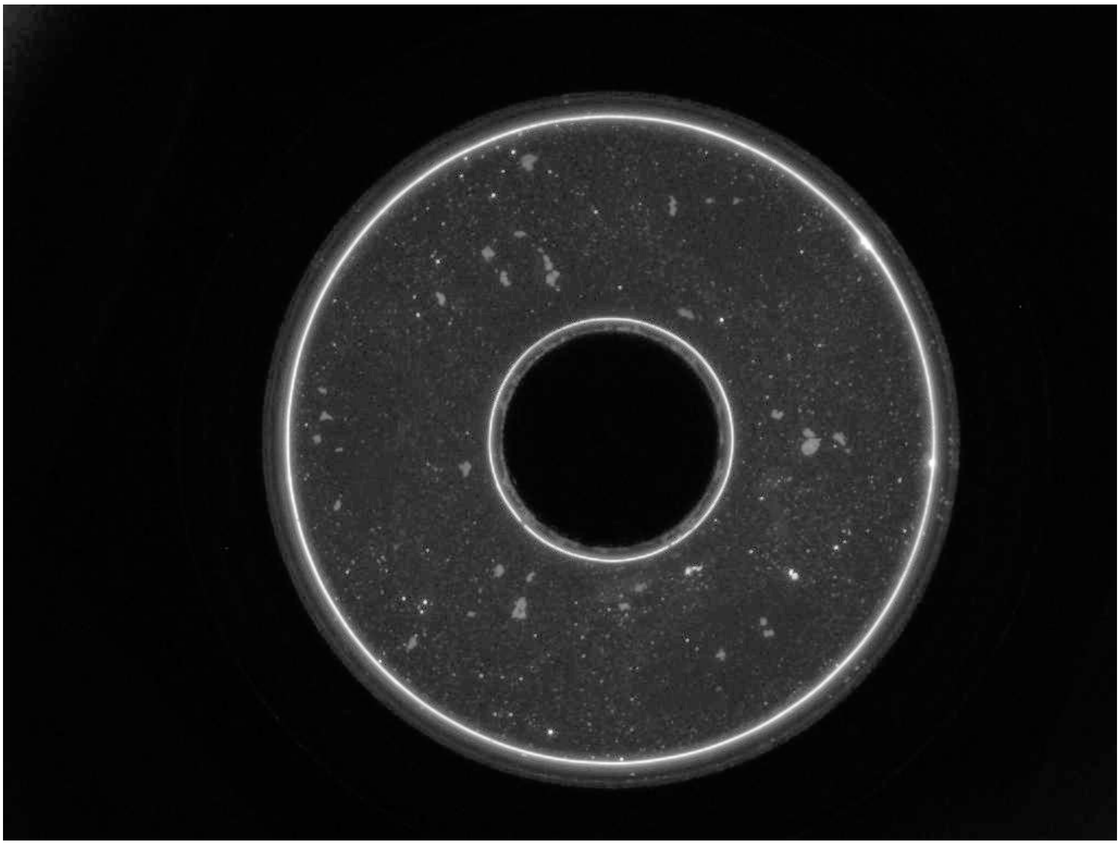


Figure 6.12: Image of the working fluid at the end of the 3hr run taken at level 3 for the 1 rad/s experiment

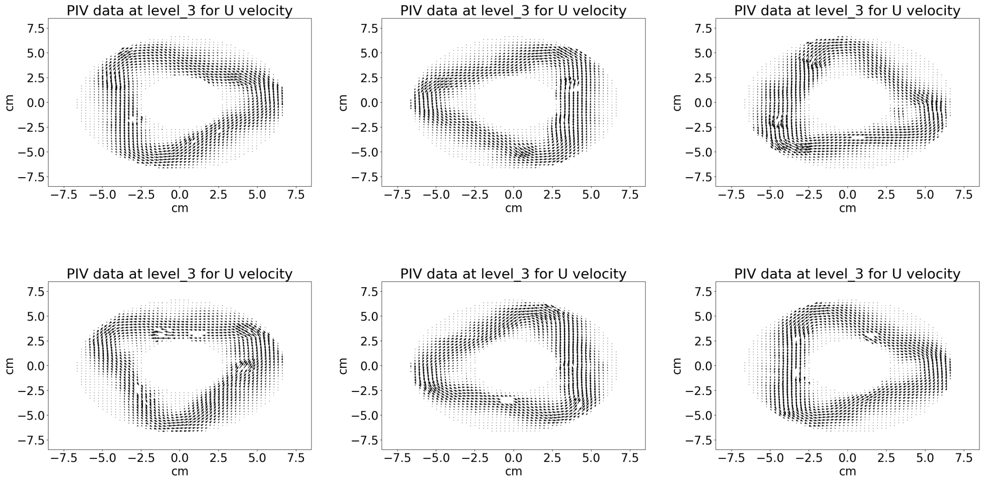


Figure 6.13: PIV result sequence towards the end of the experiment level 3 for 1 rad/s showing U velocity

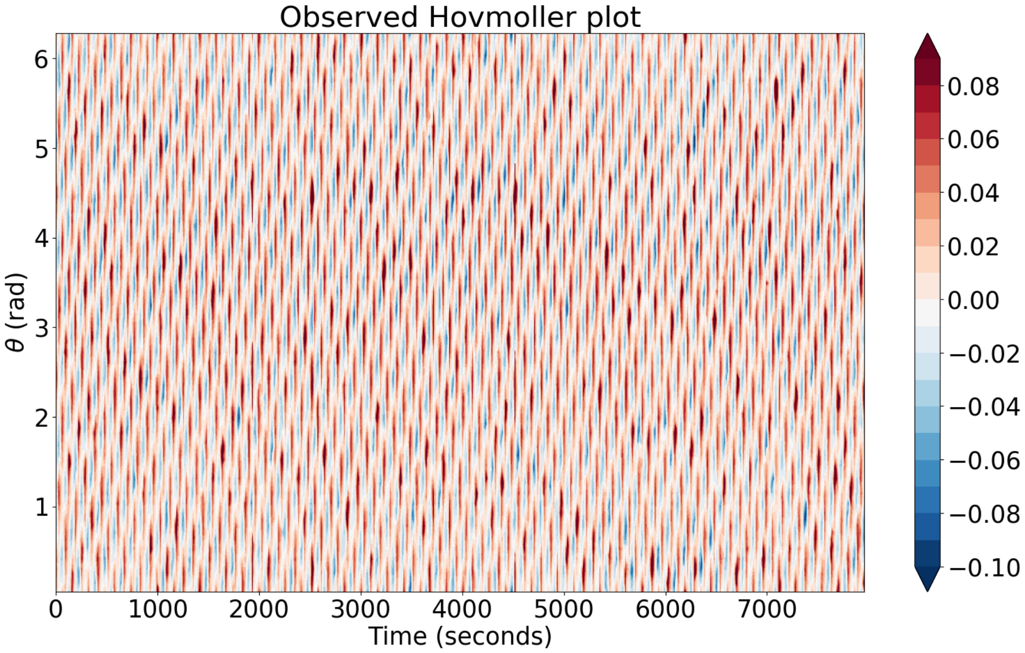


Figure 6.14: Hovmöller plot for level 3 at r=5.25cm for 1 rad/s showing the fluid between 1000 and 2000 seconds showing U velocity

is a good amount of particles observed in the wave structure that can be easily used to track the flow. There are also places at the outer wall where there is a distinct lack of particles. These could have formed as the wave has accumulated the particles as it has flowed around the annulus.

The PIV data showing U velocity is shown in figure (6.16) and matches the images and shows an $m = 3$ wave. At this height, the observations seem very clearly to show how the $m = 3$ wave travels around the annulus. The vortices that were present at lower levels are again absent at this level. Even the recirculation areas at the tips of the wave seem to be absent.

The Hovmöller plot is shown in figure (6.17), showing the U velocity movement of the fluid structure around the annulus. Similar behaviour is observed as was seen before in level 3 but the structures are more clearly seen travelling around the annulus. The amplitude vacillation is very clearly shown here. The structures are well observed because there is less obstruction at this height from the fluid above.

6.4.2.5 Level 5

The image at this high level is given in figure (6.18), with the vector plot given in figure (6.19). Clumps of particles can be easily tracked to see where the $m = 3$ wave is in the image. Looking at the image sequence there are areas where particles have become trapped. One of these areas is at the vortex created at the wave peaks near the inner wall

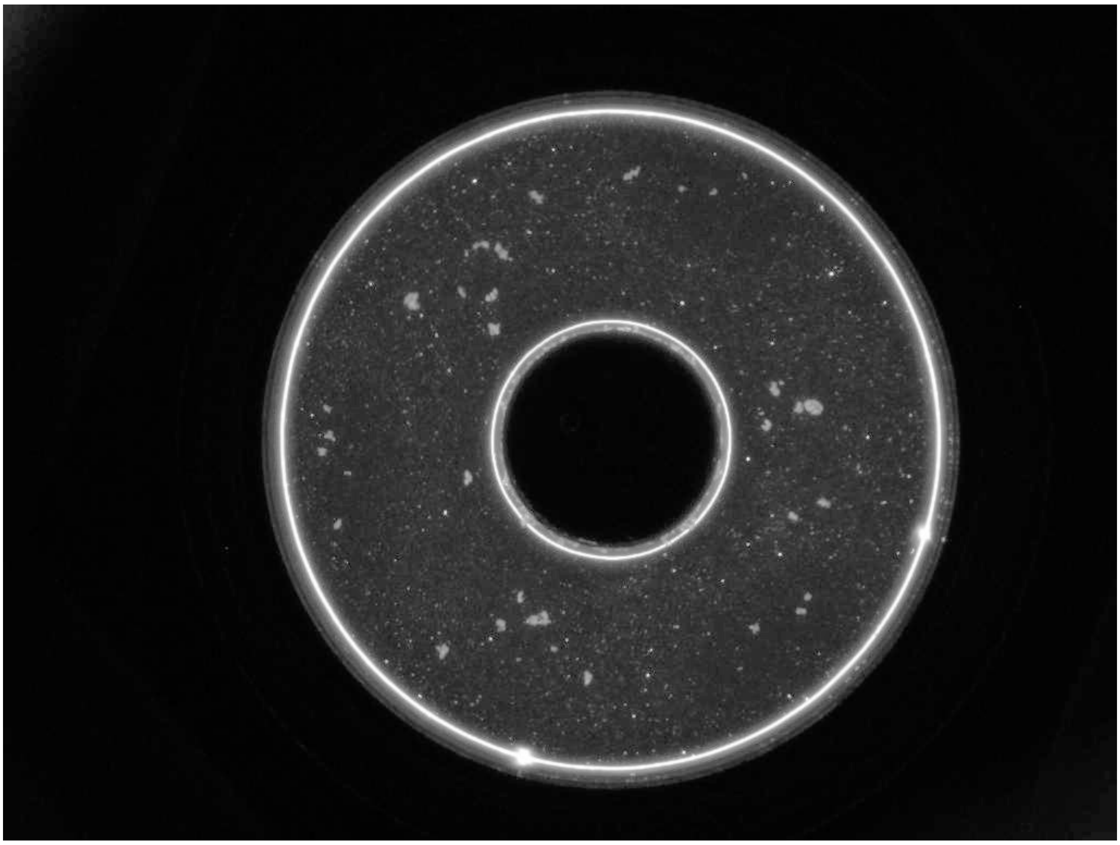


Figure 6.15: Image of the working fluid at the end of the 3hr run taken at level 4 for the 1 rad/s experiment

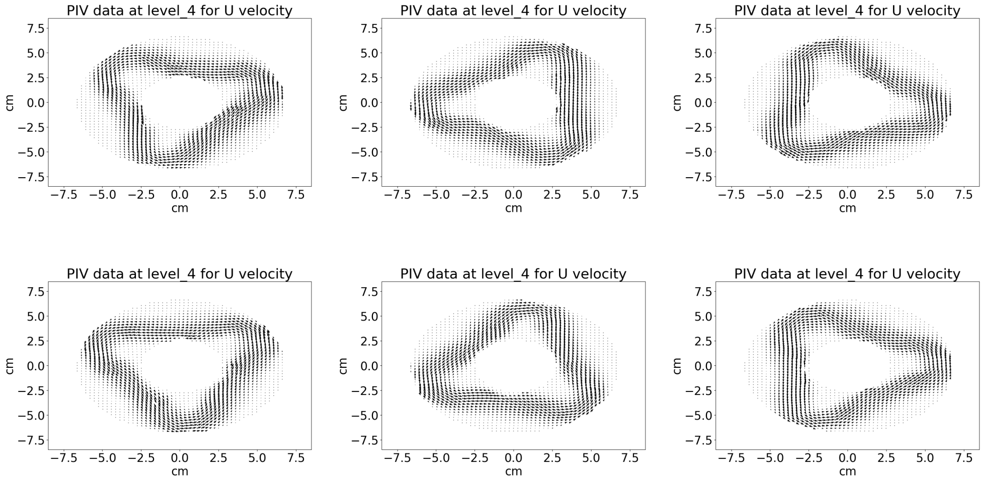


Figure 6.16: PIV result sequence towards the end of the experiment at level 4 for 1 rad/s showing U velocity

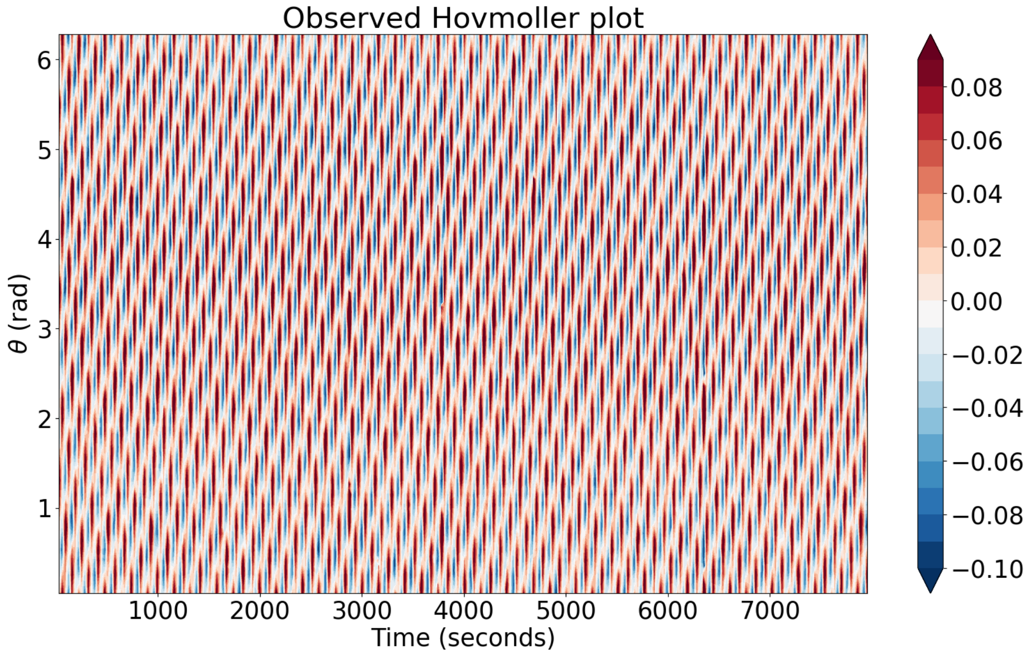


Figure 6.17: Hovmöller plot for level 4 at r=5.25cm for 1 rad/s showing the fluid between 1000 and 2000 seconds showing U velocity

and another being between wave peak areas on the outer wall of the annulus. As the wave has accumulated more and more particles over time, these areas of low velocity have allowed some particles to clump up and get trapped. These clumped particles then drift around the annulus with the wave.

In the PIV plot in figure (6.19) an $m = 3$ wave is observed in the vector plot. As with the previous images the wave is very stable and flows around the annulus without breaking down. A lack of a rigid boundary layer at the top might have also contributed to the stability of the waves.

The Hovmöller plot in figure (6.20) shows similar behaviour to level 4. The structures are very distinct and travel across the annulus as before. There is again evidence of distinct changes seen in the amplitudes of the structures, amplitude vacillation.

6.4.2.6 Temperature data

Data from the temperature sensors at the outer wall are presented in figure (6.21). A clear oscillation in temperature can be observed for each sensor. Temperature sensor 4 and 5 are placed towards the bottom of the annulus and show a very different temperature at the bottom of the graph as seen in the black and blue line. This is probably due to the recirculation of the fluid. Cold fluid is descending near the inner boundary and then moves outward near the bottom boundary, so the temperatures there are systematically lower. At all levels the temperature oscillations are all fairly closely in phase. At

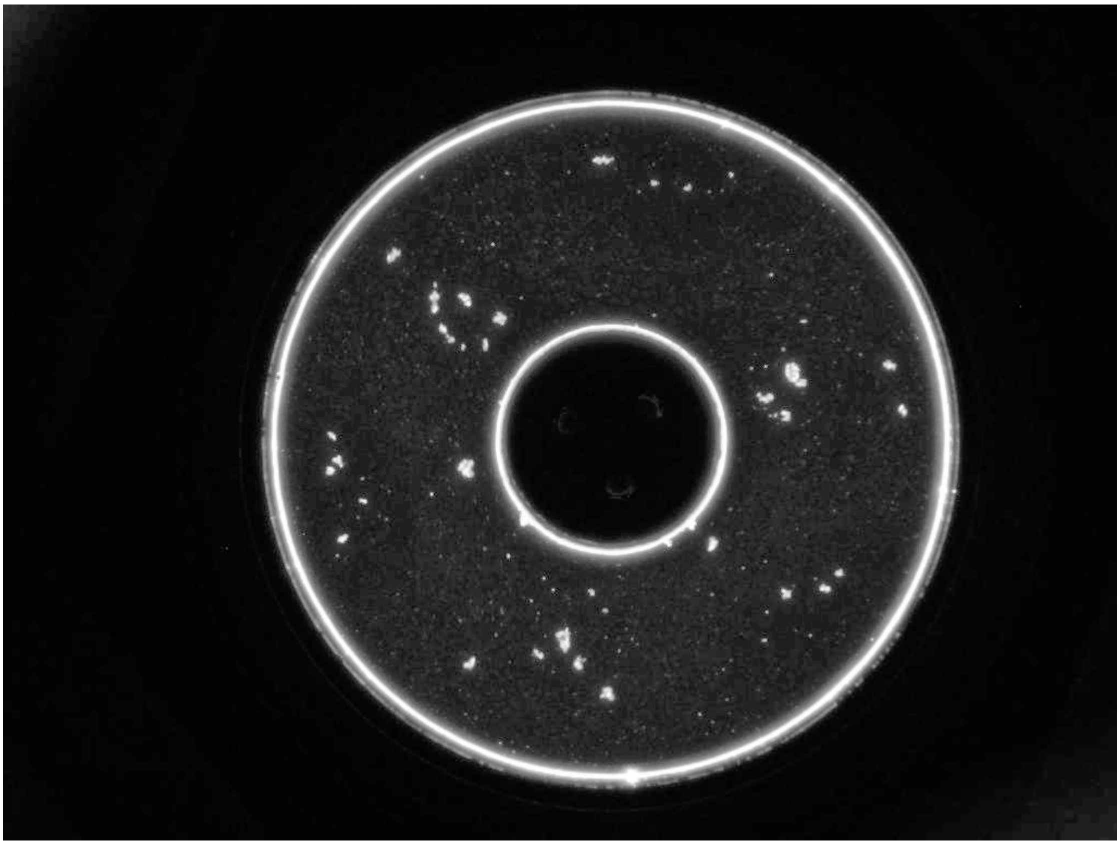


Figure 6.18: Image of the working fluid at the end of the 3hr run taken at level 5 for the 1 rad/s experiment

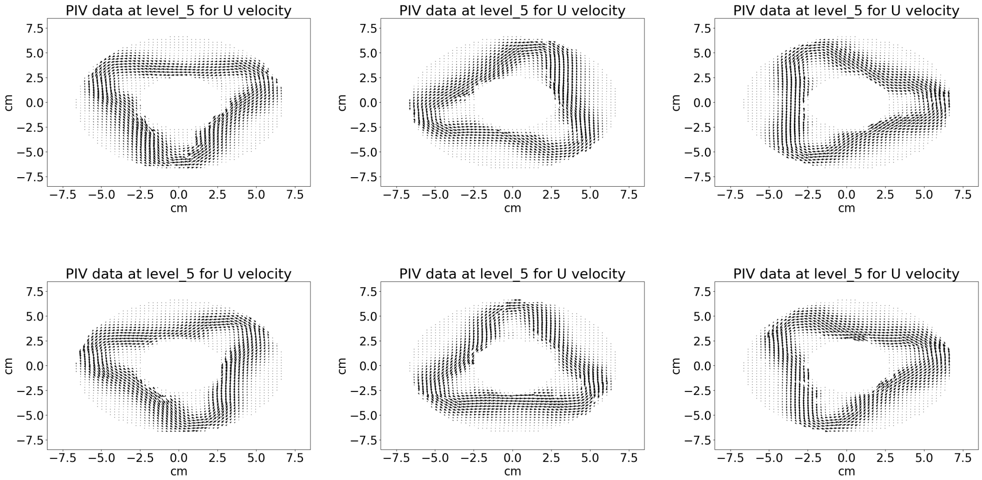


Figure 6.19: PIV result sequence towards the end of the experiment at level 5 for 1 rad/s showing U velocity

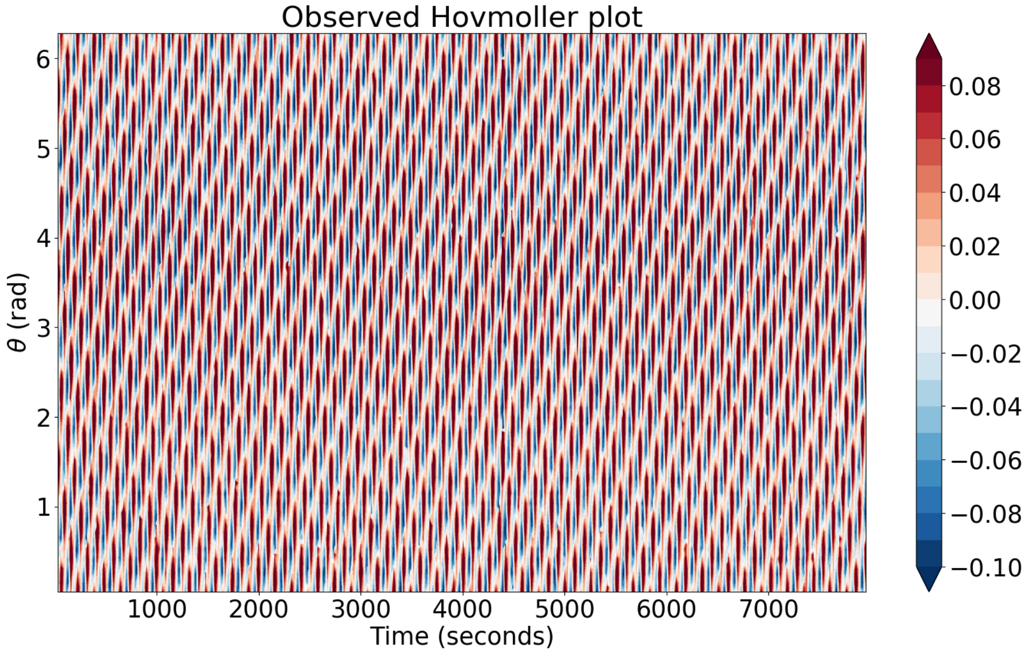


Figure 6.20: Hovmöller plot for level 5 at r=5.25cm for 1 rad/s showing the fluid between 1000 and 2000 seconds showing U velocity

the upper levels they show broadly similar temperatures and rotate at the same rate, but some of the upper sensors show slightly different temperatures indicating a temperature stratification of the liquid with depth.

6.4.2.7 Summary for rotation at 1 rad/s

An experiment was performed at 1 rad/s and an $m = 3$ wave was observed. At lower levels, the picture was not altogether clear, though the Hovmöller plots were surprisingly well-structured.. The $m = 3$ wave can be clearly observed in level 3 onwards, with the wave being very stable once it formed in the experiment. PIV data analysis was able to track the main structure of the flow but had issues tracking the localised behaviour. Hovmöller plots gave clearer results when higher levels were observed and there were clear signs of amplitude vacillation at all levels. The temperature plots from sensors on the outer wall show a steady oscillation of waves.

6.4.3 2.5 rad/s

The second experiment was done at a rotation rate of 2.5 rad/s and ran for around two and a half hours. Apart from the increase in rotation rate no other settings were changed. This rotation rate was chosen because from simulation studies this rotation regime seems

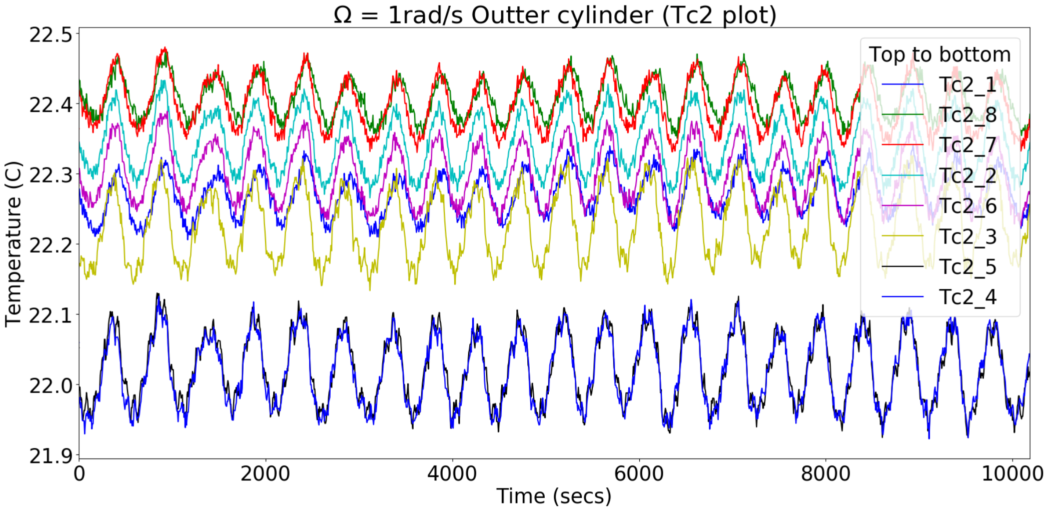


Figure 6.21: Temperature at the outer wall at 1 rad/s. The position for thermocouples on the outer cylinder is (from top to bottom): 1, 8, 7, 2, 6, 3, 5, 4.

to lie between the turbulent flow observed at higher rotation rates seen at 3 rad/s and the more steady waves seen at lower rotation rates such as 2 rad/s. Hence this regime would give some unique behaviour which can be analysed creating another test for the PIV software. Details of what was observed at each level are given below.

6.4.3.1 Level 1 and Level 2

As discussed in the results for 1 rad/s, observation of data from lower levels are not accurate due to interactions with the lower boundary and fluid from higher levels obscuring image capture. With the increase in rotation rate, these problems persist.

Figures (6.22) and (6.23) show the sequence of PIV data obtained at level 1 and level 2 respectively. Both show a very messy flow, where not much about the structure of the working fluid can observed. There are a lot of data poor regions.

6.4.3.2 Level 3

Level 3 was the first level at 1 rad/s which showed a coherent structure in the rotating fluid. The image of the experiment at the end of the study is given in figure (6.24). There are four clumps of particles in four corners of the flow that travel with the fluid. These clumps are from the higher level as they seem to block the view of particles that pass the region.

Following other bright particles around the annulus does not resolve whether there is a wave at this level. There is a flow of the fluid, but the structure is not as clean and simple as seen at 1 rad/s. This was expected with the higher rotation rates but it was hoped that the system would be capable of tracking these waves. The four particle

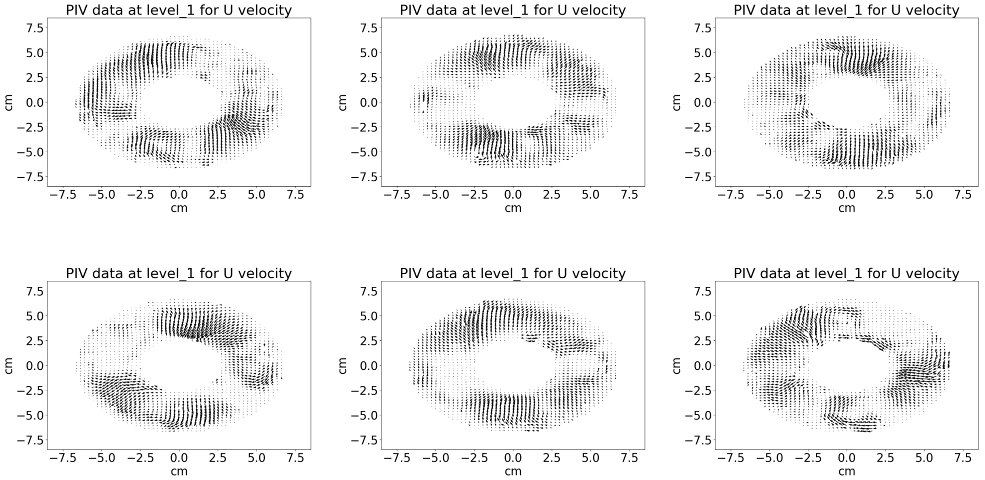


Figure 6.22: PIV result sequence towards the end of the experiment at level 1 for 2.5 rad/s showing U velocity

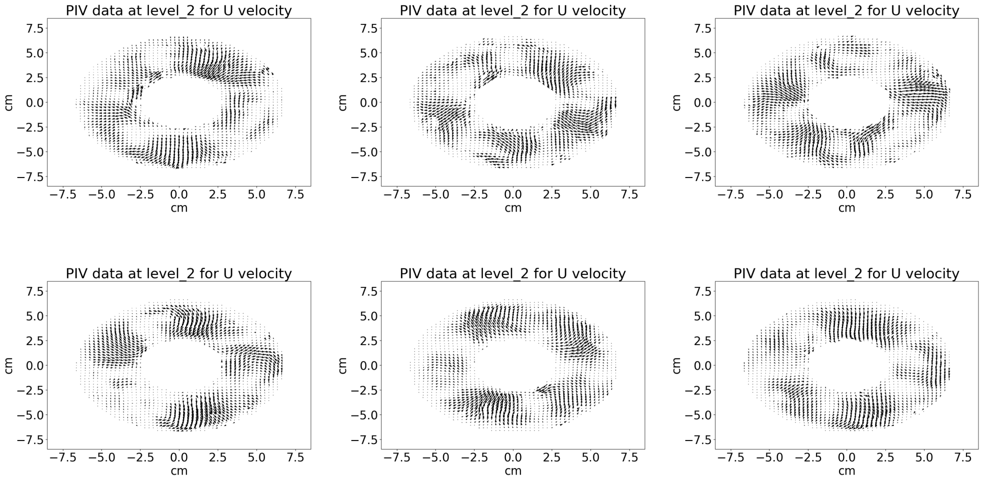


Figure 6.23: PIV result sequence towards the end of the experiment at level 2 for 1 rad/s showing U velocity

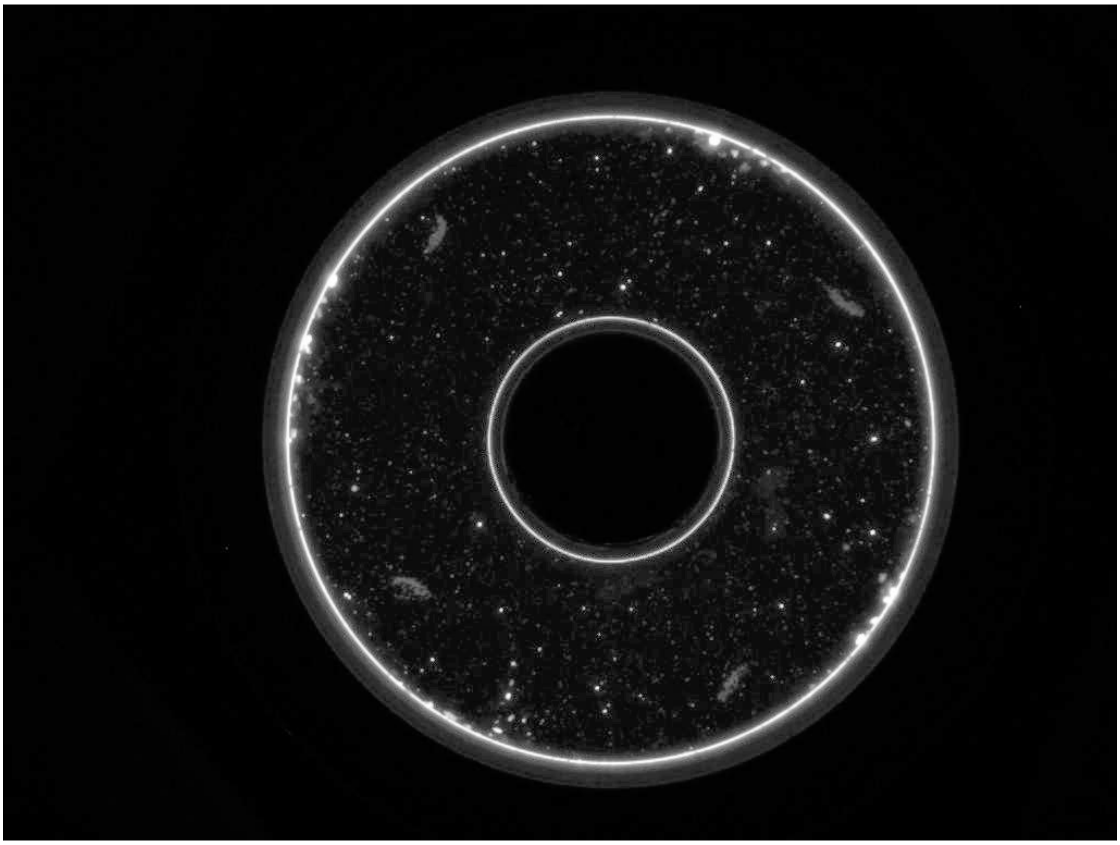


Figure 6.24: Image of the working fluid at the end of the 3hr run taken at level 3 for the 2.5 rad/s experiment

clumps do point towards an $m = 4$ wave but as they are from a higher level they cannot be used to confirm anything for this level using the image.

The PIV vector plot sequence for U velocity is shown in figure (6.25). The m=4 wave is starting to become clear in the vector plots. The four clumps of particles that were seen in the experimental image line up with the edges of the rotating wave structure in the PIV plot. This seems to indicate that the wave at the higher levels is also at the same place at this time. There are some data poor regions for the wave but structures of the flow can be deduced.

The Hovmöller plot for the experiment is shown in figure (6.26) and shows a very clear $m = 4$ wave structure travelling around the annulus. Even though the plot is noisy the structures of the flow seems very steady. There are some changes in the amplitude, but due to the noise, no confirmation can be made about amplitude vacillation, but the strong amplitude vacillation seen at 1 rad/s seems to be absent.

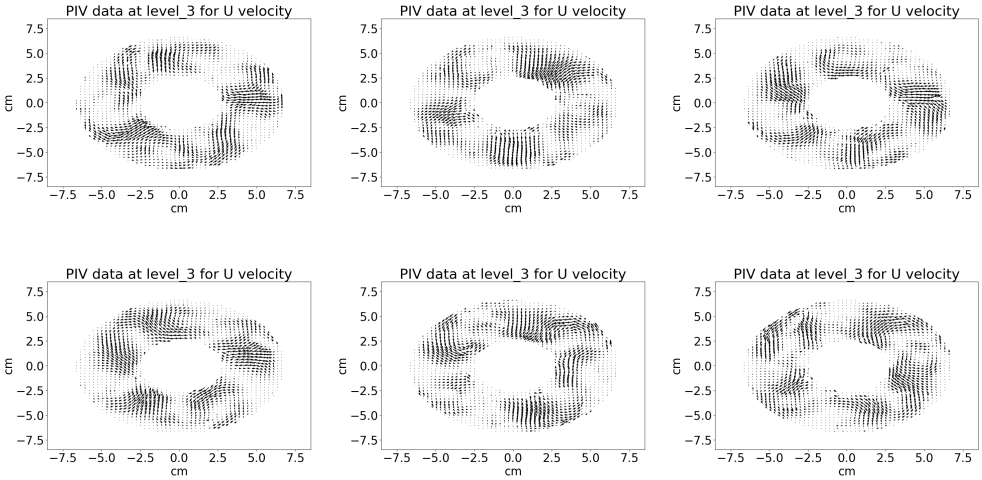


Figure 6.25: PIV result sequence towards the end of the experiment at level 3 for 2.5 rad/s showing U velocity

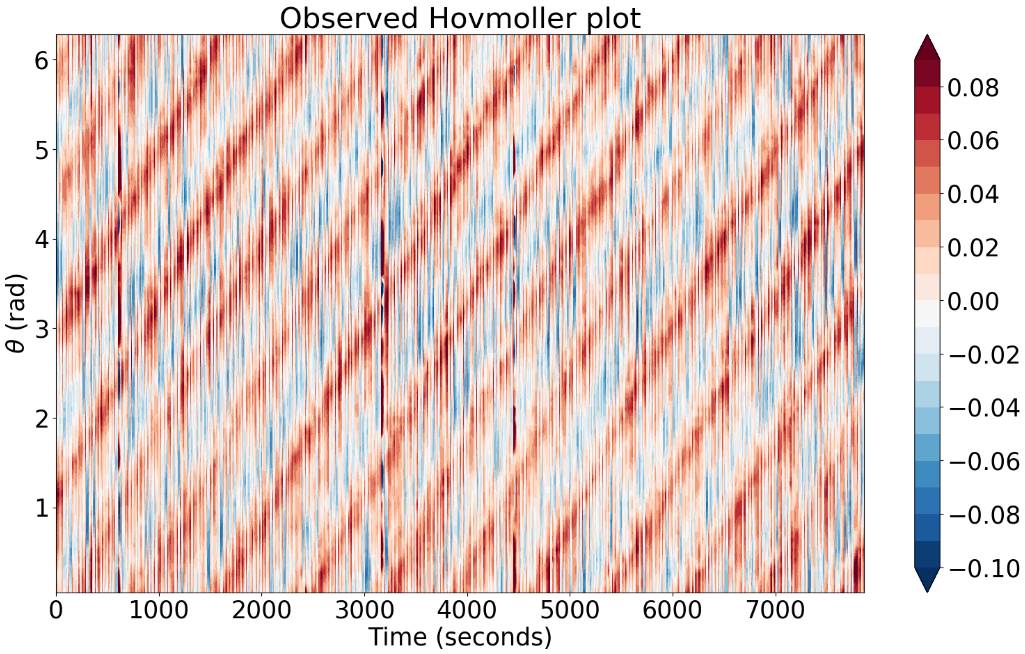


Figure 6.26: Hovmöller plot for level 3 at r=5.25cm for 2.5 rad/s for U velocity

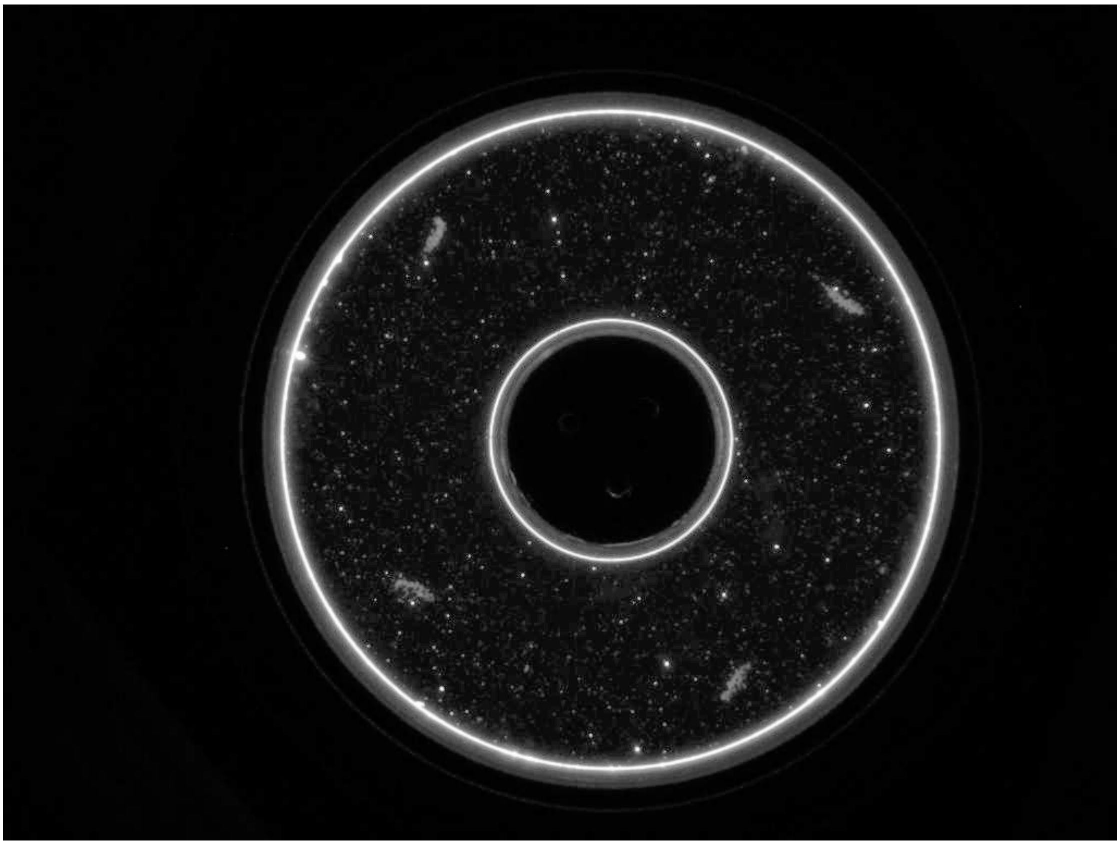


Figure 6.27: Image of the working fluid at the end of the 3hr run taken at level 4 for the 2.5 rad/s experiment

6.4.3.3 Level 4

The image taken at level 4 at the end of the experiment is shown in figure (6.27). The four clumps of particles are still visible and a few bright particles seem to pass over them indicating that these clumps are from a lower level. This means that the clumps are between level 3 and level 4. Looking at the sequence of images there is an $m = 4$ wave travelling around the annulus. It can also be seen that the flow is strong and the wave touches the inner and outer walls on its way around the annulus. The structures that are formed drift around the annulus very slowly compared to the structures observed for 1 rad/s.

The vector plot for the flow showing U velocity is plotted in figure (6.28). The $m = 4$ wave is very clear now in the vector plot with the wave being observed throughout the experiment. This results complements what was observed in the experimental images indicating the PIV analysis is on the correct path.

The strong flows around the inner and outer walls can be observed in the PIV plot as was discussed with the experimental observations. There are a lot of data poor regions between the structures on the outer wall. This seems to be working fluid that is stuck between the main flow structures and hence cannot move by itself and just follows the

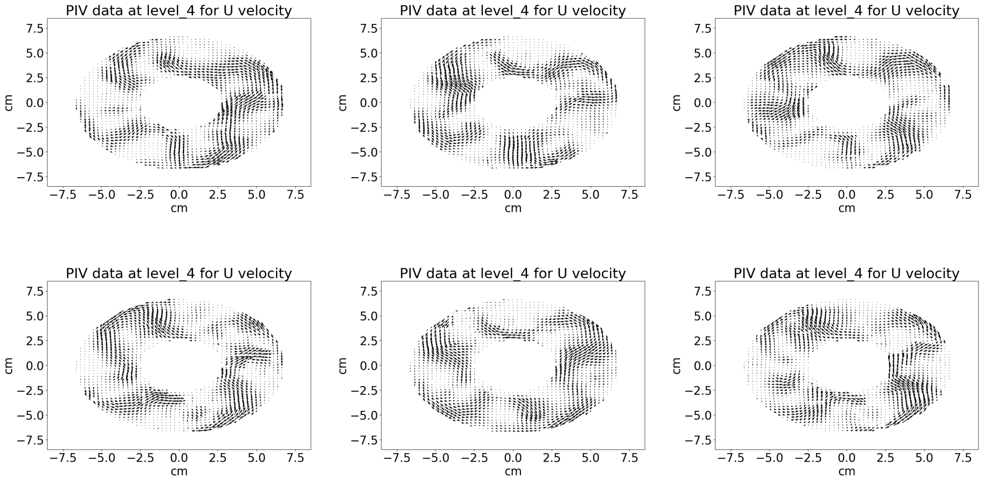


Figure 6.28: PIV result sequence towards the end of the experiment at level 4 for 2.5 rad/s showing U velocity

drift of the wave structures. The particles in this region might not move much leading to the data poor results.

The Hovmöller plot for U velocity is shown in figure (6.29) and shows the $m = 4$ wave travelling around the annulus. As before the plot is noisy and some variations are being observed in the amplitudes of the waves, but there is no effective way to say if there is any amplitude vacillation. There a few structures that become weaker towards the end of the run, the reasons for which are unknown.

6.4.3.4 Level 5

The image taken at level 5 is shown in figure (6.30). The four clumps of particles observed in other levels are still present here and follow the wave around the annulus. An $m = 4$ wave can be observed travelling around when looking at the image sequences. The clumps seem to be part of the structure of the wave as it travels around the annulus.

The U velocity vector plot for the experiment is shown in figure (6.31). Here the plot clearly shows an $m = 4$ wave travelling around the annulus confirming what was observed in the images. The structures of the flow are clear and the route the flow takes from the outer wall to the inner wall around the annulus can be observed. There are areas of poor observations around the flow which might be due to a lack of particles that can be tracked. It is interesting to see that the shape of the structure head is stretched. This seems to be a result of the faster flow pushing a lot of the working fluid towards the outer wall. Some of the behaviour might also be due to there not being a rigid top boundary.

The Hovmöller plot for U velocity is shown in figure (6.32). As before the structures shows an $m = 4$ wave and matches the structure observed in the vector plot. It is

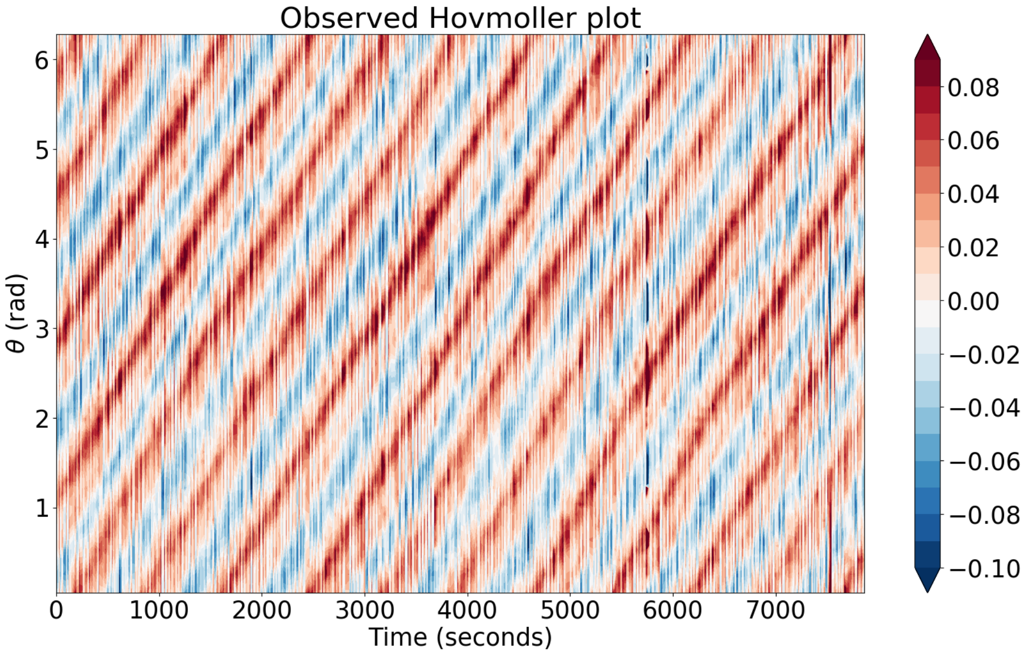


Figure 6.29: Hovmöller plot for level 4 at r=5.25cm for 2.5 rad/s for U velocity

interesting to note that at this rotation rate the structure at the highest level is still stable and not affected by the experiment not having a top lid.

6.4.3.5 Temperature

Data from the temperature sensors at the outer wall are presented in figure (6.33). As before a clear oscillation in temperature can be observed for each sensor. The data from the bottom sensors 4 and 5 are much closer to the rest of the sensors than they were at 1 rad/s. Data from sensor 8 are higher than the rest, this is the second sensor on the annulus from the top. The reason for this separation is unclear.

Another interesting structure observed in the rotation is a region where oscillation seems to stop (around 3000 seconds mark). Looking through the image sequence nothing of note is observed so the reason for this in the temperature sensor data is unclear. Nothing can be observed in the Hovmöller plot either. It could just be a glitch in the temperature sensor. This behaviour was observed previously when the image capture system crashed in earlier tests.

6.4.3.6 Summary for rotation at 2.5 rad/s

With the faster rotation rate the wavenumber has increased to an $m = 4$. But there is also a lot more complex flows due to the extra energy being pumped into the system. This extra energy pushes the particles around a lot with clumps of particles being observed

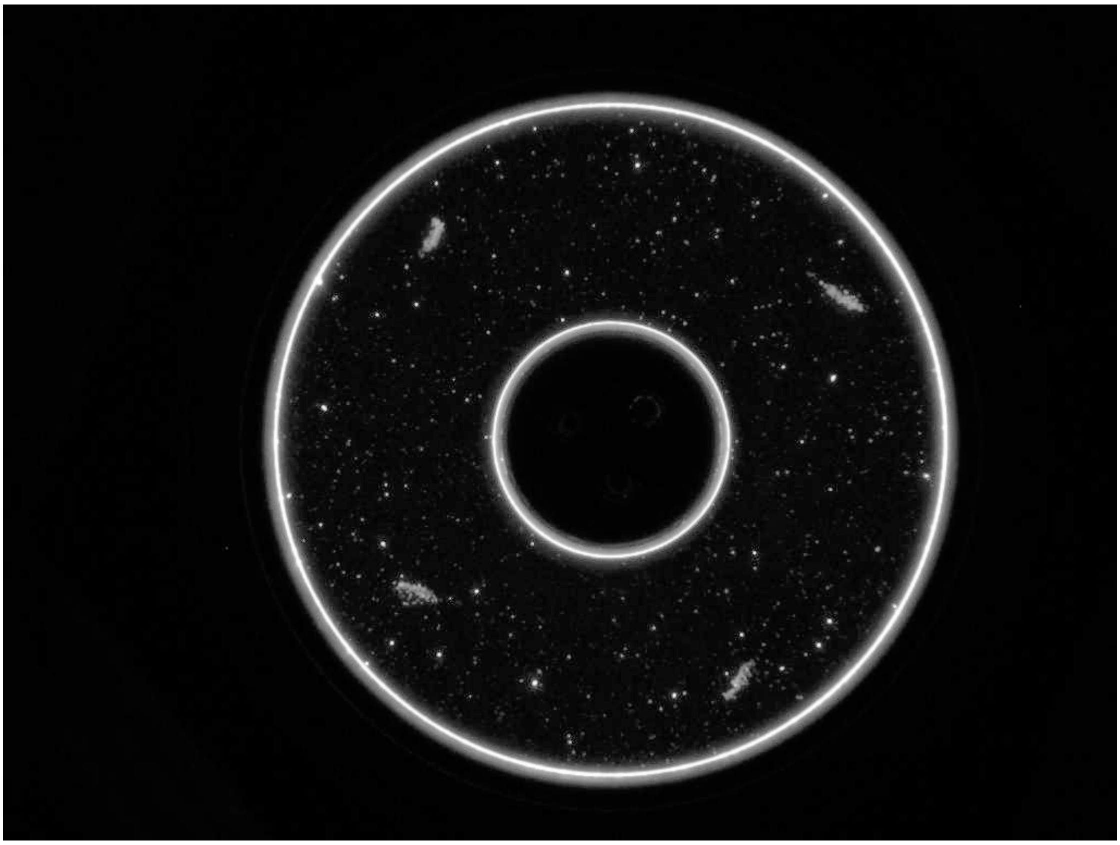


Figure 6.30: Image of the working fluid at the end of the 3hr run taken at level 5 for the 2.5 rad/s experiment

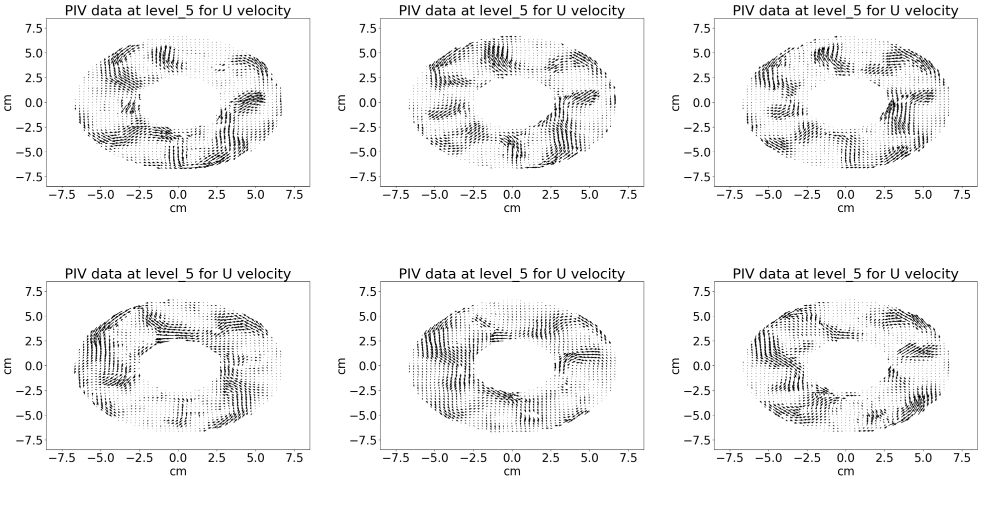


Figure 6.31: PIV result sequence towards the end of the experiment at level 5 for 2.5 rad/s showing U velocity

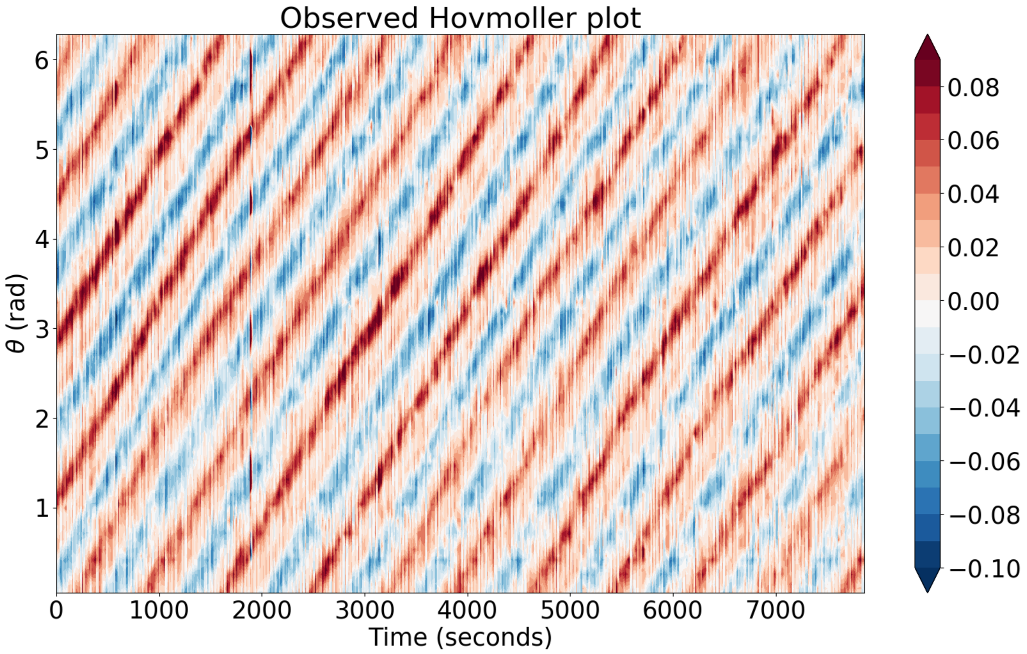


Figure 6.32: Hovmöller plot for level 5 at r=5.25cm for 2.5 rad/s

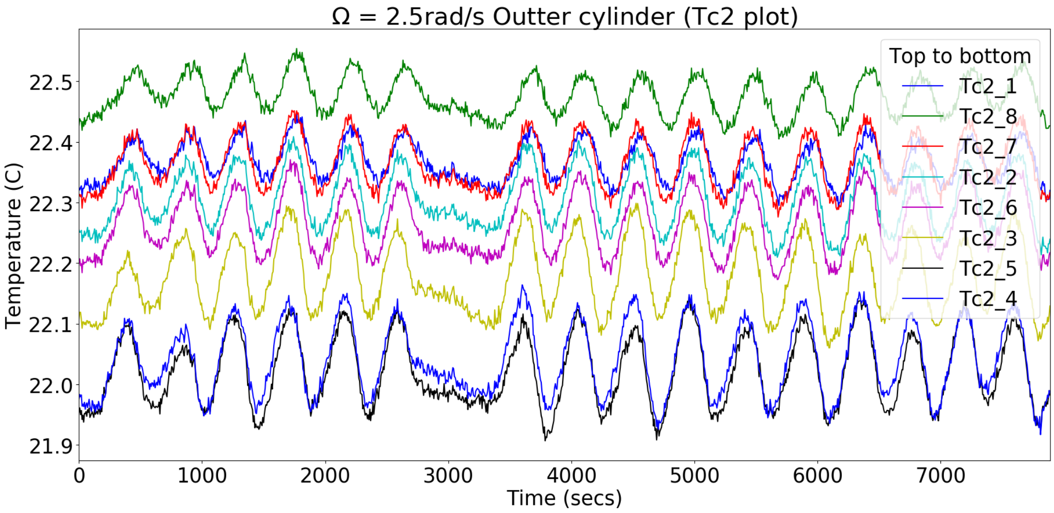


Figure 6.33: Temperature at the outer wall at 2.5 rad/s. The position for thermocouples on the outer cylinder is (from top to bottom): 1, 8, 7, 2, 6, 3, 5, 4.

for every level. At the lower levels, it is very difficult to observe the flow in the images but at higher levels, it becomes clear that flow is indeed an $m = 4$ wave.

The PIV plots confirm a $m = 4$ wave is going around the annulus. At lower levels, a lot of messy structures are observed with a lot of areas where there seems to be a lack of coherent observations. Although better observations are gathered at higher levels, areas of poor observation never go away. Higher levels clearly show the $m = 4$ waves travelling around the annulus. The flow seems to be a bit stretched on the outer wall which might be due to the faster rotation rate.

Hovmöller plots even for low levels confirm the $m = 4$ wave. As only the middle of the annulus is sampled for data points for the plot these plots indicate that there is a flow of fluid with wave like behaviour in the middle of the annulus. As the observations are taken at higher levels the wave becomes more and more clear. The temperature plot also indicates a stable wave travelling around the annulus, although there seems to be a region where the temperature sensor seems to have not logged data properly the reason for which is not known.

6.4.4 3 rad/s

The final experiment was done at the rotation rate of 3 rad/s. At this high rotation rate taking accurate observations is difficult as turbulent structures make it harder to track movement of the flow. After observing at 2.5 rad/s the number of particles seemed low, so more seeding particles were added for this higher rotation rate for better tracking. Some of the clumpings of the particles observed in images are due to this. The higher rotation rate also pushes the particles towards the inner and outer walls of the annulus causing additional clumping.

It was believed that the faster rotation rate will cause significant deformity at the top boundary of the fluid so the lid for the experiment as added back to try and stop this. Also, data acquisition was started as soon as the annulus was turned on and the experiment was run for around two and a half hours. Details of what is observed at each level are given below.

6.4.4.1 Level 1 and Level 2

With the higher rotation rate, it is harder to observe accurate data at the lower levels. Figure (6.34) and (6.35) show the sequence of PIV data obtained at level 1 and level 2. There is a lack of structures seen in the sequences of the PIV plots, this is especially true for level 1. There are glimpses of a wave in the level 2 plots but overall it is still hard to distinguish structures. This seems to indicate that the software is having a hard time when it comes to tracking particles for these levels.

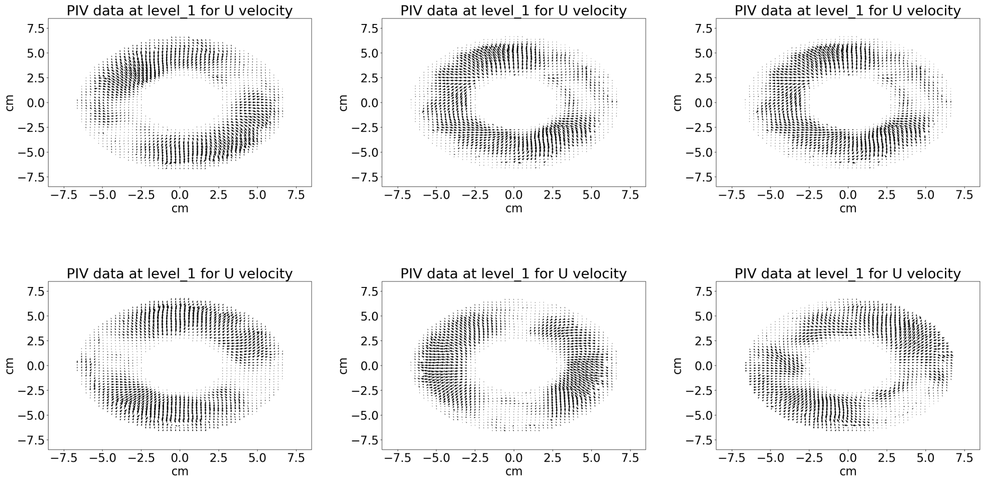


Figure 6.34: PIV result sequence towards the end of the experiment at level 1 for 3 rad/s for U velocity

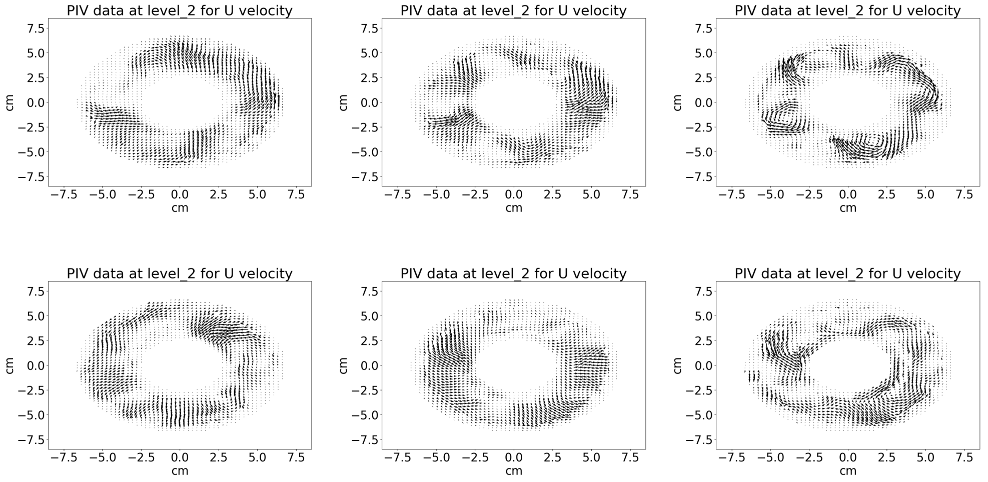


Figure 6.35: PIV result sequence towards the end of the experiment at level 2 for 3 rad/s for U velocity

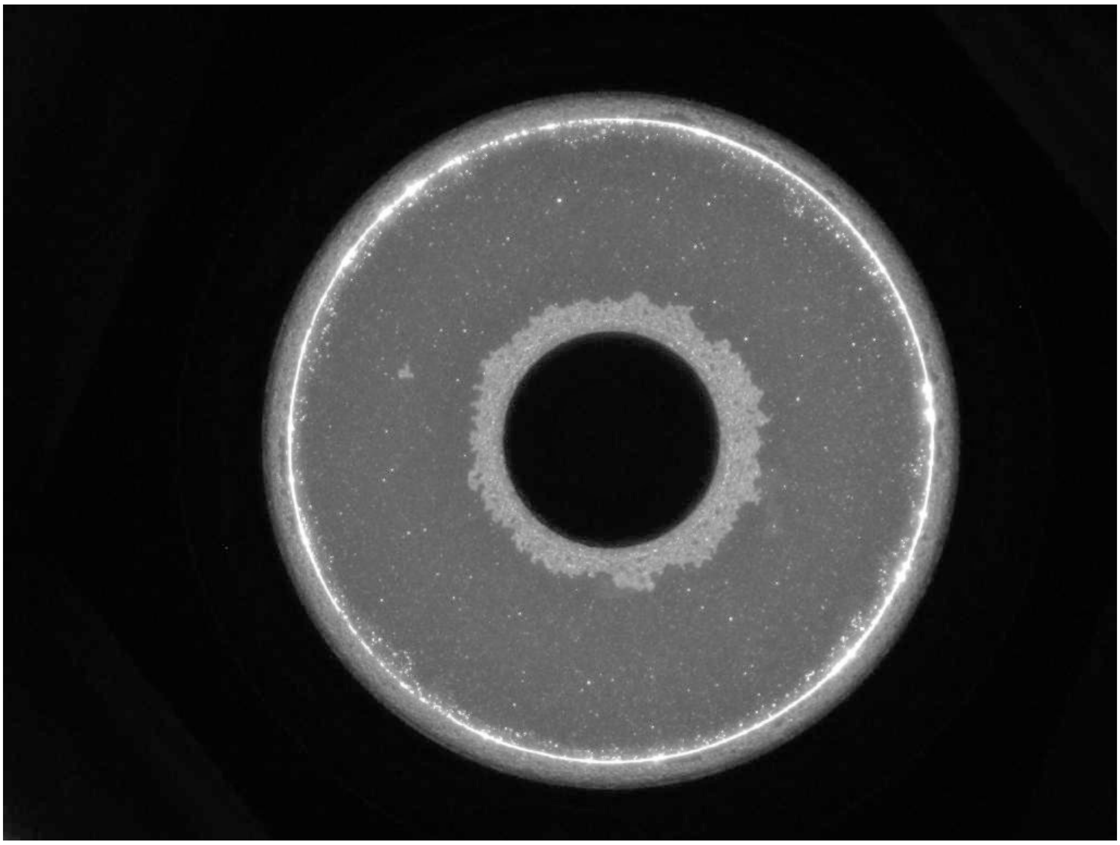


Figure 6.36: Image of the working fluid at the end of the 3hr run taken at level 3 for the 3 rad/s experiment

6.4.4.2 Level 3

At level 3 the state of the experiment at the end of the run is shown in figure (6.36). There is an accumulation of particles at the inner cylinder which has built up over time. The particles in the middle do show some movement but it is hard to asses if structures or a wave are travelling around the annulus. Some bright particles still appear and disappear but overall there is consistency in the bright particles moving around the annulus.

The vector plots for U velocity are shown in figure (6.37). The movement of the fluid is confirmed by the vector plot with the presence of structures in the flow. This flow seems to be quite chaotic with vortices and lots of data poor areas causing issues with observation. At various points, it seems the flow is being pushed towards the inner and outer walls. It does seem like the clump at the inner wall might be causing issues when tracking the flow as the PIV plot seems to be showing flows at the inner wall where they should not be.

The Hovmöller plot for U velocity is shown in figure (6.38). There is a lot of noise that is observed in the plot making it hard to see clear structures. This corroborates what was observed in the PIV plots. Vague wave like structures can be seen but it is very hard to observe them.

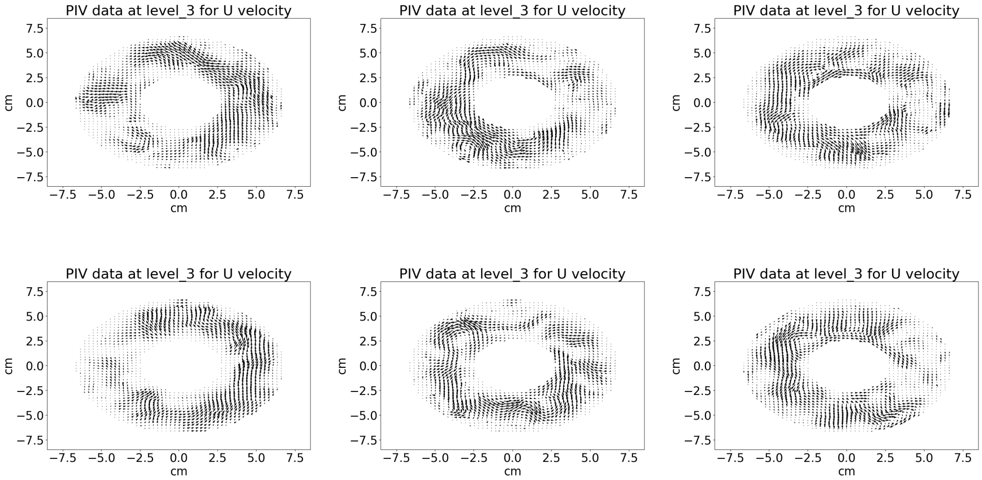


Figure 6.37: PIV result sequence towards the end of the experiment at level 3 for 3 rad/s showing U velocity

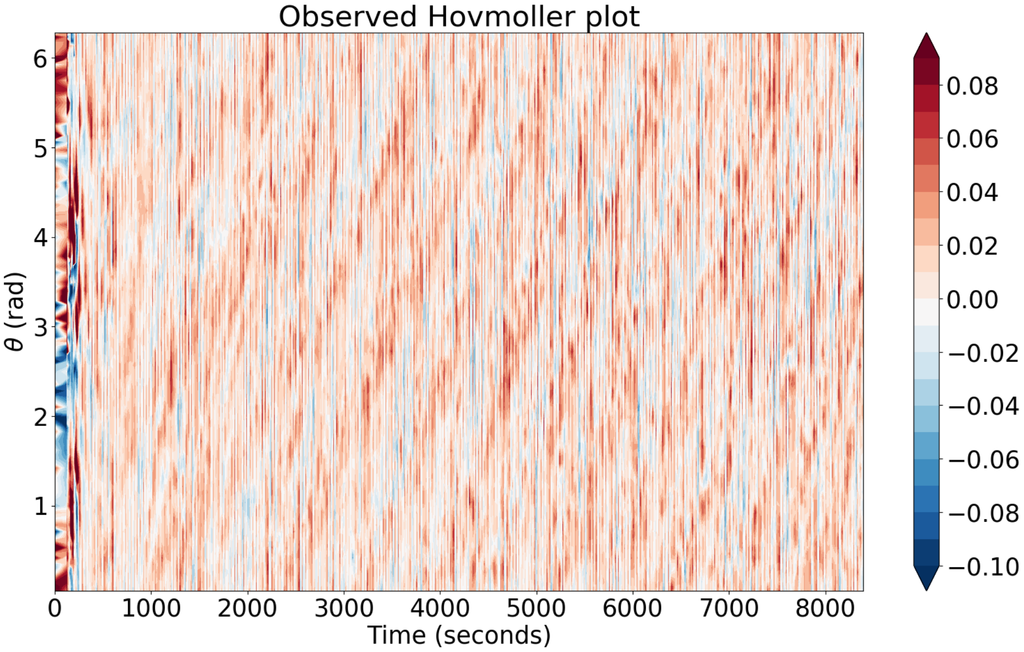


Figure 6.38: Hovmöller plot for level 3 at r=5.25cm for 3 rad/s for U velocity

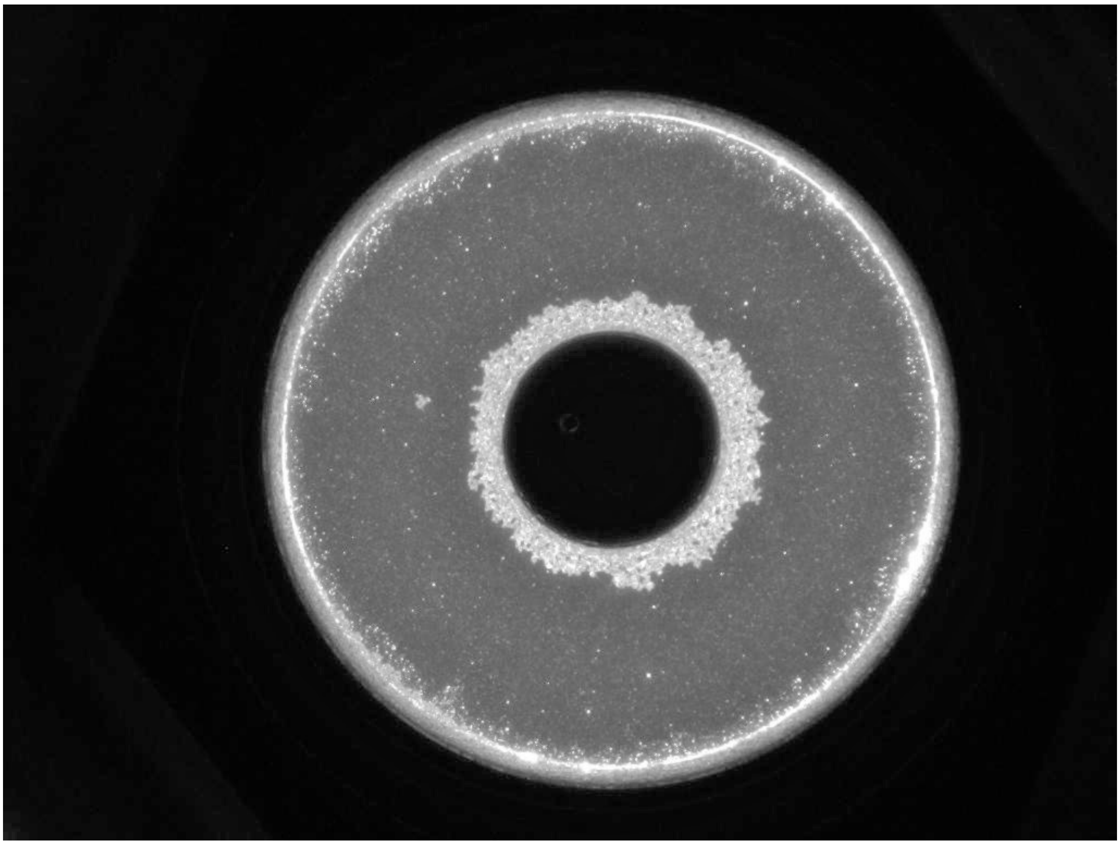


Figure 6.39: Image of the working fluid at the end of the 3hr run taken at level 4 for the 3 rad/s experiment

6.4.4.3 Level 4

At level 4 the image of the flow is shown in figure (6.39). The clump of particles at the inner wall is still present showing the same behaviour as discussed previously. Bright particles are also observed around the outer wall but they don't clump up as observed at the inner wall. Looking at the particles in the middle of the annulus a clear flow can be observed when looking a images sequences and a wave like structure can be seen going around.

The vector plot for U velocity is shown in figure (6.40). An $m = 4$ wave can be observed and seems stable for the whole of the run. This is a very drastic change from what was observed at the lower levels where the flow was hard to determine. There still are data poor areas between the waves as seen in the other runs as well. These areas might be populated by slow/trapped particles which lie between the arms of the waves. The particle clump in the middle hasn't seemed to have had an effect on the PIV analysis at this level.

The Hovmöller plot for U velocity is shown in figure (6.41) and shows an $m = 4$ wave which corresponds to what is observed in the vector plots. The structures here are very coherent from the start, and no breakdown of the structures are observed. It seems

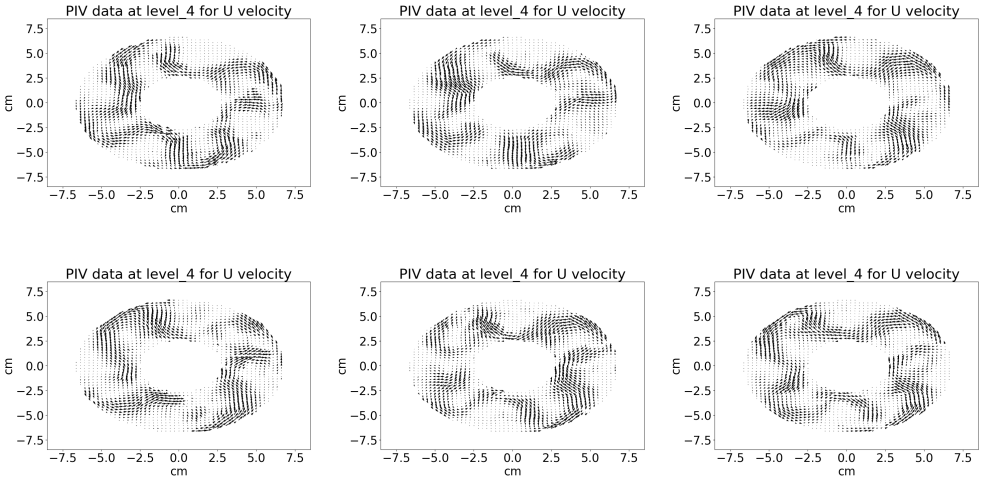


Figure 6.40: PIV result sequence towards the end of the experiment at level 4 for 3 rad/s for U velocity

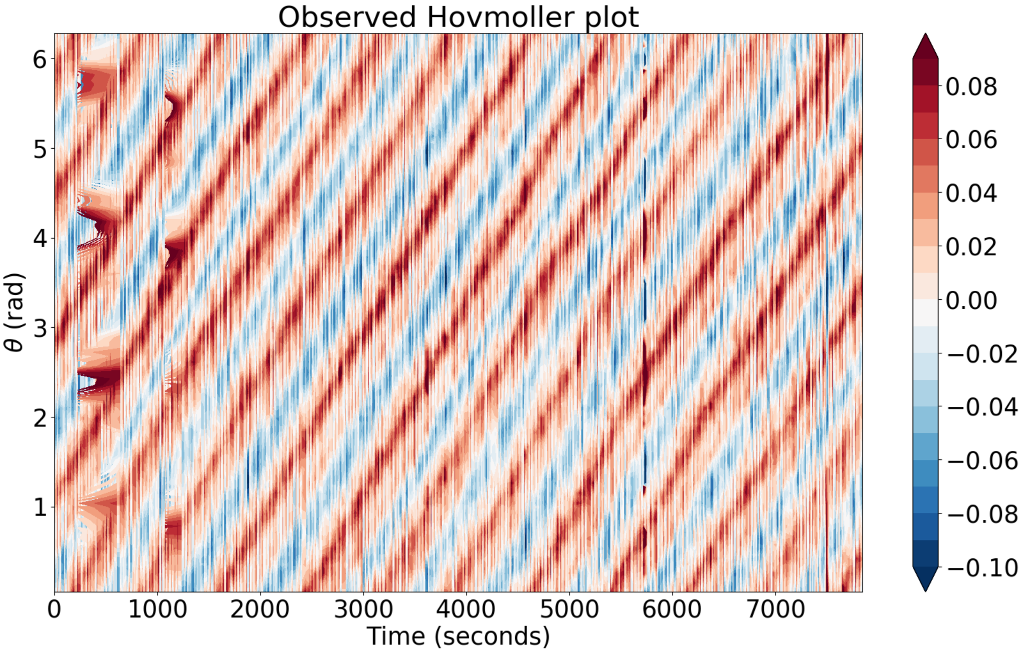


Figure 6.41: Hovmöller plot for level 4 at r=5.25cm for 3 rad/s

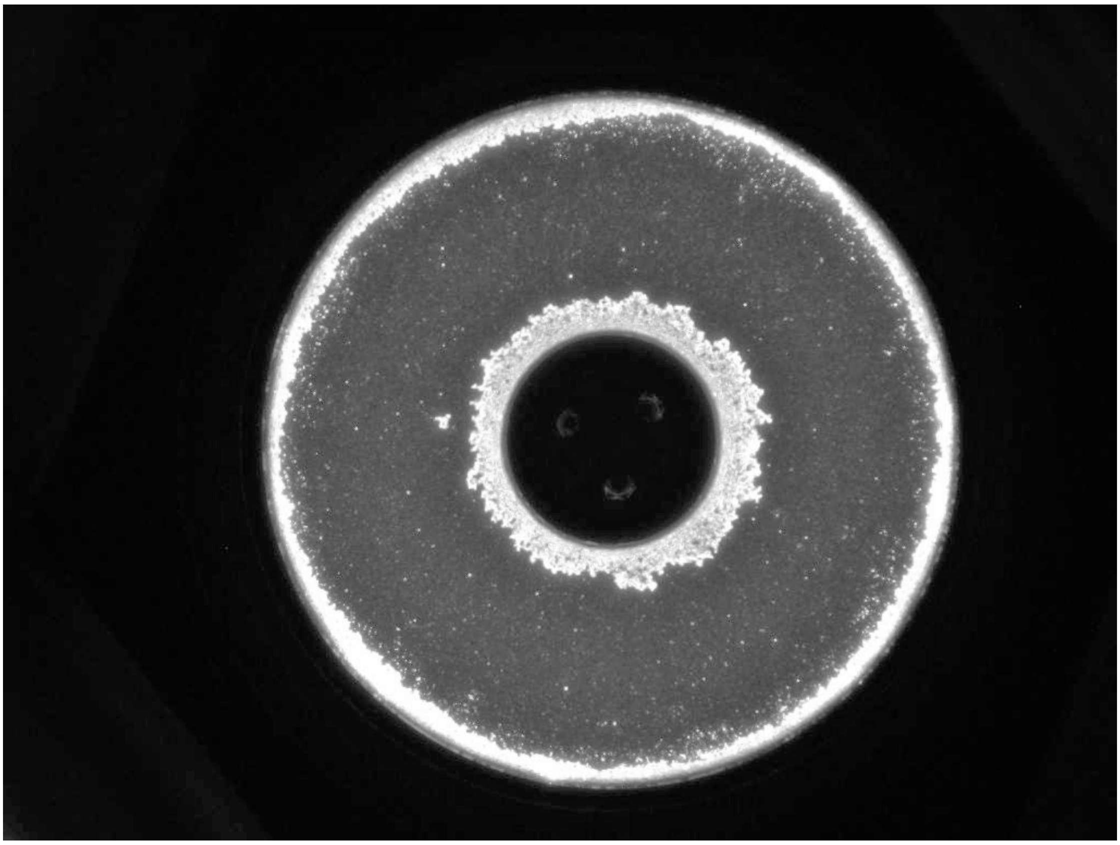


Figure 6.42: Image of the working fluid at the end of the 3hr run taken at level 5 for the 3 rad/s experiment

that the drift rate of the wave changes towards the end of the run. The reason for this is unclear but might be due to the unstable nature of the working fluid at this rotation rate.

6.4.4.4 Level 5

The image at the end of the experiment for level 5 is shown in figure (6.42). The clump of particles at the inner wall as observed in other levels is still present here indicating that this formation is near the top of the annulus. It could be that this is a formation between the lid and inner wall of the annulus. There are also particles at the outer wall but these are not as clumped as the one observed at the inner wall. Looking at a sequence of images a fluid flow can be observed. At various points, the fluid seems to be pulling particles from the outer wall. This could be due to particles getting dragged back into the flow when the wave interacts with the outer wall.

The PIV plot of the image is shown in figure (6.43). The wave that has formed does not seem to be stable at all. A lot of the clear $m = 4$ wave structure which was observed in level 4 is lost and no coherent wave travelling around the annulus can be seen although a few structures seem to indicate that there is a wave. Looking through the

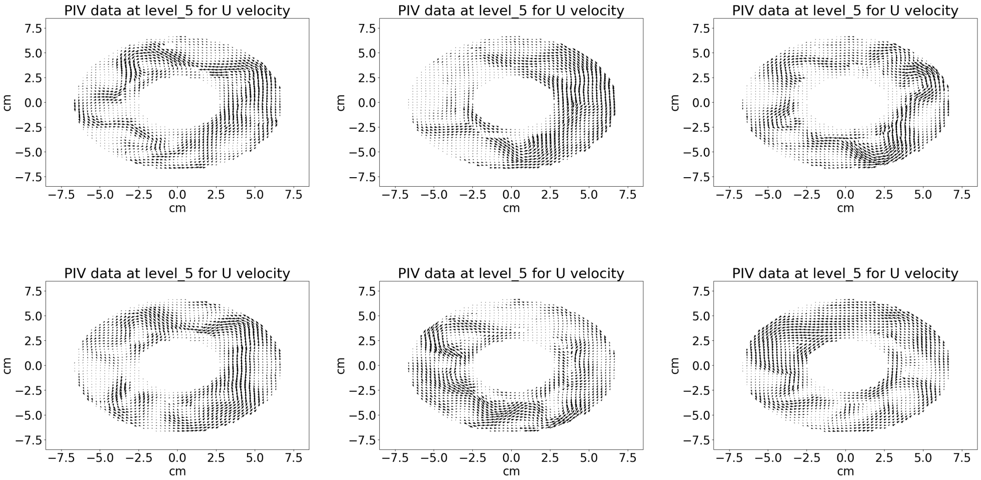


Figure 6.43: PIV result sequence towards the end of the experiment at level 5 for 3 rad/s for U velocity

PIV data, the wave seems to fluctuate a lot, sometimes disintegrating and then reforming as the experiment goes on. It could be that the faster rotation rate along with rigid upper boundary has played a role in this.

The Hovmöller plot for this case is shown in figure (6.44). Wave structures are vaguely visible at this level but the plot is very noisy. Unlike the fluid at level 4, there seems to be some problem with the formation of the wave at the early stages of the experiment. From here on the waves seem to get more and more compressed and extra structures start to form destabilising the wave. This corresponds to what is observed in the PIV plots and seems to be due to the faster rotation rate and the top boundary.

6.4.4.5 Temperature

Data from the temperature sensors at the outer wall are presented in figure (6.45). As before oscillations can be observed. Sensor 5 and 4 are nicely grouped with the data from other sensors, this indicates that fluids from different layers are interacting and mixing well. The amplitude of the oscillation is larger than what is previously observed, this relates to the faster rotation rate mixing fluid of different temperatures from different layers. Sensor 8 again seems to be a bigger anomaly than before, the reason for this is not clear but might have to do with how the fluid is behaving when interacting with the top boundary.

6.4.4.6 Summary for rotation at 3 rad/s

The high rotation rate caused a lot of problems when trying to gain accurate observations. At the lower level of the annulus, the flow seems to be stationary or very slow

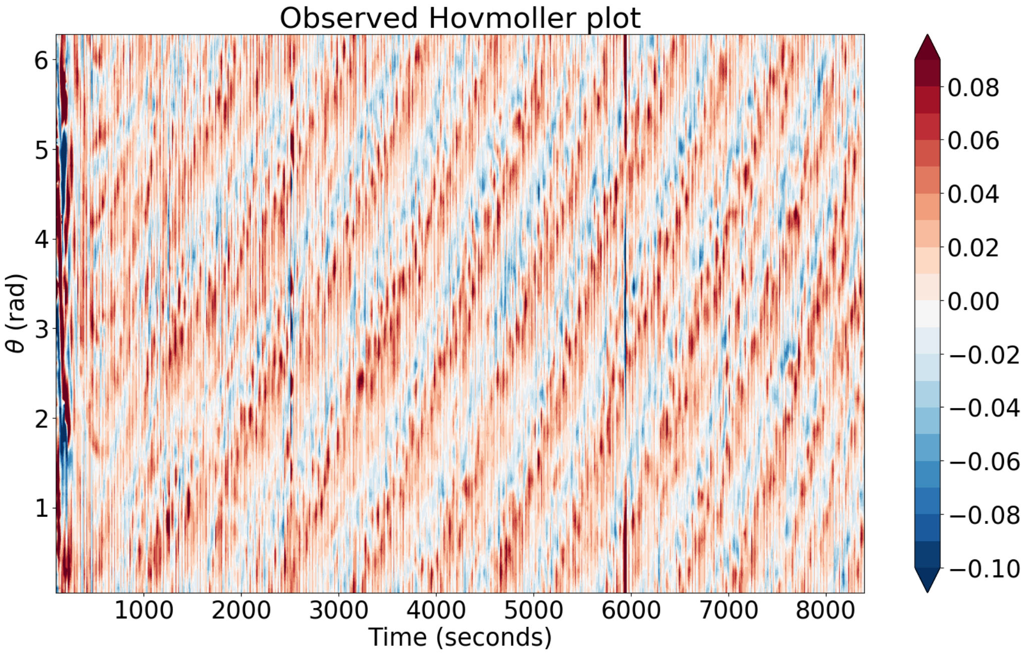


Figure 6.44: Hovmöller plot for level 5 at r=5.25cm for 3 rad/s for U velocity

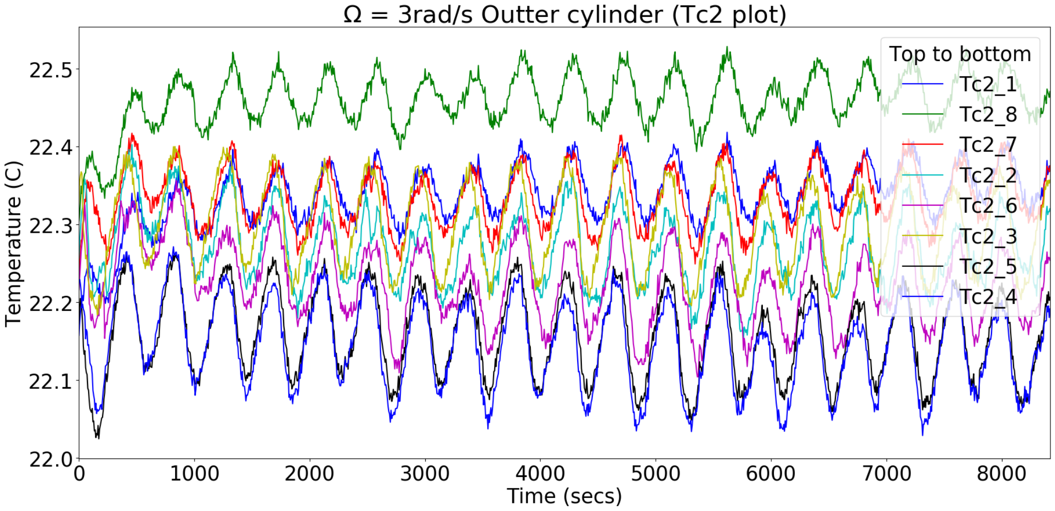


Figure 6.45: Temperature at the outer wall at 3 rad/s. The position for thermocouples on the outer cylinder is (from top to bottom): 1, 8, 7, 2, 6, 3, 5, 4.

when it comes to rotating around the annulus. It is only at level 4 where a clear wave is observed with level 5 also having a lot of problems. It could be that the faster rotation rate is causing a lot of problems with the formation of the waves. Coupled with this, the high rotation rate also caused the seeding particles to clump up at the inner wall and outer wall, and the observations overall were very badly affected.

These observations are corroborated by what is observed in the PIV plots, where a lot of messy flows are observed at lower levels, with areas showing very strong flows and other areas being very data poor. The higher levels do show an $m = 4$ wave, this is observed at level 4 and then breaking down again at level 5.

Hovmöller plots also show this $m = 4$ wave being observed at level 4 with level 5 showing more chaotic flow with a vague wave structure. The later might be due to the fluid interaction with the top boundary at this high rotation rate. The temperature plots are very good and show a steady oscillation indicating a wave travelling around the annulus.

6.5 Conclusions

In this chapter results were presented for experiments done at a velocity of 1 rad/s, 2.5 rad/s and 3 rad/s. These experiments aimed to test the new equipment that had been installed at AOPP and to get data that can be used for DA. Young and Read (2013) had used an older experiment for DA where the rotation rate of thre annulus changes every 20 minutes. But as this is the first time EnKF will be applied to this system it was decided to do a long run to get stable data at different rotation rates.

Overall the experiments showed behaviour that is expected at the various rotation rates. It was very difficult to observe the behaviour of the fluid at lower levels consistently. But much clearer behaviour was observed at higher levels. This shows how the fluid at higher levels can obscure the camera view when it comes to observing deep inside the apparatus.

The experiment at 1 rad/s showed a steady $m = 3$ wave that travelled around the annulus. This could be observed in the images as well as being confirmed by the PIV data. At lower levels, the behaviour is hard to garner as observed in both the images and PIV but from level 3 onwards the observations become clearer.

Increasing the rotation rate to 2.5 rad/s also increases the complexity of the behaviour that is observed. An $m = 4$ wave is observed around the annulus. Observations at lower levels are again poor and get better with the height of the observed level. The Hovmöller plots for this run show the $m = 4$ wave very clearly. The structures can be followed throughout the run. There are a few clumps of particles at this rotation rate but they don't seem to interfere with the PIV calculation.

The last experiment at 3 rad/s caused a lot of problems for the PIV software to generate accurate observations. At lower levels, it seems that the fluid was stuck with only the higher levels showing fluid circulating around the annulus. This was confirmed by the PIV plot and the Hovmöller plots. An $m = 4$ wave is observed at higher levels and the wave slowing drifts around the annulus. There were also issues with a massive clump of particles along the inner cylinder. It seems that the high rotation rate was pushing the particles towards the inner and outer walls of the annulus. At level 5 there also seems to be issues with the fluid interacting with the top boundary, resulting in a breakdown of the clear $m = 4$ wave observed at level 4.

Overall the one size fits all approach to PIV settings in DynamicStudio for analysing images does not seem to work well. Future work will have to try and fine tune the settings for observations taken at a different level and different rotation rates. That being said, the data at level 4 for all experiments seems to be reasonably good and can be used further in data assimilation against a simulation.

Chapter 7

EnKF and the Twin Setup using MORALS

7.1 Introduction

Data assimilation (DA) has become very important in fields such as weather prediction and oceanography (Evensen, 2003). But the process is not as simple as just coupling real life observation and simulation, and then using it for prediction. The more complex the system/model, the more time is needed to fine tune DA settings and parameters. In most cases, synthetic data are used or a twin experiment is carried out to evaluate DA settings and their sensitivity to different parameters (Reichle et al., 2002; Burgers et al., 2002).

This chapter presents the results for Ensemble Kalman Filter (EnKF, chapter 4 for more details) twin experiments which were carried out using the MORALS code, which is a rotating annulus code that solves the Navier-Stokes equations in the context of a Boussinesq, viscous, incompressible thermally conducting liquid (chapter 3 for more details on MORALS). The twin experiment is carried out to test different scenarios for EnKF with the aim of finding suitable settings for the model and EnKF, which will then be applied for observations obtained via experiments to test model prediction. For the twin experiment studies will be carried out at different rotation rates with different observational settings (see section 7.6, 7.7, 7.8, 7.9 for results).

A high resolution model is first run to create synthetic data that emulates the experiments. This data will be used to create the ‘truth ’that will be used in the data assimilation. An ensemble of low resolution models will then be used to test the capabilities of EnKF. This twin experiment gives the opportunity to test EnKF while knowing all the details/settings about the ‘truth ’and the ensembles. The twin experiment also makes it easier to line up observation points with the ensemble model when it comes to data assimilation. This provides an opportunity to identify how the parameters of EnKF can best be chosen for the different scenarios. This will help provide a baseline for EnKF's abilities when it comes to predicting and tracking observations.

Section (7.2) outlines the setup for the twin experiments. Here different aspects of the setup are described such as the grid in section (7.2.1), the MORALS code in section (7.2.2), and the Python EnKF code in section (7.2.3). How DA performs when fewer observations are included is an important issue, as is the way the observation points are chosen. The observational settings were therefore varied in DA experiments to test their effects. These different observational settings used in the twin experiment are described in section (7.3) and finally the errors and associated perturbations that are used in the twin experiment are detailed in section (7.5).

The results for the different studies are then presented, first for when no data assimilation was done in section (7.4) for rotation at 1 rad/s, 2 rad/s, 2.5 rad/s and 3 rad/s. This section shows how the low resolution ensemble behaves when it is set free to follow its own path. Finally results are presented for data assimilation for different rotation rates at 1 rad/s, 2 rad/s, 2.5 rad/s and 3 rad/s in sections (7.6), (7.7), (7.8) and (7.9) respectively. Subsections in each of these sections present results for the different observational settings that were used.

7.2 Setup

7.2.1 Grid sizes

Two grids are used for the twin study, the observation/truth grid and the ensemble grid. The observations/truth are obtained by running a high resolution model and has 96 points in the radial r-direction and the vertical z-direction, so $II = 96$ and $KK = 96$. There are 64 points in the azimuthal θ-direction, so $JJ = 2^{(MJ+1)} = 64$, where $MJ = 5$.

By comparison, the low resolution ensemble model had 24 points in the radial and vertical direction ($II, KK = 24$). However, the full 64 points in the azimuthal direction ($JJ = 64$) were retained in the low resolution ensemble model. This is to make sure that there is enough resolution for fluid structures to properly form during the simulation and that information correctly transfers/travels around the model. When significantly less than 64 points were used (e.g. 32 or 16), the code still ran, but the number of azimuthal waves in the annulus changed significantly, and the drift speed of the pattern became incorrect. This behaviour makes it difficult to get satisfactory results from the DA. Therefore models with low azimuthal resolution were considered inadequate and a model with 64 points was used. When studying the system at higher rotation rates (e.g. at 3 rad/s) the number of points in the azimuthal grid was increased to 128 to make sure the fluid details were sufficiently resolved. This was done for both the high resolution truth and the low resolution ensemble. This mean that for the low resolution cases the total number of vectors is $n = 147456$ $(24 \times 24 \times 64 \times 4)$ for each ensemble and $p = 2359296$ for the high resolution model.

7.2.2 MORALS Code

The Meteorological Office/Oxford Rotating Annulus Laboratory Simulation (MORALS) is written in a combination of Fortran 77 and 90 with a shell script being used as a wrapper to bring the different parts of the codes together. More details about the MORALS code are given in chapter (3) and can be found online at MORALS (2020). This section briefly outlines how MORALS is used for the twin experiment study.

The running of the MORALS code for any resolution can be divided into three parts: first run the 2D axisymmetric model, next use the axisymmetric model output to create suitable initial conditions for the 3D model, finally run the 3D model. Each of these steps come with their respective parameter files, where aspects of the model behaviour can be changed. The parameter files contain user defined instructions such as grid resolution (II, KK, MJ), time step size (s), how long the simulation should be run and at what intervals full field data should be saved. It also contains physical constants that will be used when solving the Navier-Stokes equations such as the dimensions of the cylinder (cm), the temperature of the inner and outer cylinder wall ($°C$), rotation rate (rad/s), density, viscosity and thermal diffusivity parameters (table 7.1).

7.2.2.1 Running the simulation

After reading the relevant parameter file the 2D axisymmetric model is simulated. In the next stage, the 2D data are used to initialise the 3D model. After the creation of the 3D grid, the 2D data is copied to each θ-point on the 3D grid. A nonaxisymmetric perturbation can then be added to this data to help baroclinic waves to develop. There are a few ways of adding perturbations that are available in MORALS, which are all applied to the temperature field.

For the studies presented here a non-axisymmetric Gaussian perturbation of standard deviation of $0.1°C$ is added to each point in the temperature field. The current CPU time is used as the seed to generate the Gaussian distribution. After the creation of the 3D model using the 2D model, the 3D simulation is run to generate results. When DA is done on the ensemble, the 3D data is read into the system, corrected using the DA algorithm and then this corrected values are used to continue the simulation.

7.2.2.2 Parameters

As the aim for the twin experiments is to gain suitable setting for EnKF to be used with experimental data, the parameters for the studies used here are the same as the experimental equipment. The inner cylinder wall is at 2.5cm from the vertical axis of rotation, with the outer wall at 8.0cm and a height of 14.0cm. The temperature of

Annulus parameters	
Inner cylinder radius	a=$2.5cm$
Outer cylinder radius	b=$8.0cm$
Depth of fluid	d=$14.0cm$
Inner cylinder temperature	T$_a$=$18.0°C$
Outer cylinder temperature	T$_b$=$22.0°C$
Fluid properties	
Working fluid	83% water, 17% glycerol
Density	$\rho = 1.044g/cm^3$
Kinematic viscosity	$\nu = 1.715 \times 10^{-2}cm^2/s$
Thermal diffusivity	$\kappa = 1.284 \times 10^{-3}cm^2/s$

Table 7.1: Summary of annulus and fluid properties used in the simulations

the inner wall is $18.0C$ and the outer wall is $22.0C$ with the working fluid mixture begin 17% (by volume) mixture of glycerol and water. Other constants such as density, viscosity, thermal diffusivity parameters, etc. are not changed either. A summary of the parameters used in the simulation are shown in table (7.1).

7.2.3 Python code

For data assimilation a code needs to be written which can:

1. edit and run the MORALS Fortran code,
2. read and edit the data created by the MORALS code,
3. organise the multiple files created during the data assimilation run,
4. execute DA on the system.

Python 2.7 was chosen as the language of choice for this task as it has the capabilities to tackle all the different aspects due to its vast library of modules. Python can then be further used to analyse and easily plot the results at the end. Since Python is freely available, the code can be run easily on any machine without any licensing issues. So a group of Python files are created which deal with the different aspect of the study and linked together using a 'main'file.

In the code a Python parameter file is used to define all the criteria that will be used for the MORALS simulation like the rotation rate, time step, etc. and data assimilation parameters such as ensemble number, errors, file locations, etc. At the beginning of the simulation, this parameter file is read and the respective files are edited to make sure they contain the correct settings. This is followed by the 2D axisymmetric model being simulated. A loop is used which turns the 2D data into an ensemble of 3D data and runs the 3D simulations to create the ensemble.

Before any data assimilation can be done, the grid points between the high resolution truth and low resolution ensemble need to be correlated. For this, a Python code is used to first read a binary file for the ensemble ($.unx$) and obtain the r, z and θ grid positions in the ensemble. As all ensemble members have the same resolution they also have the same grid positions and therefore only one binary file needs to be read to get positional values for all the ensemble.

Using the positional values of r, z and θ in the low resolution ensemble the closest corresponding positional values in the high resolution truth are obtained. This data is used to create truncated data files of the observation which will be used at each of the data assimilation steps. Creating these separate set of data files are necessary as the original files a very large and reading an ensemble of them into the system every time DA occurs will slow down the system and depending on how DA is done may even crash the system by using up all of the RAM.

The DA loop is now started, where the ensemble and observation are read into the system. The analysis step is then performed and the ensemble files are corrected using the solution provided by DA. The MORALS script is edited and the 3D simulation is restarted from this point. This DA loop continues until the end of the study.

7.3 Types of Observation

Deciding how much and which observations to use in the twin setup study was not straightforward, as a great deal of information is available from the high resolution model. But as the number of observations obtained in any experiment will not be as extensive as numerical models, it was decided that this number would be kept low in the twin setup to aid comparison. This will help recreate the conditions of a real life problem and test EnKF capabilities. MORALS provides data for U, V and W velocity along with temperature at each point in the annulus.

It is important to remember that MORALS uses not only a staggered grid but also a stretched grid in the r and z direction. As how the gird is stretched is dependent on the number of grid points, this means that the grid points in the high resolution truth don't exactly line up with the low resolution ensembles. When choosing observation points, it seemed simpler to choose the point on the low resolution ensemble, as there are fewer points to go through, and then try and find the nearest corresponding point on the high resolution model. As the high resolution model has more points there is a good chance that a point close to the one chosen on the low resolution ensemble will be found. It was also decided that there would be no interpolation done on the observational data, as the error that will be incurred due to this will be a part of the observational error given to the system. This helps make it a more realistic test for EnKF as well.

When the simulation using the low resolution ensemble was started, the grid points in the high resolution model were truncated to match the low resolution. The data at these points are saved in separate data files corresponding to the simulation time. In the high resolution runs, full-field data were saved every minute, hence a file is created for each of these when truncating the data. This was done because reading the high resolution data file takes a very long time due to its size and the date structure of the file.

When trying to decide the scheme for choosing the observation points two different ways of proceeding were entertained. Either you pick new random points every time you have to do data assimilation or you use the same random points at every data assimilation step. Both of these studies were carried out. When choosing the random points, the last few points near the boundary were ignored (the first 2 and last 2 points) as the focus is to get the larger structures created in the annulus correct and not those structures near the boundary. When the random points have been chosen, they must be ordered so that they are in ascending order as they are stored on the data file.

7.3.1 Truth

Data for the high resolution observations are truncated into files which correspond to full field data saved at every 60 seconds. When it comes to DA, the data file corresponding to the time in the ensemble simulation is just read into the system. Once this data is read into the system it is simple to use a map that corresponds to the type of observation that is being used (Mid, Randph, etc.) on the data to get observational data.

7.3.2 Mid

In this setup the nine middle points in each θ-planes are selected as observations for DA, i.e. points $r = 11, 12, 13$ for $z = 11, 12, 13$ in each θ. This is done for each ensemble member along with their respective points from the truncated high resolution model. This gives a total of 576 grid points (9×64) that are used for DA. With four variables observed at each point, this gives a vector of $p = 2304$ components (576×4). A few variations of this method were also tested which will be discussed in their respective results sections.

In Python, this simply meant taking the ensemble and observation matrix and extracting the middle three rows in z in ascending order. Then looking at the middle three r points in ascending order. This is done for all θ. These vectors are used to create new matrices HX and HY for the ensemble and observation respectively. The main fragment of code used to extract the data is shown below.

```
HX = x[:,:,II/2-1:II/2+2,KK/2-1:KK/2+2,:]
HY = Y[:,:,II/2-1:II/2+2,KK/2-1:KK/2+2]
```

Here x is the ensemble matrix and Y is the observation matrix. II, KK correspond to points in r and z respectively.

7.3.3 randph

The second method of obtaining observations is to use different random points at each θ every time DA is done. Usually, this was 9 random points in the $r - z$ plane for each θ, the same number as the previous method, and as before this leads to the same number of the total observed vector components $p = 2304$.

In the Python code this is created by using the *numpy.random.randint* command to generate random points in the r and z grids. The r and z values are coupled together and are then sorted in ascending order for z and then r respectively. A check is made if any positions are repeated and if so they are replaced with new values. A snippet of the code is shown below.

```
for i in range(JJ):
...
for k in range(n
):
b =lst[k]
Hx[:,i,k,:]=x[:,i,b[0],b[1],:]
HY[:,i,k]=Y[:,i,b[0],b[1]]
```

Here n is the number of vector components and b is the matrix that stores the positions of the values being observed in the form $b[r, z]$. The i is the θ position. So this i loop goes through all the positions in θ and data is extracted and saved in the Hx and HY matrices for the ensemble and observations respectively.

7.3.4 randpph

This method is the same as *randph* but now the random points are created only once at the start of the first DA and saved to be used when DA is done again. As before the user can decide how many points are used, but in most cases this number was kept at 576 grid points giving a total of $p = 2304$ vectors just as before. This was used to see if the reliability of EnKF results depend on whether observations are taken at the same points or taken at points which change every time.

In Python, this meant checking if this is the first data assimilation of the study and if so generating the random points which are going to be observed. As before, the r and z points are coupled and sorted in ascending order and checked for repetition. These are then saved in a text file which will be read for the other DA steps. Using these points observation matrices HX and HY are created and used in DA. If this is not the first data assimilation then the text file is read to recover the observed points and data is extracted as before. A bit of the code used for this is shown below.

```
b=0
for i in range(JJ):
for j in range(g):
c=a[b]
```

```
Hx[:,i,j,:]=x[:,i,int(c[0]),int(c[1]),:]
HY[:,i,j]=Y[:,i,int(c[0]),int(c[1])]
b=b+1
```

Here b is the position in the list that has all the position values. g is the number of observations in θ and a is the list that holds the position information as $c[r, z]$. These are used to extract data from the ensemble and observation and saved in Hx and HY matrices respectively for data assimilation.

7.3.5 randp

The fourth method used to test DA was to look at random points in the model at every data assimilation. The difference between this method and randph is that before points were selected at every θ-position but now the θ value is also chosen randomly. The standard $p = 2304$ vectors were once again used. In many instances choosing random points will lead to an uneven distribution of observed points throughout the model, leading to data rich and data poor areas. This is a good test to see if even distribution of observation points is necessary for accurate data assimilation.

In the Python code this means that random points are generated out of all the r, z and θ grid points. They are then coupled and sorted out in ascending order,and checked for repetition. A snippet of the code is shown below

```
 for k in range(n):
b=lst[k]
Hx[:,k,:]=x[:,b[2],b[0],b[1],:]
HY[:,k]=Y[:,b[2],b[0],b[1]]
```

Here n is the number of observed points. lst stores all the points in the form $lst[\theta, r, z]$. These points are passed to b one by one in the k loop and using these, data in extracted from the full ensemble(x) and truth (Y). This data is stored in the Hx and HY matrices to be used in the DA.

7.3.6 randpp

This method is the same as Randp but now the random points created at the start of the first data assimilation are saved and used for every subsequent data assimilation. As before, in most cases $p = 2304$ observations are used for data assimilation.

In Python this means that the points are generated during the first data assimilation, coupled, sorted and checked for repetition. They are then saved in a text file to be used for the other data assimilation cycles. The code for this is more or less same as before in Randp.

7.4 No Assimilation Results

Before any data assimilation was performed, a control study was analysed where the ensemble low resolution model was run without any data assimilation. This study was carried out to observe how the low resolution system behaves without any correction and compare it to the behaviour observed in the high resolution model. It is expected that the ensembles will not converge, with each ensemble going about their own behaviour. Ideally the low resolution runs will show similar behaviour as the high resolution observations in terms of structure and wavenumber. But there will be differences in individual behaviour for DA to correct. In all cases an ensemble of $N = 10$ is run for a 100 minutes to look at the behaviour of the system.

7.4.1 Results for 1 rad/s

The first simulation was run at 1.0 rad/s with a timestep of $dt = 0.01s$, with the system stopping every 300 seconds (5 minutes) to save full field data. The simulation ran for 100 minutes giving 20 data sets for each ensemble member.

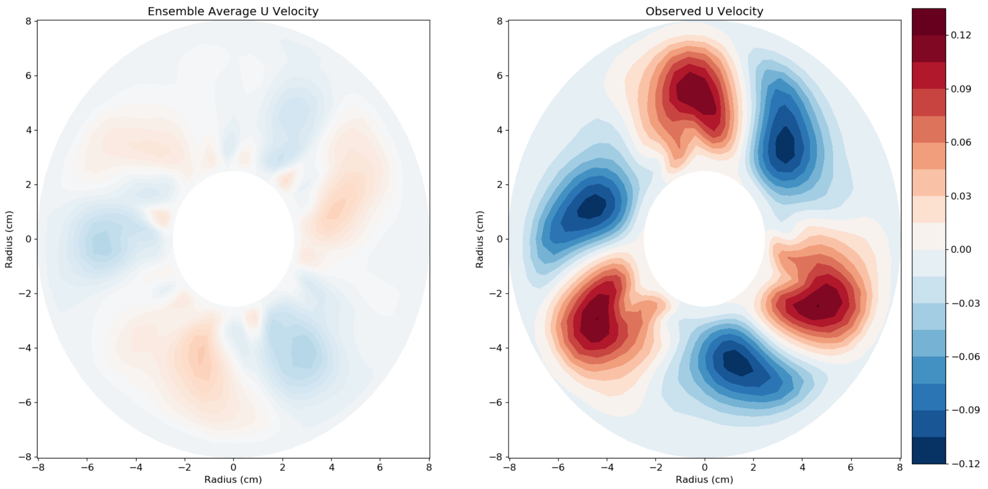
The state of the system at the end of the simulation is shown in figure (7.1a). On the left is the contour plot for the low resolution ensemble average and on the right is the contour plot for the high resolution run. Both contour plots are for U velocity data at $z = 7.0$ cm from the bottom, i.e. the mid height of the annulus and show the state of the system towards the end of the run. The high resolution run shows a $m = 3$ structure and the same can be vaguely observed in the ensemble average.

Figure (7.1b) shows the U velocity at $z = 7.0$ cm and $r = 5.25$ cm for all θ. The thick black line in the figure is the truth, the high resolution run, the thin lines are the individual ensemble members. The thick blue line is the ensemble average. As all the ensemble members have unique initial conditions, and as no DA is done, the ensemble does not converge and each individual ensemble shows unique behaviour. This is why the ensemble average is small compared to its truth counterpart.

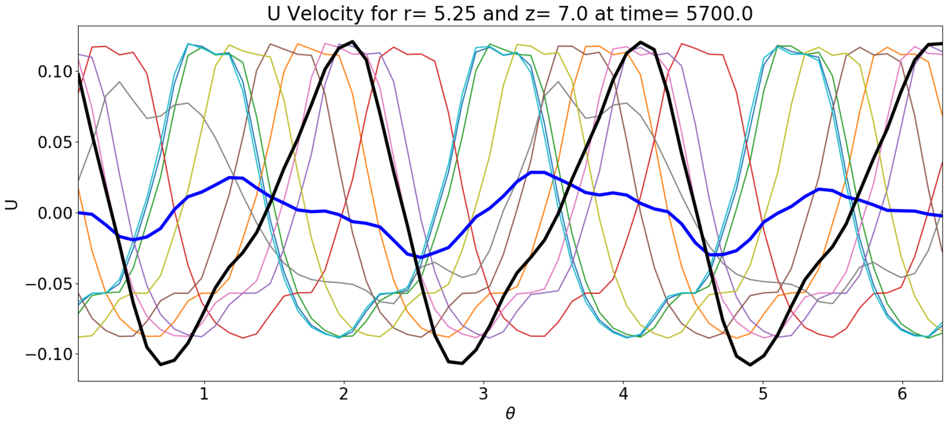
Following the individual ensembles in figure (7.1b), a $m = 3$ wavenumber can be observed here in all ensemble members indicating that the low resolution simulation is able to replicate the high resolution run structurally. A good spread is also observed in the ensembles throughout θ. This indicates that there is a good distribution of perturbations when converting the 2D model data to 3D. This leads to a good spread of the ensemble when initialising the system. This wide spread of the ensemble in state space is key when it comes to an accurate correction during data assimilation.

Figure (7.2) shows the standard deviation calculated for the ensemble. This was calculated by looking at the four variables at $r = 5.25$ cm, $z = 7.0$ cm for $\theta = 0$. The standard deviation remains well-defined throughout the run and does not vary for the large part. The temperature has the biggest standard deviation with the V and W velocities showing the smallest.

The Hovmöller plot for the high resolution simulation is shown in figure (7.3). These were made from the values of U velocity at $z = 7.0$ cm and $r = 5.25$ cm for all θ. An



(a) The left panel shows a contour plot of the U velocity for the ensemble average at the end of the run. The right panel shows the same for the high resolution run.



(b) The graph shows the U velocity around the annulus at $r = 5.25\,\text{cm}$. The thick black line is the high resolution observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

Figure 7.1: Results for 1 rad/s when no data assimilation is done at $z = 7.0\,\text{cm}$

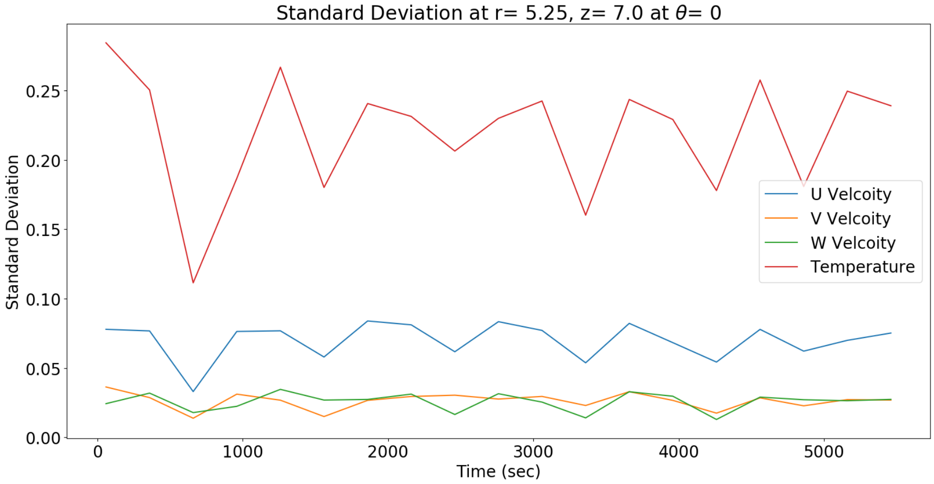


Figure 7.2: Standard deviation at 1 rad/s with no data assimilation.

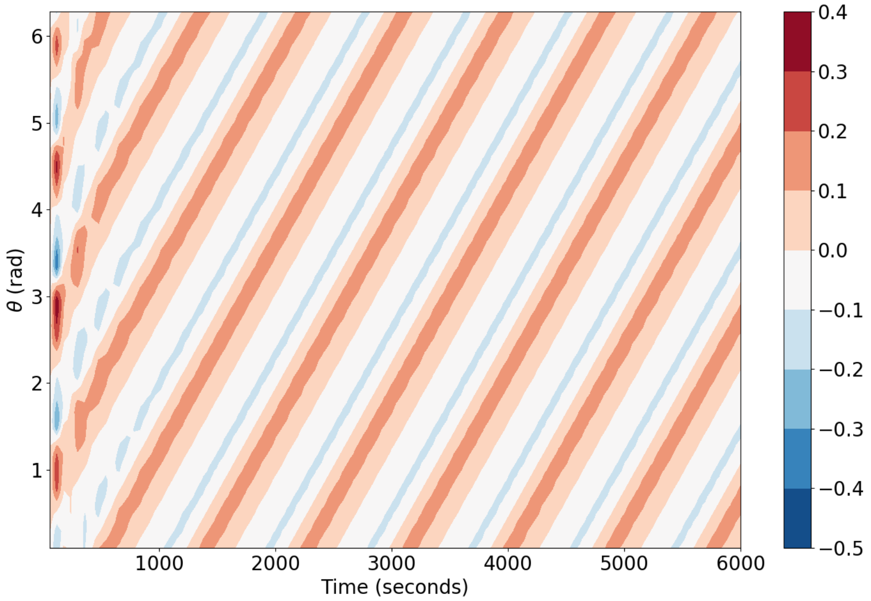


Figure 7.3: Hovmöller plot for the high resolution truth at 1 rad/s.

$m = 3$ wave is clearly seen travelling around the annulus. The drift rate of the wave can be calculated by measuring the time taken for one of these structures to travel completely around the annulus. In this case, the position of the wave was checked at a suitable time towards the beginning of the run and then checked again towards the end. From this data, the drift rate was calculated. For the high resolution run, this was found to be around 0.0023945 ± 0.0000001 rad/s. For the low resolution ensemble, the drift rate was calculated for all 10 members using the same method as above but then the values were averaged and a standard deviation was calculated. As each ensemble has different initial conditions slight variation in the structural formation and rotation rate is to be expected. Using this method a value of 0.0022580 ± 0.0000490 rad/s was calculated. Table (7.2) shows the values of the individual ensemble members along with those of the high resolution run.

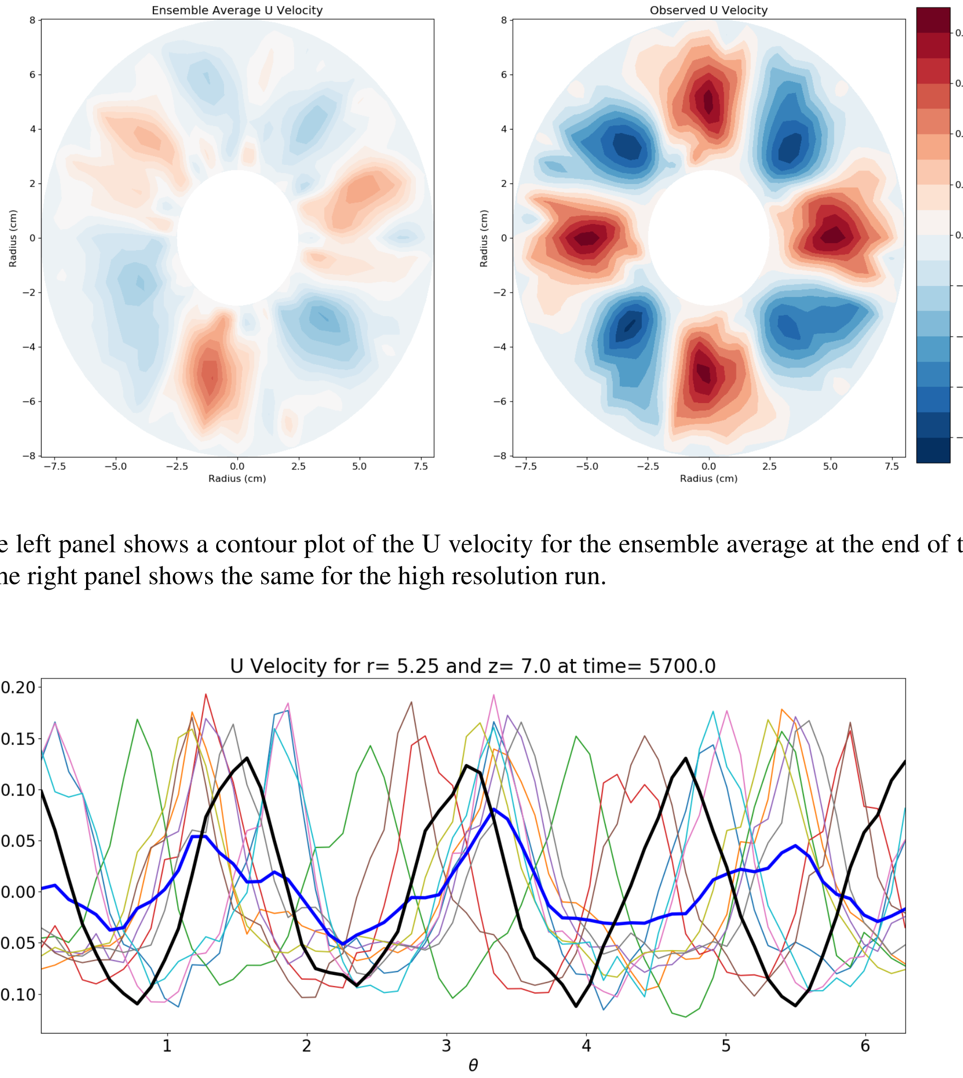
7.4.2 Results for 2 rad/s

The 2.0 rad/s simulation was also run with a timestep of $dt = 0.01$ s with the system stopping every 5 minutes to save full field data and running for a total of a 100 minutes. Figure (7.4a) shows the contour plots for this run. The higher resolution run clearly shows the $m = 4$ wave but the same cannot be said when looking at the contour plot for the low resolution ensemble average. This is partly due to the ensemble spread around the annulus as no DA is done, but is partly caused because unlike the 1 rad/s case, two different azimuthal wavenumbers, $m = 3$ and $m = 4$, appear to be stable for the low resolution model. So the ensemble is a mixture of these two types of solutions as seen figure (7.6). This multiple equilibrium has been reported by multiple studies such as Vincze, Borchert, Achatz, von Larcher, Baumann, Hertel, Remmler, Beck, Alexandrov, Egbers, Froehlich, Heuveline, Hickel and Harlander (2014); Larcher and Egbers (2005).

Looking at the U velocity graph in figure (7.4b) at $z = 7.0$ cm and $r = 5.25$ cm for all θ and tracking an ensemble across the annulus, some $m = 4$ waves can be observed. This is an encouraging result and proves that even at this higher rotation rate the low resolution model can still provide an accurate structure similar to that in the high resolution run. The thick blue line showing the low resolution ensemble average is higher than in the equivalent figure for 1 rad/s. This might be due to the initialisation values not spreading the ensemble members enough when converting the models from 2D to 3D and due to the different wavenumbers are are observed.

The standard deviation of the U,V,W and T variables calculated at $r = 5.25$ cm, $z = 7.0$ cm for $\theta = 0$, and are shown in figure (7.5). The values for the variables are equivalent to the ones observed in the previous study at 1 rad/s. The temperature again shows the highest standard deviation with V and W velocity showing the lowest. The standard deviations of these quantities are generally similar to those in the 1 rad/s case.

The Hovmöller plot for the high resolution simulation is shown in figure (7.7) and a $m = 4$ wave is observed at the end of the run. The drift rate for the wave was calculated to be around 0.0015324 ± 0.0000001. This is slower than the 1 rad/s run. The low resolution ensemble has an average drift rate of 0.0010230 ± 0.0001768 rad/s (table



(a) The left panel shows a contour plot of the U velocity for the ensemble average at the end of the run. The right panel shows the same for the high resolution run.

(b) The graph shows the U velocity around the annulus at $r = 5.25\,\text{cm}$. The thick black line is the high resolution observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

Figure 7.4: Results for 2 rad/s when no data assimilation is done at $z = 7.0\,\text{cm}$.

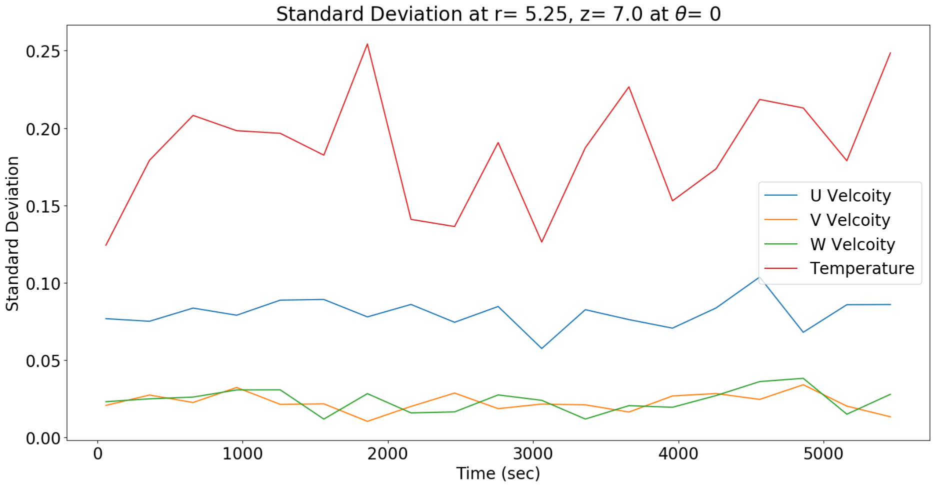


Figure 7.5: Standard deviation at 2 rad/s with no data assimilation.

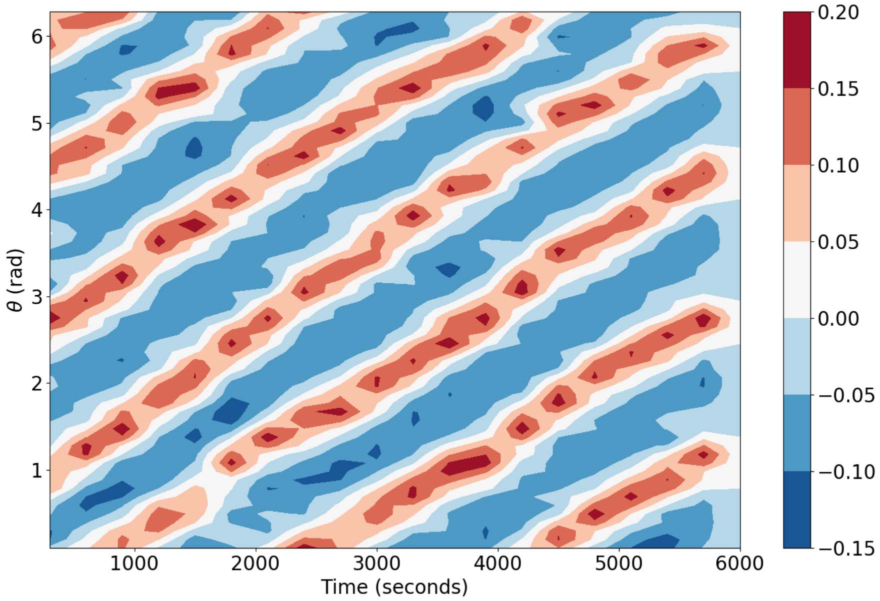
(7.2)), which is slower than the high resolution run.

Looking at the drift rate of the individual ensemble members in table (7.2) different variations are present. Ensemble members with $m = 4$ azimuthal wavenumber have systematically lower drift rates than those with $m = 3$. Two examples of this can be seen in figure (7.6), where ensemble 6 shows an $m = 4$ wavenumber solution with drift rate 0.0008945 rad/s and ensemble 8 shows an $m = 3$ wavenumber solution with drift rate 0.0011999 rad/s. This is consistent with the high resolution 2 rad/s $m = 4$ pattern having a lower drift rate than the high resolution $m = 3$ pattern found at 1 rad/s. However, there is also a resolution effect, because the $m = 4$ members of the low resolution ensemble have a lower drift rate than the high resolution $m = 4$ run. So not only is there a decrease in drift rate with higher m, but there is also a decrease in drift rate as the resolution is lowered.

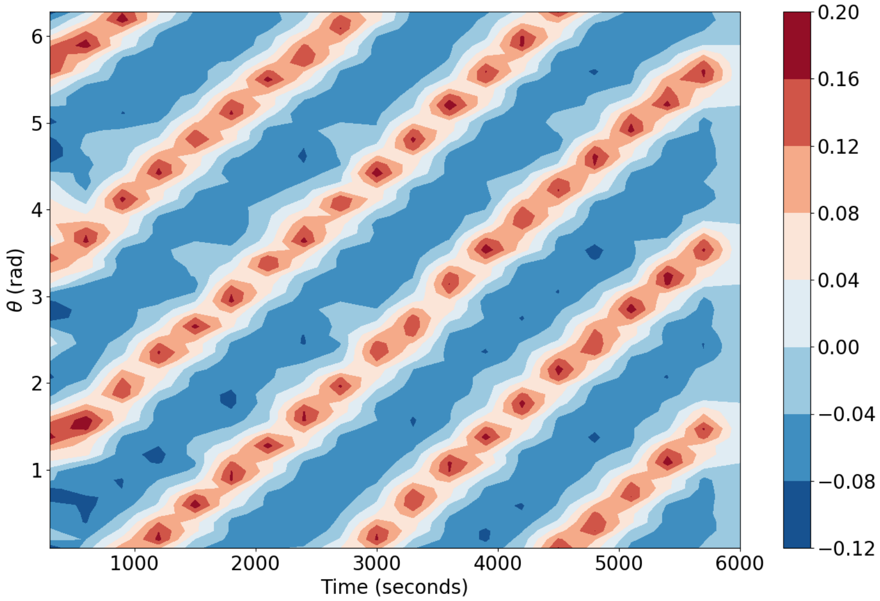
7.4.3 Results for 2.5 rad/s

The rotation rate was increased further to 2.5 rad/s and the same study as before was done with a timestep of $dt = 0.01$ s with the system stopping every 300 seconds (5 minutes) to save full field data and running for 100 minutes. The contour plots for the system at the end of the run are shown in figure (7.8a). An $m = 3$ wave is observed in the high resolution run at this time but earlier in the run the system displayed a $m = 4$ wave. So the rotating structure in the observation started as a wavenumber $m = 4$ solution but collapsed later on in the simulation. This might indicate that the $m = 3$ wave is more stable than a $m = 4$ wave for this rotation rate at high resolution. The low resolution ensemble average shows a mixture of $m = 3$ and $m = 4$ behaviour. Individual ensemble members need to be looked at to better ascertain their behaviour.

Figure (7.8b) shows the U velocity at $z = 7.0$ cm and $r = 5.25$ cm for all θ. The thick



(a) An $m = 4$ wave number present in ensemble member 6



(b) An $m = 3$ wave number present in ensemble member 8

Figure 7.6: Hovmöller plot at 2 rad/s for ensemble members 6 and 8 showing wavenumber $m = 4$ and $m = 3$ respectively.

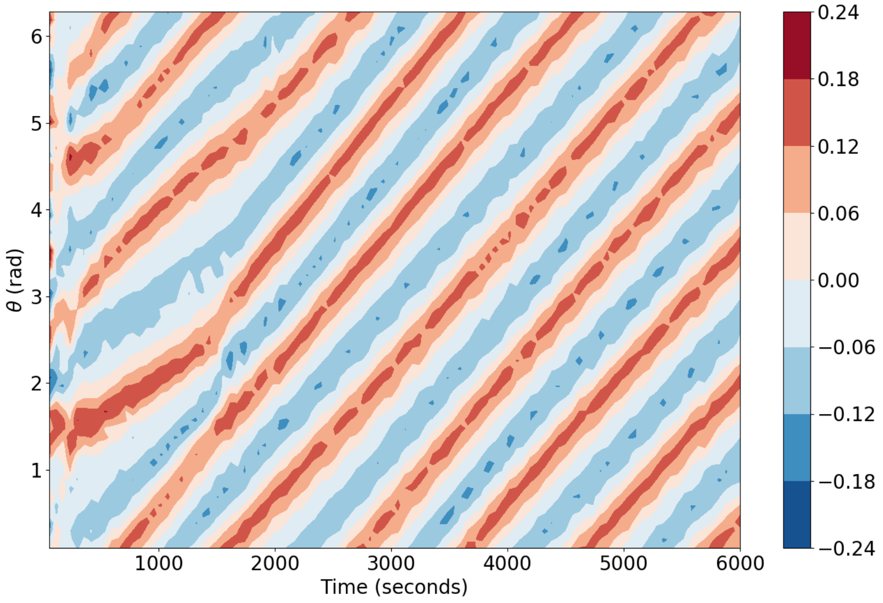
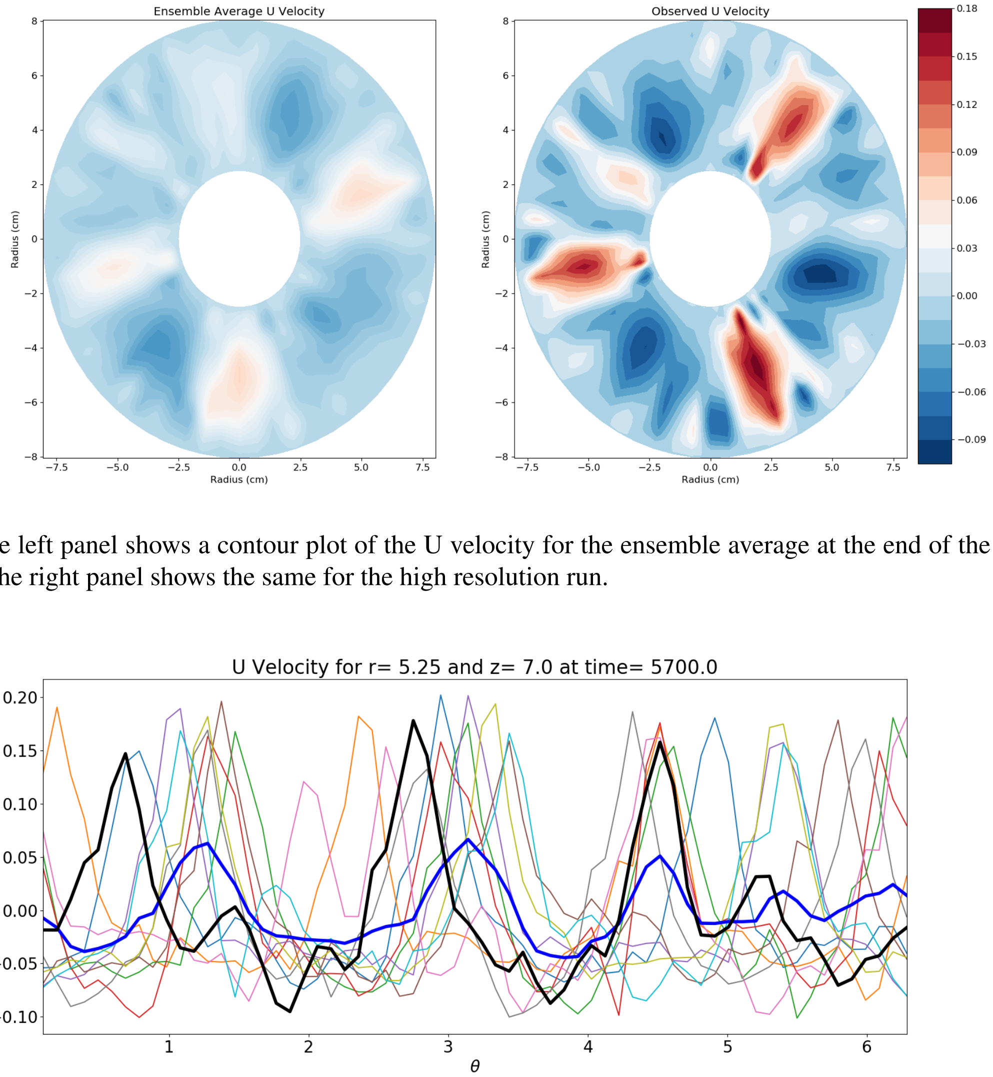


Figure 7.7: Hovmöller plot for the high resolution truth at 2 rad/s.

black line in the figure is the truth, the thin lines are the individual ensemble members and the thick blue line is the ensemble average. Looking at the truth, the structure of the U velocity is not as smooth as observed previously at lower rotation rates. Here a somewhat messy $m = 3$ wave is observed for the truth towards the end of the run. The individual ensemble members show a variety of behaviours (see figure (7.9)), with some ensemble members showing a $m = 3$ wave, others show $m = 4$ waves, while some transition between the two. This variety of wavenumber indicates that at this rotation rate the system may again take different wavenumbers when running.

The standard deviation of the ensemble is shown in figure (7.10) and even with the higher rotation rate, the amplitude of the standard deviation is still very similar to that at lower rotation rates. As before temperature has the highest standard deviation, with the V and W velocities showing the lowest. There is a rise in the standard deviation for all the variables just after 3000 seconds. This might be related to the wavenumber of some of the ensemble members changing from $m = 4$ to $m = 3$ wavenumber.

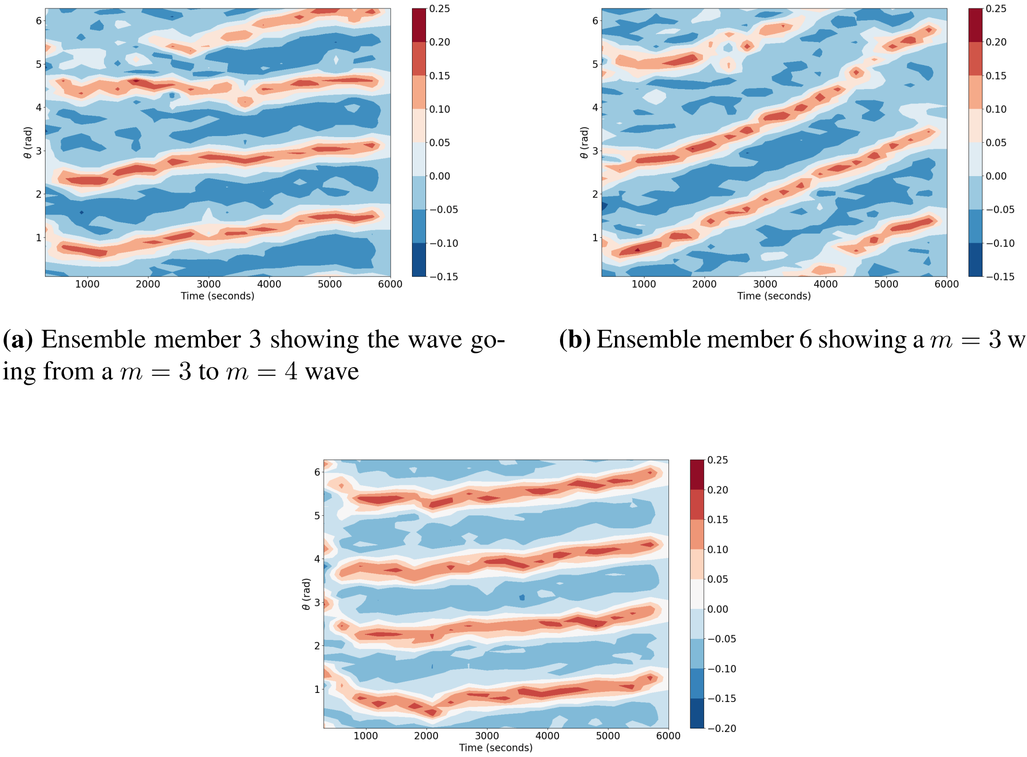
The Hovmöller plot for the high resolution simulation is shown in figure (7.11) and a $m = 3$ wave is observed. The drift rate for the wave is 0.0001197 ± 0.0000001 rad/s, which is the lower than the 1 rad/s and 2 rad/s results. The ensemble average also shows a very slow drift rate with 0.0004566 ± 0.0002205 rad/s (table 7.2). Again there are different values for individual ensemble members. This is due not only because of the different wavenumbers but also to the structural changes that happen mid way into the run for some of the ensembles. A selection of Hovmöller plots of ensemble members is shown in figure (7.9). At this rotation velocity, the flow seems to be at a borderline



(a) The left panel shows a contour plot of the U velocity for the ensemble average at the end of the run. The right panel shows the same for the high resolution run.

(b) The graph shows the U velocity across the annulus at $r = 5.25\,\text{cm}$. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

Figure 7.8: Results for 2.5 rad/s when no data assimilation is done at $z = 7.0\,\text{cm}$.



(a) Ensemble member 3 showing the wave going from a $m = 3$ to $m = 4$ wave

(b) Ensemble member 6 showing a $m = 3$ wave

(c) Ensemble member 8 showing a $m = 4$ wave

Figure 7.9: Hovmöller plot at 2.5 rad/s for ensemble members 3, 6 and 8 showing the diversity of behaviour observed at this velocity.

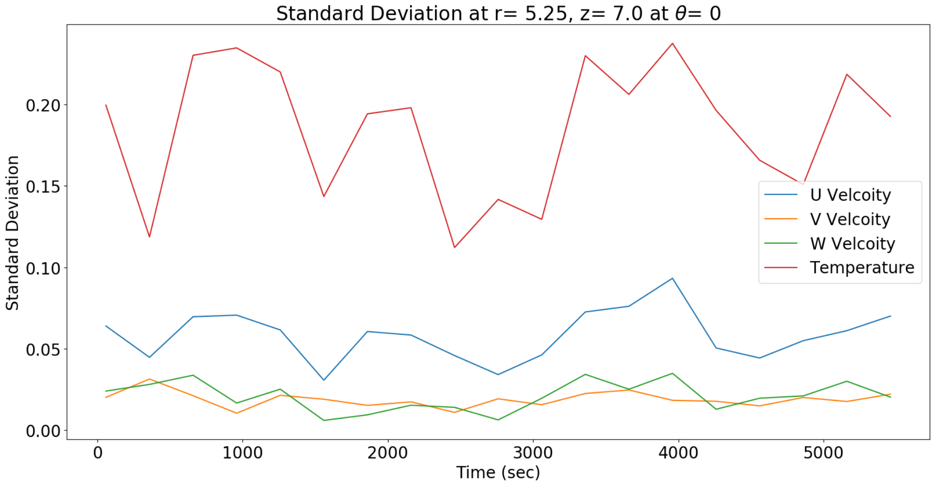


Figure 7.10: Standard deviation at 2.5 rad/s with no data assimilation.

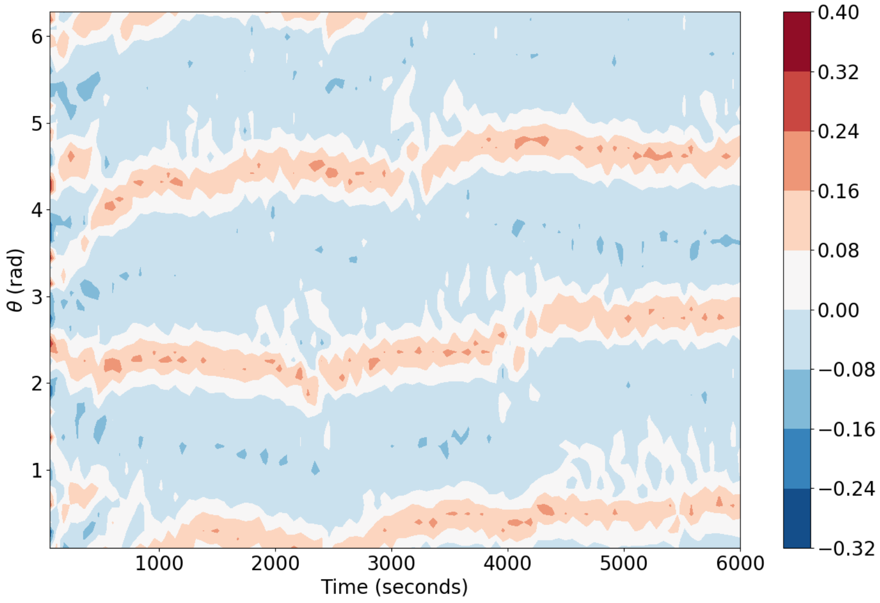


Figure 7.11: Hovmöller plot for the high resolution truth at 2.5 rad/s.

between two different wavenumber solutions being preferred, and this may contribute towards the slow drift rate.

7.4.4 Results for 3 rad/s

As seen at 2.5 rad/s, high rotation rate brings more instability into the system. To get a better understanding of how the ensemble is behaving at this higher rotation rate it is now stopped every 60 seconds to output full field data. This gives a total of 100 output files for each ensemble for the 100 minute run. Other settings were also changed to increase the resolution, such as making the timestep smaller, $dt = 0.005$, and increasing the θ-resolution to 128 points for both the hight and low resolution runs. Apart from these changes, other parameters were kept the same, with the high resolution run still having $r, z = 96$ points and the low resolution runs having $r, z = 24$ points.

The contour plot for the system at the end of the run is shown in figure (7.12a). As observed the high resolution plot shows a $m = 3$ wave. This wave is stable and remains a $m = 3$ wave throughout the run. Unlike the 2.5 rad/s run, there is no change of wavenumber in this run. Although there is more unstable behaviour, with more small scale flow, the dominant $m = 3$ wavenumber remains throughout the run. Looking at the low resolution ensemble, the system shows a large variety of behaviour as before, with both $m = 3$ and $m = 4$ wave structures being observed for ensemble members. The standard deviation for the ensemble is shown in figure (7.14). Although the plot

	1 rad/s	2 rad/s	2.5 rad/s	3 rad/s
Truth	0.0023945	0.0015324	0.0001197	0.0011454
Ensemble 1	0.0022907	0.0011921	0.0006136	0.0008827
Ensemble 2	0.0023375	0.0011921	0.0005035	0.0009688
Ensemble 3	0.0021972	0.0008290	0.0001510	0.0008827
Ensemble 4	0.0021817	0.0008415	0.0000503	0.0009350
Ensemble 5	0.0022362	0.0011921	0.0005035	0.0009817
Ensemble 6	0.0022498	0.0008945	0.0006293	0.0010549
Ensemble 7	0.0022498	0.0008999	0.0006797	0.0009473
Ensemble 8	0.0023375	0.0011999	0.0002014	0.0008612
Ensemble 9	0.0022498	0.0012083	0.0005790	0.0009258
Ensemble 10	0.0022498	0.0007804	0.0006545	0.0009688
Average	0.0022580	0.0010230	0.0004566	0.0009409
STD	0.0000490	0.0001768	0.0002205	0.0000545

Table 7.2: Drift rate in rad/s of the waves at various rotation rates for the high resolution run (truth) and the low resolution ensemble.

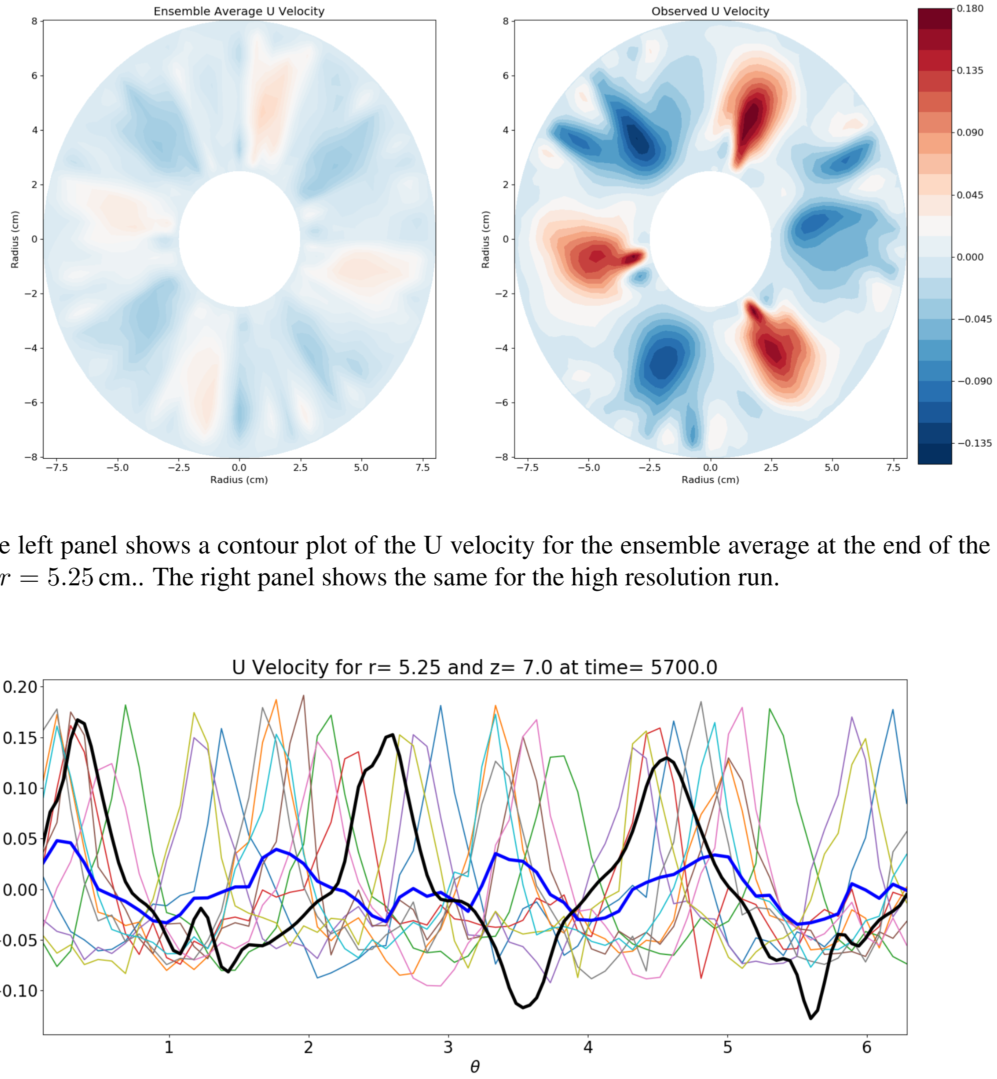
looks chaotic the amplitude of the standard deviations are still the same as before.

The Hovmöller plot for the high resolution simulation is shown in figure (7.15) which shows a $m = 3$ wave. The drift rate was calculated to be 0.0011454 ± 0.0000001 rad/s. The low resolution ensemble had an average drift rate of 0.0009409 ± 0000545 rad/s (table (7.2)). A few different types of behaviour are observed in the ensemble members, in two cases the drift rate of the wave completely changed as seen in figure (7.13). For these ensemble members, the more stable lower m waves were used to calculate the drift rate.

7.4.5 Summary

Before any data assimilation was done a benchmark/control study was done where the ensemble model was run without any data assimilation. Studies were done at 1 rad/s, 2 rad/s, 2.5 rad/s and 3 rad/s. In all cases, an ensemble of $N = 10$ was run for around a 100 minutes to look at the behaviour of the system. The timestep was kept at $dt = 0.01$ for the study apart for 3 rad/s where it was decreased to a $dt = 0.005$. This increase in resolution and decrease in timestep was done to get a more accurate high resolution result, for 3 rad/s is a very chaotic flow.

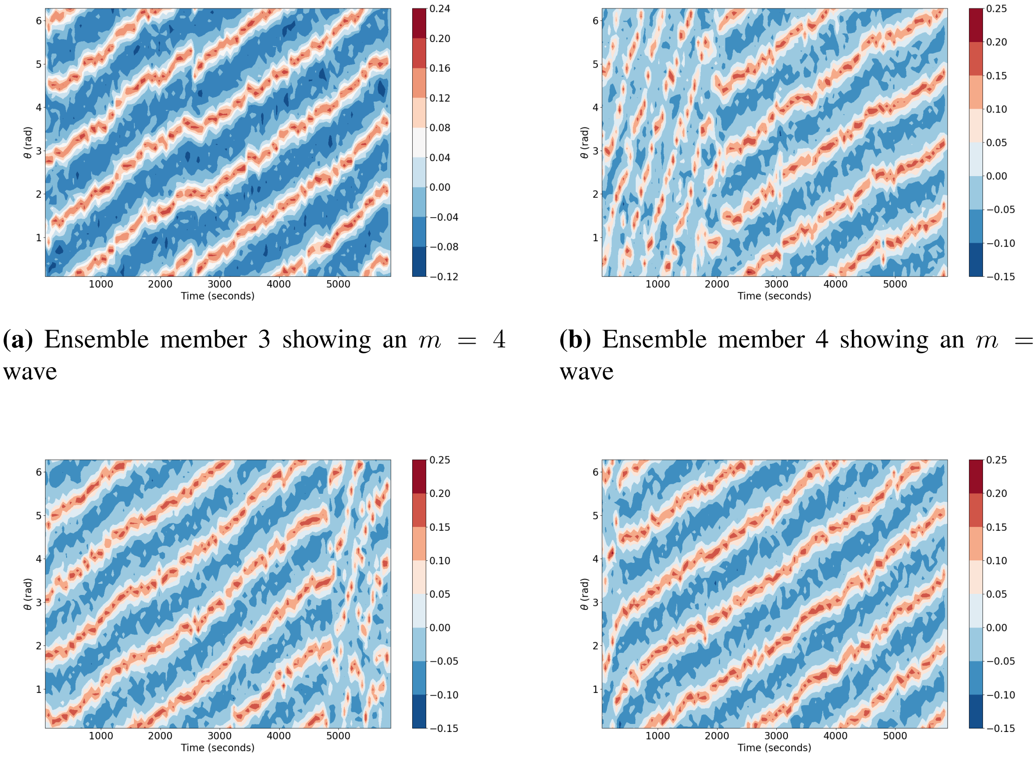
In all the studies there seems to be a good spread of the ensembles at the end of the run. Interestingly similar standard deviations are produced for the different rotation rates indicating that changing the rotation rates in the system does not affect the errors significantly. The high resolution runs show well-resolved structures for the different simulations which will work well when used for DA. The low resolution ensemble show a range of behaviour at higher rotation rates with differing drift rates and wavenumbers.



(a) The left panel shows a contour plot of the U velocity for the ensemble average at the end of the run at $r = 5.25$ cm.. The right panel shows the same for the high resolution run.

(b) The graph shows the U velocity across the annulus. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

Figure 7.12: Results for 3 rad/s when no data assimilation is done at $z = 7.0$ cm.



(a) Ensemble member 3 showing an $m = 4$ wave

(b) Ensemble member 4 showing an $m = 4$ wave

(c) Ensemble member 5 showing an $m = 4$ wave which collapses towards the end

(d) Ensemble member 6 showing an $m = 4$ wave

Figure 7.13: Hovmöller plot at 3 rad/s for ensemble members 2,3,4 and 5 showing the diversity of behaviour observed at this velocity.

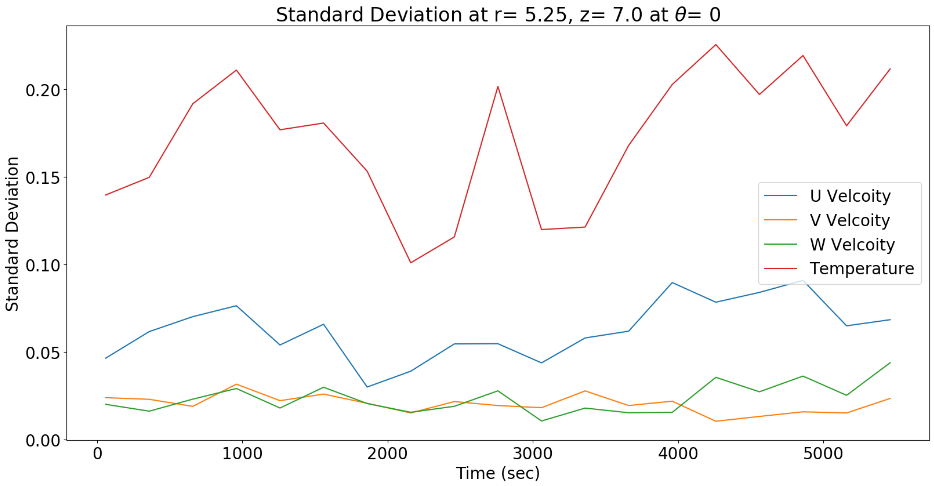


Figure 7.14: Standard deviation at 3 rad/s with no data assimilation.

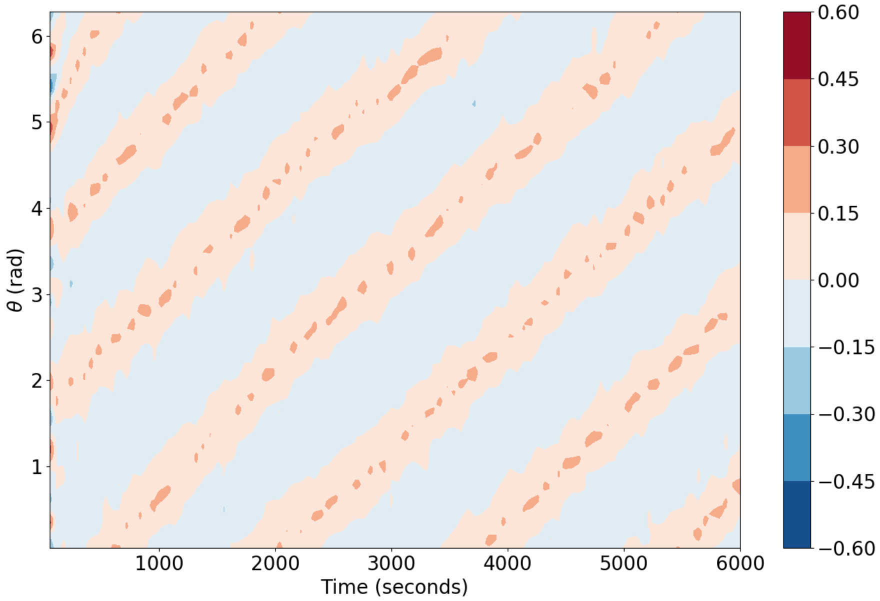


Figure 7.15: Hovmöller plot for the high resolution truth at 3 rad/s.

7.5 Error

For the optimal correction in EnKF, errors for the ensemble and the 'truth' are needed. The ensemble error is calculated using the covariance matrix but the error for the 'truth' is usually provided by the user. In an experiment, this error is estimated by considering the error in the instruments used to measure the variables (eg. instrument sensitivity in satellite observations) but in a twin setup experiment, it is not easy to calculate the error because the observations come from a computer simulation run rather than a physical experiment.

In our twin experiment, errors are needed for the velocity components, U, V and W and temperature, T. At the beginning of the study, the error was just taken as the maximum variation in the value of the variables for rotation at 1 rad/s. This is calculated as $R = 0.15, 0.2, 0.17, 0.35$ for U, V, W and T respectively (referred to as minmax), see table (7.3). These were used just for the initial tests before deciding on a better way to calculate the error.

A more robust error estimate for the 'truth 'was found by looking at the standard deviation for an ensemble of 10 rotating at 1 rad/s when no data assimilation is done (figure 7.2 in section 7.4.1). Over a 100 minute run, the standard deviation of each variable was found not to vary much. This was true even when the rotation rates were increased. This, therefore, gives a simple way of estimating the errors, referred to as 'pert '. These 'pert 'error values were found to be $R = 0.08, 0.03, 0.03, 0.2$ for U, V, W and T respectively. A majority of the simulations were done using double these values

Name	U Velocity	V Velocity	W velocity	Temperature
minmax	0.15	0.20	0.17	0.35
pert	0.08	0.03	0.03	0.20
2pert	0.16	0.06	0.06	0.40

Table 7.3: Errors used for the twin experiment

and were referred to as '2pert 'in the studies presented here as experimentation showed that this gave the best results: see below. A summary of the different errors is shown in table (7.3).

7.6 Results for 1 rad/s

In this section results for the annulus rotating at a velocity of 1 rad/s are presented for various forms of DA. At this speed, the flow is very stable and its structure is well known from many computational and experimental studies (Young and Read, 2013; Vincze, Harlander, von Larcher and Egbers, 2014; Vincze, Borchert, Achatz, von Larcher, Baumann, Hertel, Remmler, Beck, Alexandrov, Egbers, Froehlich, Heuveline, Hickel and Harlander, 2014). The flow at this rotation rate presents a $m = 3$ wave structure that drifts around the annulus. This makes it a good test bed to try out different approaches in EnKF and see how the data assimilation system adapts to the different scenarios.

7.6.1 Mid Results

To start the study a simple setup was used, with only a small number of grid points at the middle of the model being observed. As discussed in section (7.3.2), the middle nine points of θ were observed giving a total of 576 grid points. With 4 variables (U, V, W velocity and Temperature) for each point, this gives an observation vector with $p = 2304$ components. This is a very small number compared to the total number of point that are available in an ensemble $24 \times 24 \times 64 = 36864$ grid points giving $n = 147456$ vectors as there are 4 variables for each point.

7.6.1.1 Initial Tests

To begin the study, DA is applied to a system where all ensemble members are corrected, as represented by equation (7.1). No multiplicative inflation is used during the DA and after correcting and no perturbation (additive inflation) is added to the system. The run continues as normal after DA.

The model is run with a time step of $dt = 0.01$, assimilating data every 300 seconds (5 minutes). The system was run for 100 minutes which gives 19 times where EnKF was used. The studies were conducted using an observational error of $R =$

$0.15, 0.2, 0.17, 0.35$ for U, V, W and T respectively, reasons for which were discussed in section (7.5). An ensemble of 10 runs was used for the study. At each data assimilation, we use the formula

$$X^a = X^f + K \cdot (Y - HX^f). \tag{7.1}$$

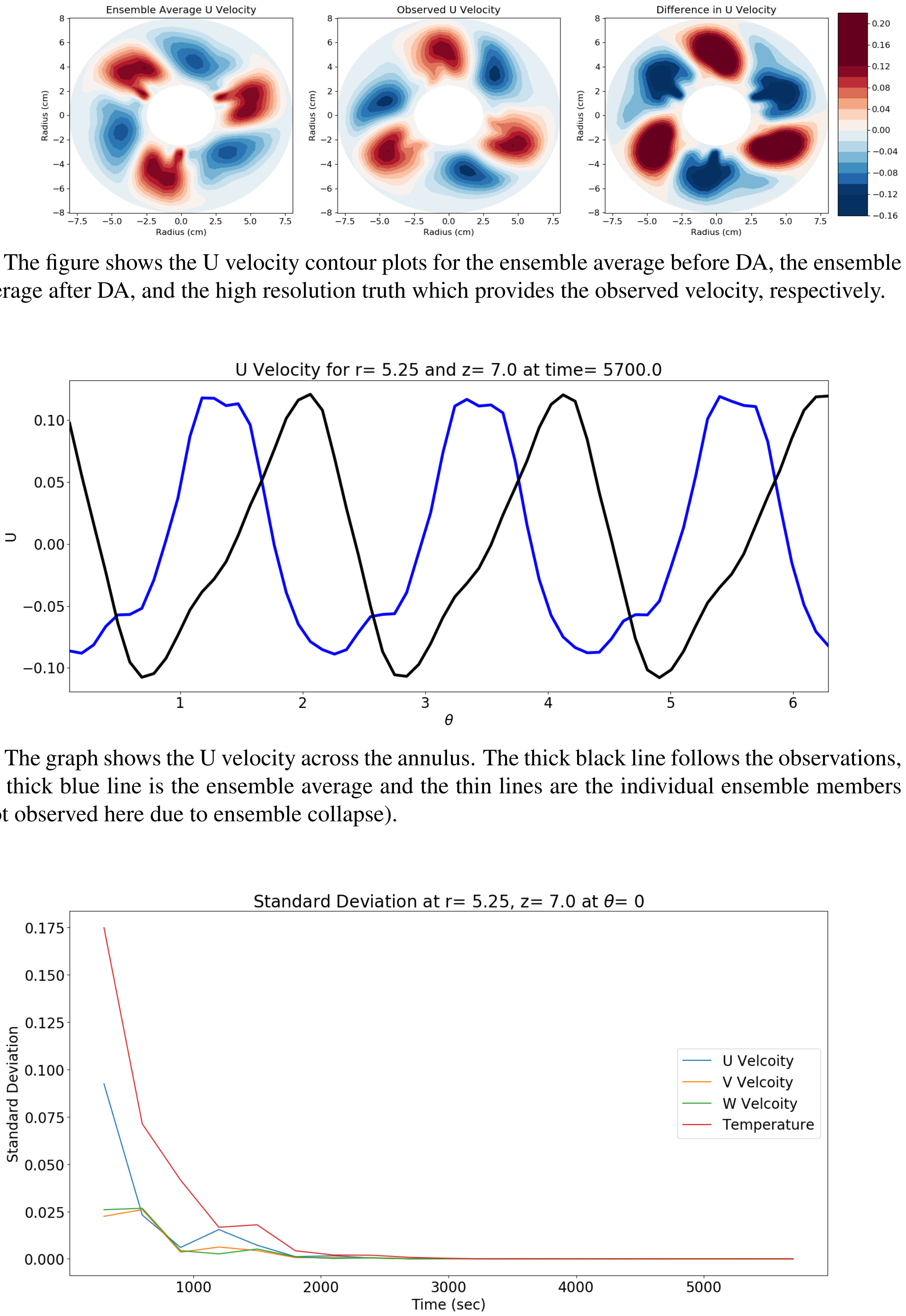
Here X^f is the forecast matrix which has the ensemble ready for DA, Y is the observation matrix, H is the observational matrix which maps the forecast to the observation. K is the Kalman gain, the weight which is used to correct the ensemble. X^a is the corrected matrix which contains the corrected ensemble.

The results for the simulation at the end of the run along with the standard deviation are shown in figure (7.16). A snapshot of the system is shown in figure (7.16a) for $z = 7.0$ cm. The first contour plot shows the ensemble average before any data assimilation is done. The second contour plots shows the observed truth that is used as the observation to correct the ensemble. The third and final contour plot shows the difference between the ensemble and the observation before data assimilation is done. The first plot shows the a well-defined rotating $m = 3$ structure found at 1 rad/s. This is very different from what was observed previously when no DA was done (figure 7.1a), here DA has pulled all the ensembles together which results in a very clear average contour plot. Since all members of the ensemble are behaving similarly and have converged, DA is not able to correct the solution back towards the truth. This can be clearly seen in the third contour plot where the difference in U velocity can be clearly observed, if the ensembles are truth were closer the contours will be more obscure. The ensemble members need to be spread out more to avoid ensemble collapsing on to an incorrect solution.

This divergence between the ensemble and the truth can be seen in figure (7.16b), here the graph shows the U velocity of the fluid in the middle of the annulus at $z = 7.0$ cm and $r = 5.25$ cm. The thick black line is the observed U velocity and the thick blue line is the ensemble average. Due to the collapse of the ensemble no thin lines of the individual ensemble can be observed as was seen before in figure (7.1b) when no DA was done. The thin blue lines all lie underneath the thick blue line. It is also clear from this plot that the ensemble members are drifting out of phase with the truth, and are unable to get back in phase with it.

The standard deviation for the system is shown in figure (7.16c). Here the errors have quickly collapsed close to zero for all the variables further confirming the convergence and that there is no spread in the ensemble. This is a big contrast to section (7.4.1) where no DA was done and each variable displayed its own unique spread throughout the run.

Figure (7.17) shows how the ensemble system evolves as the simulation progresses. At the beginning of the run, the ensemble does a good job of tracking the observation. But as the simulation goes along the ensemble starts to lag behind the observation, drifting further and further away. This shows that the drift rate of the ensemble is clearly slower than the observation. This difference in drift rate is something which was observed before in section (7.4.1) when no assimilation was done. What is also interesting to note is that after DA the position of the ensemble average does not change. This indicates that the correction being done by EnKF is suboptimal.

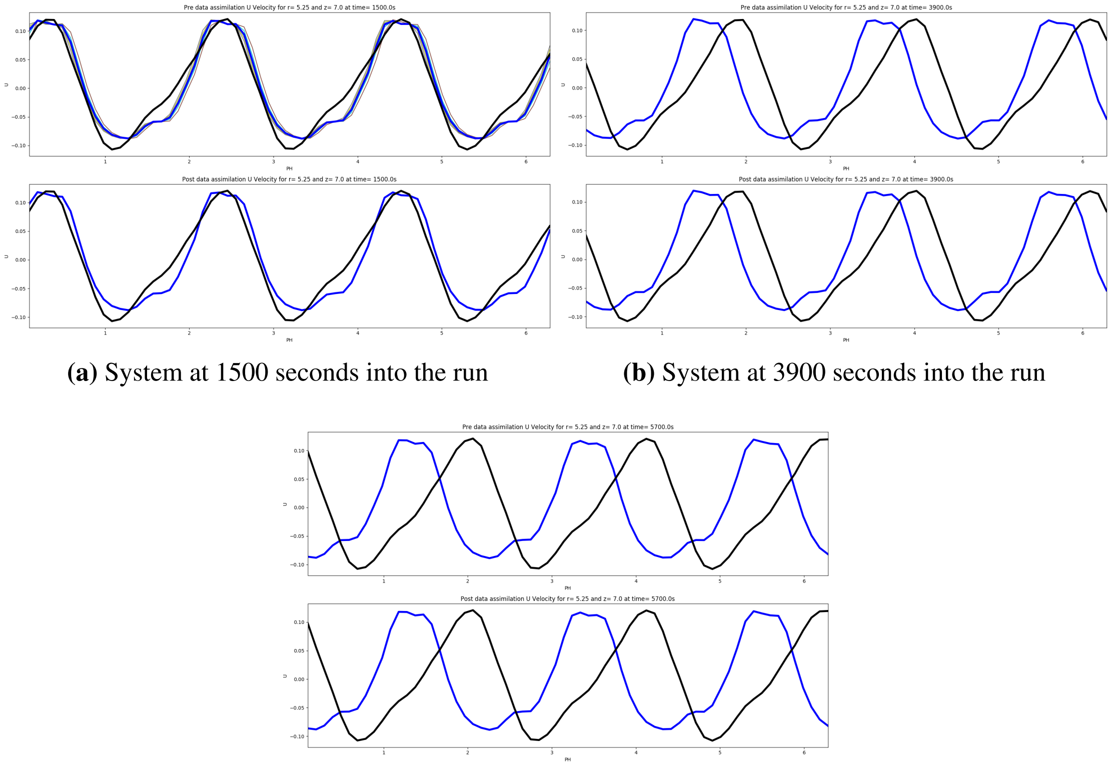


(a) The figure shows the U velocity contour plots for the ensemble average before DA, the ensemble average after DA, and the high resolution truth which provides the observed velocity, respectively.

(b) The graph shows the U velocity across the annulus. The thick black line follows the observations, the thick blue line is the ensemble average and the thin lines are the individual ensemble members (not observed here due to ensemble collapse).

(c) Standard deviation for data assimilation for when all members for ensemble are corrected for a rotation of 1 rad/s.

Figure 7.16: Data Assimilation results when all members of the ensemble are corrected for a rotation of 1 rad/s. No inflation was used in this runs.



(a) System at 1500 seconds into the run

(b) System at 3900 seconds into the run

(c) System at 5700 seconds into the run

Figure 7.17: System drift for 1 rad/s as the run progresses when no inflation is used. In each figure the top graph is the system before DA and the bottom graph is the system after DA. The thick black line follows the observations, the thick blue line is the ensemble average and the thin lines are the individual ensembles. In the later runs the ensemble members are hidden under the thick blue line as the ensemble spread is so small. As observed, the ensemble lags behind the observation during DA indicating it has a slower drift rate.

7.6.1.2 Multiplicative Inflation

To avoid the collapse of the ensemble onto an incorrect solution, a natural approach is to impose some type of inflation in the DA process. To study the impact of multiplicative inflation on DA the covariance of the ensemble is multiplied by a number/factor to artificially inflate/spread the error in the ensemble. After the data has been corrected, an additive inflation using a Gaussian perturbation with a mean of 0.0 and standard deviation of 0.01 is also added to the ensemble. Apart from this, the runs had the same parameters as used in the previous section. More information about multiplicative inflation can be found in section (4.7.2).

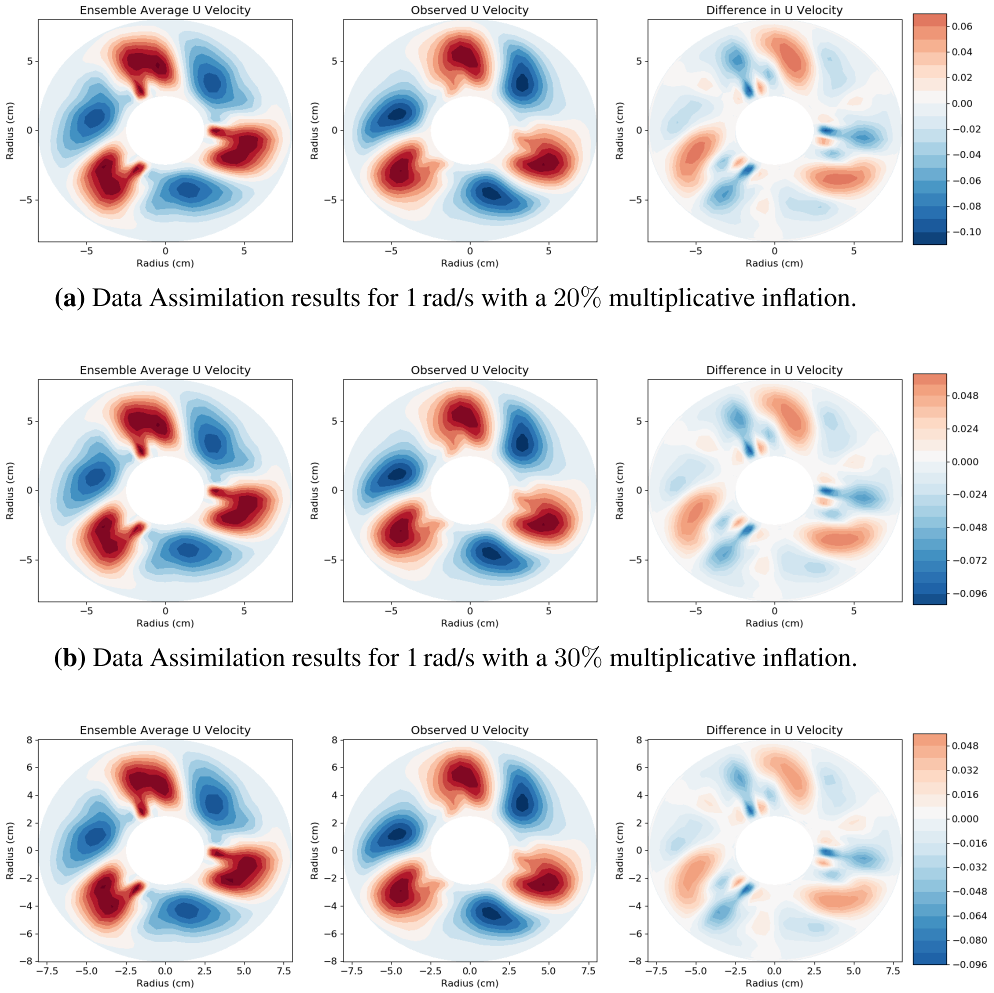
Runs were done with multiplicative inflation of 20%, 30% and 40% with figure (7.18) plotting the results for the contour plots at the end of the runs. In all three cases the $m = 3$ structure can be observed for both the ensemble average and the truth. The inflation has helped to better track the observations when compared to the initial test done in section (7.6.1.1), where no inflation was applied. The ensemble average stays much closer to the observation with multiplicative inflation, but as the inflation was increased, there was no marked progress in the accuracy of tracking the observation. As observed in the difference contour plot in all three cases, the difference between the ensemble and the truth seems small, with the ensemble average seeming near identical in all three cases. When checked, all three cases started with different initial conditions but it seems that with the use of DA all ensembles ended up in the same state-space giving the near identical structures.

This homogenising of the ensembles given enough time can also be observed in figure (7.19). This graph shows how the difference/gap between the ensemble average and the truth evolves through time. As observed in the figure, after the gap minimises around the 2000 seconds mark there is a steady increase in the difference as the run continues. This is because the ensemble is starting to drift away from the truth and EnKF is not able to correct the ensemble optimally.

7.6.1.3 Additive Inflation

With the results in the previous section indicating that multiplicative inflation cannot fully solve the problem of tracking the observation, a new approach is needed. In this section, we present results for additive inflation. In this method, a perturbation is added to the ensemble after DA, this spreads the ensemble making it less likely that the system will converge. The spread should lead to better covariance matrix leading to a more accurate Kalman gain and correction. More information about additive inflation can be found in section (4.7.2).

As observed in section (7.4) the standard deviation for each variable is different as each variable has distinct behaviour. This means that the perturbation needs to be tailored for the variables as well. Hence unlike the previous study where a Gaussian of mean 0.0 and a standard deviation of 0.01 was added to all variables, for this study the standard deviation calculated in section (7.4.1) was used. This is given as



(a) Data Assimilation results for 1 rad/s with a 20% multiplicative inflation.

(b) Data Assimilation results for 1 rad/s with a 30% multiplicative inflation.

(c) Data Assimilation results for 1 rad/s with a 40% multiplicative inflation.

Figure 7.18: Data Assimilation results for 1 rad/s with multiplicative inflation. The figure shows the U velocity contour plots for the ensemble average before DA. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

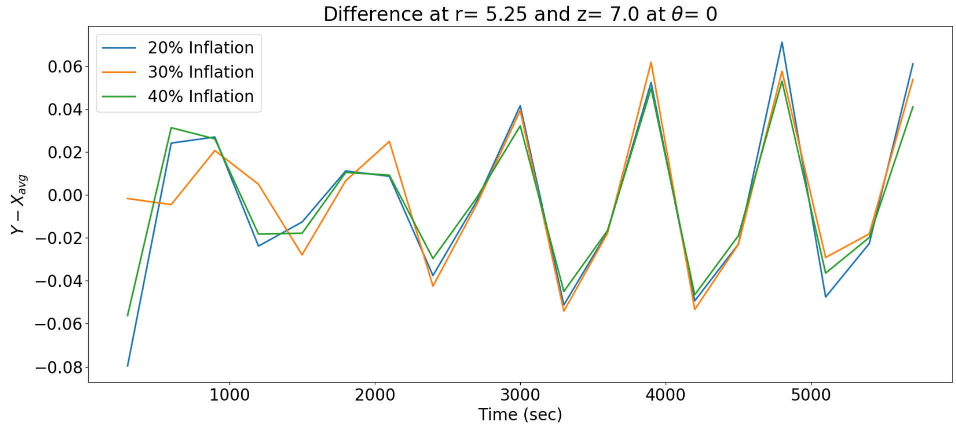


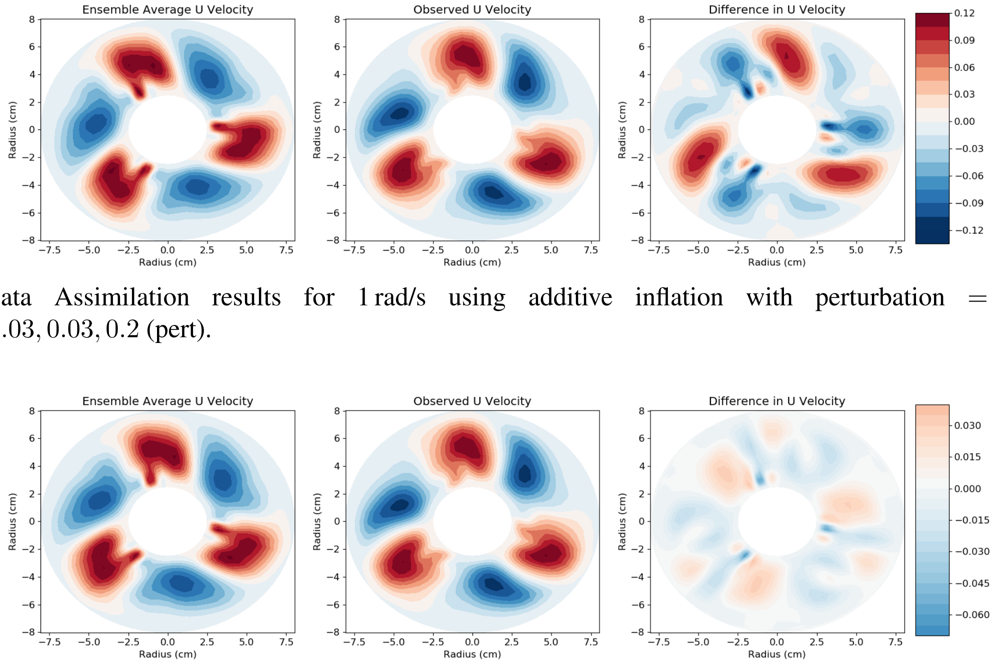
Figure 7.19: The difference between the ensemble average and the observations when using 1 rad/s and different multiplicative inflation

$0.08, 0.03, 0.03, 0.2$, for the variables U, V, W and T respectively, and will be referred to as ”pert” for the rest of the study. These values are used as the standard deviation around which a Gaussian of mean zero is created for the variables. These values were also used as the error in the respective observed variables.

The simulation is run using the same parameters as before (i.e. timestep = $0.01s$, ensemble size=10, etc) and data assimilation is done at every 300 seconds. The results for the run using pert are shown in figure (7.20a). The results are better than when a Gaussian of 0.01 standard deviation was used in section (7.6.1.1), and similar to the cases where multiplicative inflation was used on the system in section (7.6.1.2). These results show how important it is to have the correct perturbations for the different variable. The U velocity contour plots look similar to the ones observed for multiplicative inflation, with the pre-DA ensemble seeming to collapse. The difference between the ensemble and the truth is larger then before, with the ensemble seemingly behind the truth.

The next step for the study is to just double the perturbation i.e. 0.16, 0.06, 0.06, 0.4 (hereafter referred to as ”2pert”). Running this simulation, the ensemble now does a great job of tracking the observation as seen in figure (7.20b). The proximity of the ensemble and the truth can be clearly observed when looking at the difference between the two in the last contour plot. The difference observed here are very small with most values being close to zero, signifying that the ensemble average is nearly on top of the truth.

The success of 2pert can be clearly observed in figure (7.21), which shows the difference between the ensemble average and the truth throughout the run. Here, while the gap increases in the run where pert was used, the difference for 2pert decreases and remains close to zero. In both studies the difference are similar around the 2000 seconds mark but while one increases the other remains stable. This result shows the importance of having the correct perturbation for inflating individual variables. It is encouraging to see that EnKF can regularly track the observation at the rotation rate of 1 rad/s. In



(a) Data Assimilation results for 1 rad/s using additive inflation with perturbation $= 0.08, 0.03, 0.03, 0.2$ (pert).

(b) Data Assimilation results for 1 rad/s using additive inflation with perturbation $= 0.16, 0.06, 0.06, 0.4$ (2pert).

Figure 7.20: Data Assimilation results for 1 rad/s with additive inflation. The figure shows the U velocity contour plots for the ensemble average before DA. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

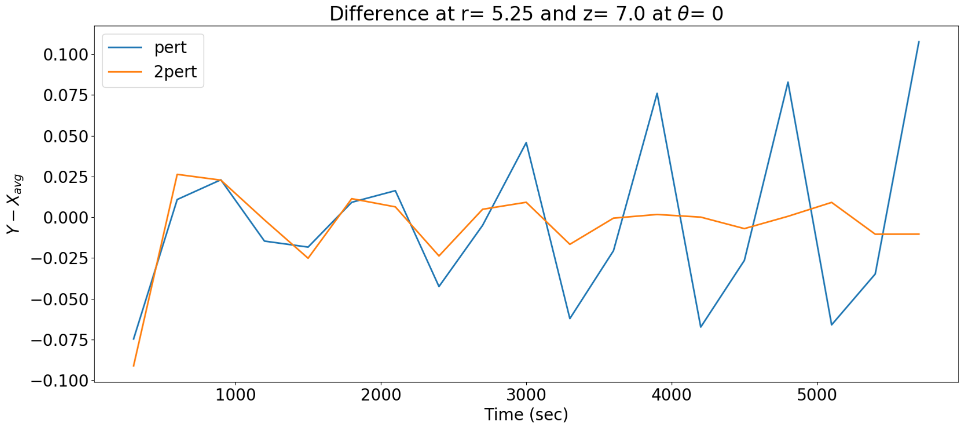


Figure 7.21: The difference between the ensemble average and the observations when using 1 rad/s and different additive inflation.

the following sections, EnKF will be pushed further to see how it behaves in different situations using these perturbation values. In view of this, 2pert will be used for all the other studies unless specified.

7.6.1.4 The Ensemble Size

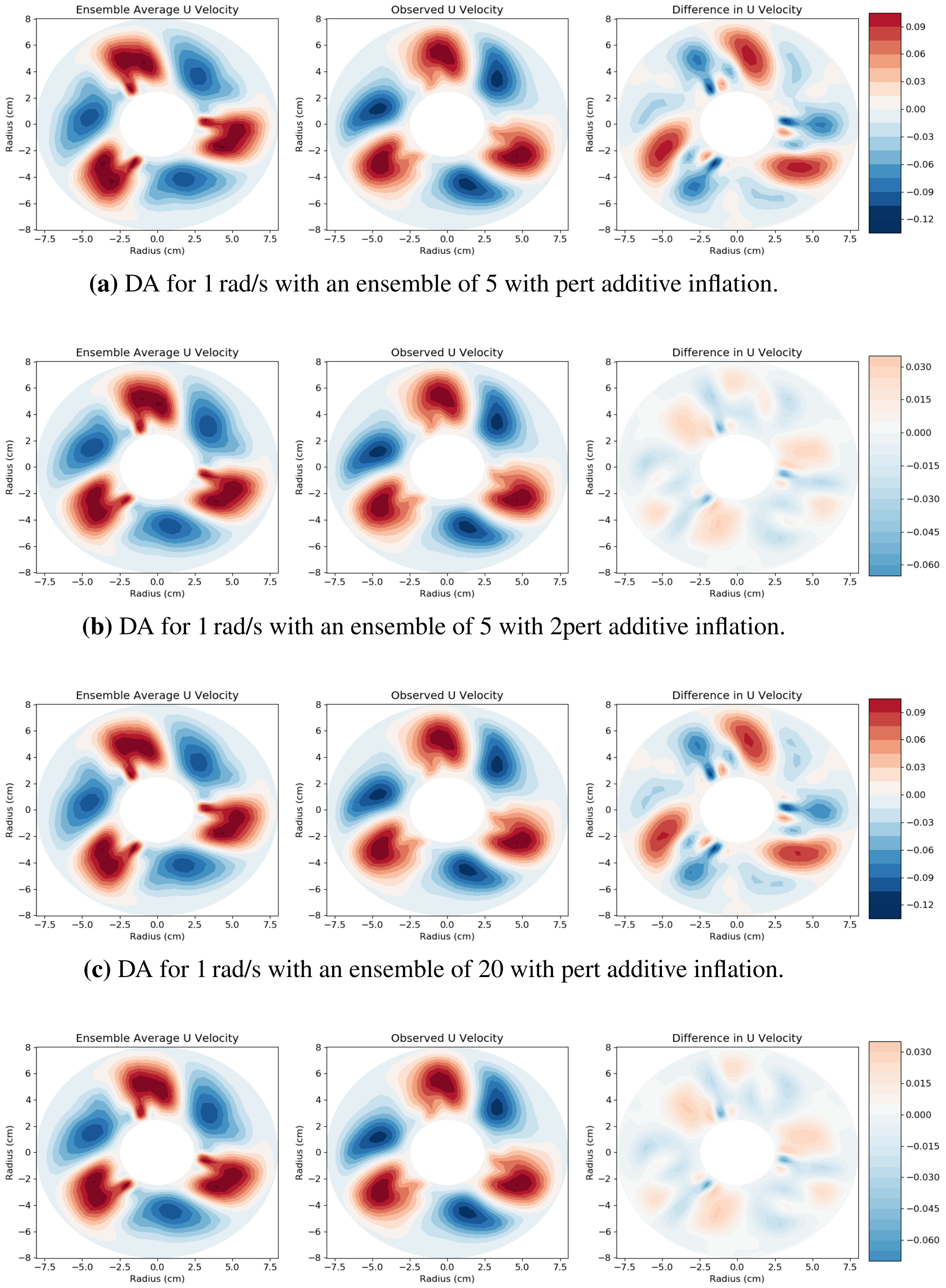
Having a suitable ensemble size is very important when using EnKF (section 4.7 for more details). A small ensemble size leads to under sampling of the state space leading to suboptimal correction, while a large ensemble size leads to a better sampling of the state space but needs longer simulation time for the extra ensembles, negating the usefulness of using low resolution ensembles. Hence a compromise needs to be made to get an optimal ensemble size which can be used to get fast and accurate results. In this section, results are presented for different ensemble sizes using additive inflation of pert and 2pert.

In the previous sections, an ensemble of 10 was used to study the effectiveness of EnKF. Through testing, it was concluded that using an additive perturbation of 2pert, the system does a very good job of keeping the ensemble from drifting away from the observation. In the results presented in this section, additive perturbations of pert and 2pert are used to show how the different ensemble sizes compare. In all the studies a time step of $dt = 0.01$ is used, with a run time of 100 minutes, and DA is done every 5 minutes (300 seconds). Studies were done for an ensemble size of 5, 20 and 30 and figure (7.22) and (7.23) show the contour plots for these runs.

In the first study, an ensemble of 5 (5-pert) was used with additive inflation of pert to study EnKF. The snapshot of the system before the last DA is shown in figure (7.22a) and show distinct structures for the ensemble average indicating that the ensembles have converged. The difference contour plot shows that the two system are close but the ensemble is lagging behind the truth. The additive inflation was increased to 2pert (5-2pert) for the second study with the results plotted in figure (7.22b). Distinct structures are again observed in the contour plots for the ensemble average showing that the change in inflation does to diverge the ensembles. The two system appear very close to each other which is confirmed when looking at the difference plot.

As ensemble of 10 has been already studied in the previous section, the next study was of a larger ensemble size of 20 to test EnKF. After running the simulation with the additive inflation pert (20-pert), the state of the system at the last DA is shown in figure (7.22c). Surprisingly, increasing the ensemble size has not changed the effectiveness of EnKF significantly and the results seem similar to the results when an ensemble of 5 or 10 was used with an additive inflation of pert. Increasing the additive inflation to 2pert (20-2pert) for the second test has helped the ensemble track the observations as observed in figure (7.22d). These results are again similar to what was observed with the smaller ensembles of 10 and 5 with the higher inflation.

For the final study the ensemble size was increased further to 30, and two studies were again conducted for additive inflation of pert (30-pert) and 2pert (30-2pert). The results for pert are plotted in figure (7.23a) and the results for 2pert are plotted in fig-



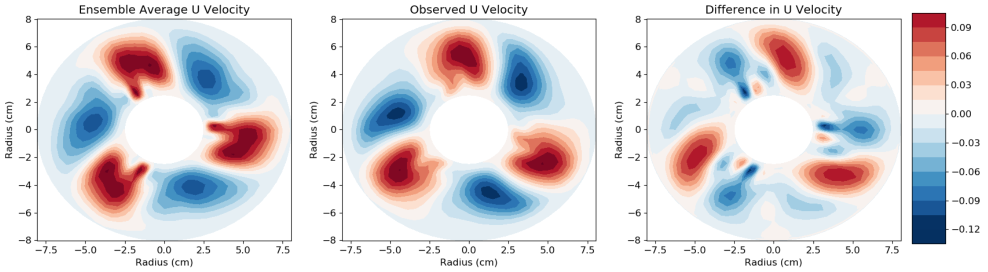
(a) DA for 1 rad/s with an ensemble of 5 with pert additive inflation.

(b) DA for 1 rad/s with an ensemble of 5 with 2pert additive inflation.

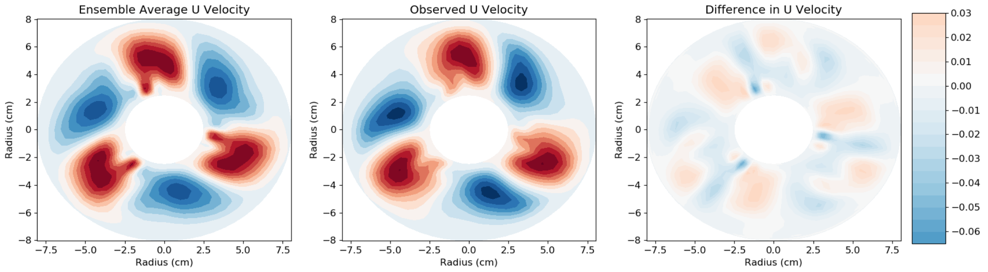
(c) DA for 1 rad/s with an ensemble of 20 with pert additive inflation.

(d) DA for 1 rad/s with an ensemble of 20 with 2pert additive inflation.

Figure 7.22: Data Assimilation results for 1 rad/s with additive inflation and different ensemble sizes. The figure shows the U velocity contour plots for the ensemble average before DA. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.



(a) DA for 1 rad/s with an ensemble of 30 with pert additive inflation.



(b) DA for 1 rad/s with an ensemble of 30 with 2pert additive inflation.

Figure 7.23: Data Assimilation results for 1 rad/s with additive inflation and different ensemble sizes. The figure shows the U velocity contour plots for the ensemble average before DA. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

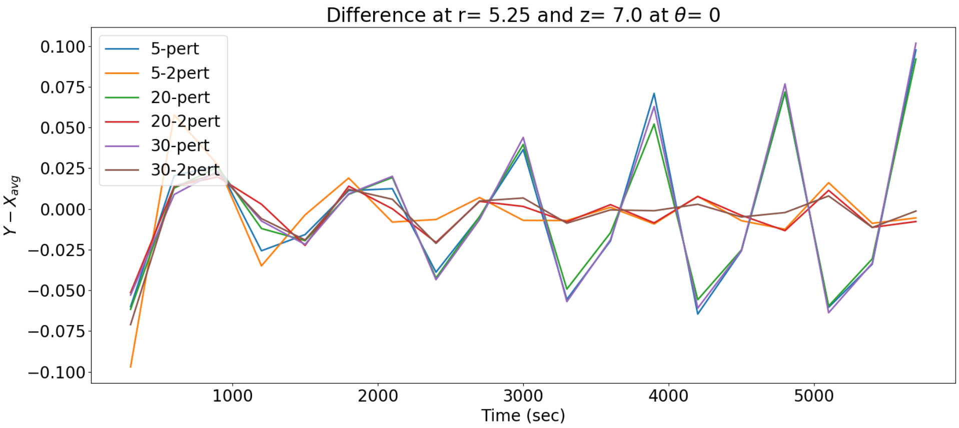


Figure 7.24: The difference between the ensemble average and the observations when using 1 rad/s and different ensemble sizes.

ure (7.23b). Both sets for results are very familiar when compared to what has been observed with an ensemble of 5, 10 and 20. The ensemble in the pert studies lag behind the observations and the ensemble track the observations well when the inflation of 2pert is used.

The effectiveness of using 2pert additive inflation can be observed in figure (7.24), where the stability of the system using this larger inflation can be clearly observed when compared to when only pert was used. The gap in 2pert system stays very small while the difference in the pert system increase as the runs propagates.

Overall increasing the ensemble size did not lead to a significantly better DA solution. Using the additive inflation of pert, the correction was still suboptimal for all ensemble sizes, with the ensemble average lagging behind the observation. The use of 2pert additive inflation solved this problem and all ensembles were able to track the observation.

The reason for the increase in ensemble size not leading to a more accurate DA could be because the results for 1 rad/s are very stable and hence do not have enough variety in drift rate and wavenumber. So when the ensemble is corrected they all collapse and settle down on a particular mode, which then becomes difficult to change out of as the ensemble members all have a similar too-slow phase speed. This problem cannot be resolved by increasing the ensemble size, but it can be solved only by increasing the amount of additive inflation.

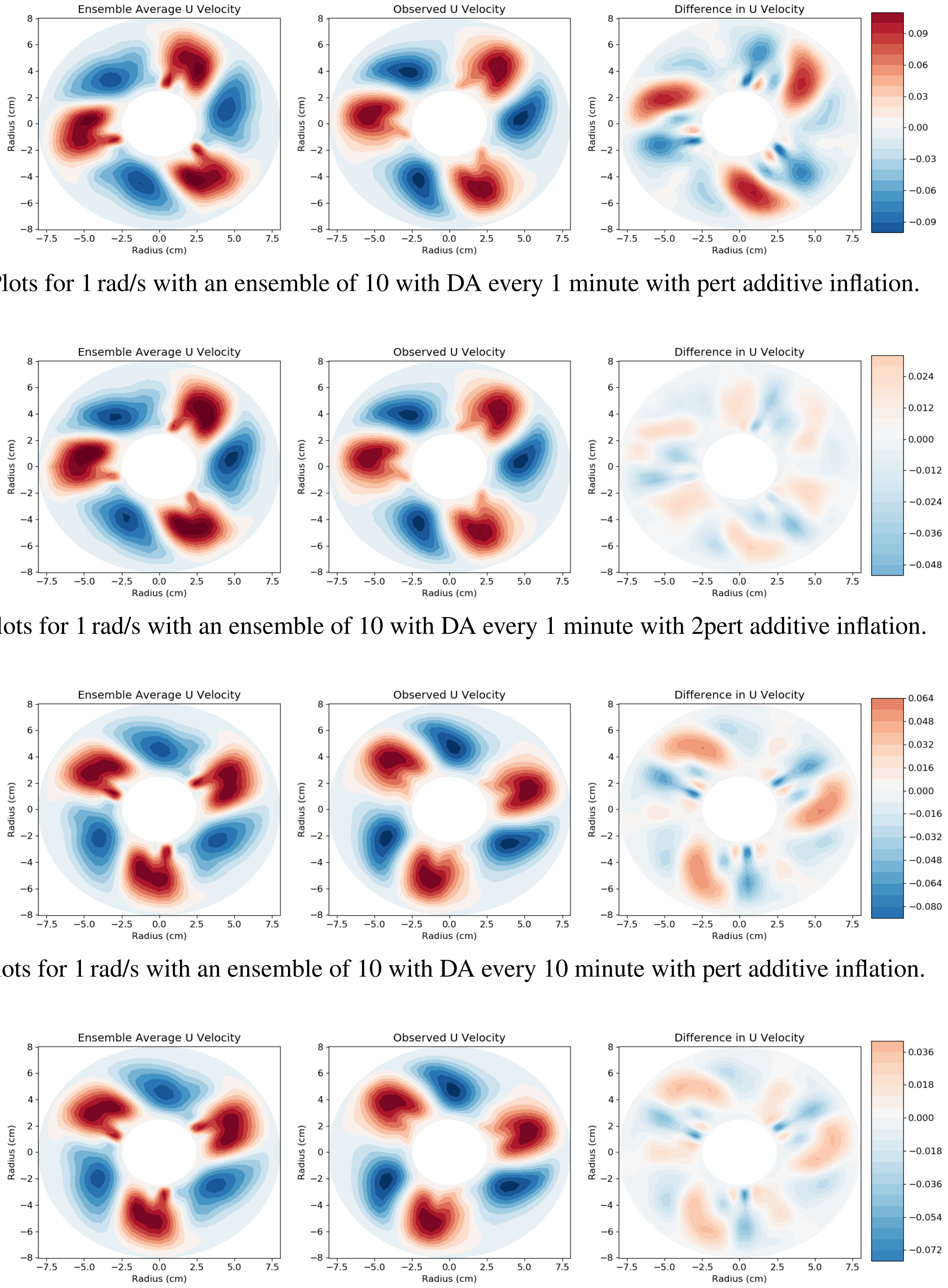
7.6.1.5 Changing DA times

The time between DA plays an important role on how accurate predictions can be obtained from the imperfect model. Assimilating frequently leads to a better prediction, but it can be impractical as it increases computational time and sufficient observations may not be available. Inaccurate predictions are especially true when inflation is not properly used as seen in the previous section. Leaving too long between DA gives more time for the ensemble to diverge. This means DA correction might be suboptimal leading to the ensembles not being on able to accurately predict the truth.

As previous section have presented results for DA every 5 minutes, in this section results are presented for studies where DA was done at every 1 minute and where DA was done every 10 minutes. In both studies additive inflation was used to see how the system behaves and the simulation was run for a total of 100 minutes with an ensemble of 10.

In the first study, the time taken between DA was reduced from 5 minutes to 1 minute. This means that for a 100 minute run DA was done 99 times. Using an additive inflation of pert (min1-pert), the results for the run are shown in figure (7.25a). Even with the shorter interval between DA, the ensemble with this smaller additive inflation cannot track the observation closely. The difference between the ensemble average and the truth is still large indicating that shortening the time between DA does not automatically lead to more accuracy.

The additive inflation was increased to 2pert (min1-2pert) for the second test to see



(a) Plots for 1 rad/s with an ensemble of 10 with DA every 1 minute with pert additive inflation.

(b) Plots for 1 rad/s with an ensemble of 10 with DA every 1 minute with 2pert additive inflation.

(c) Plots for 1 rad/s with an ensemble of 10 with DA every 10 minute with pert additive inflation.

(d) Plots for 1 rad/s with an ensemble of 10 with DA every 10 minute with 2.5pert additive inflation.

Figure 7.25: Results for an ensemble of 10 with DA at various times. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

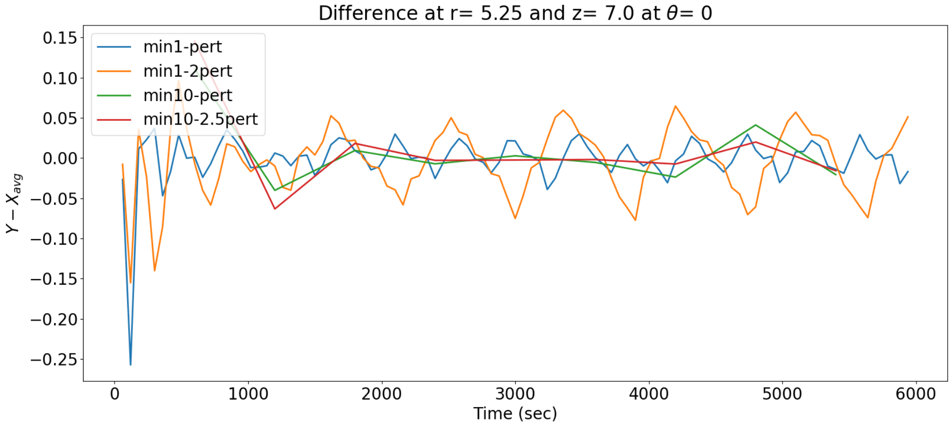


Figure 7.26: The difference between the ensemble average and the observations when using 1 rad/s and different DA times.

if the system can make better predictions. The results for the run are plotted in figure (7.25b), showing that with the higher additive inflation the system does a much better job of tracking the observations. The ensemble structure is very close to the truth as seen in the difference contour plot.

For the second set of studies DA was done every 10 minutes, so for a 100 minute run DA was done only 9 times. As with when DA was done every 1 minute, the first test uses an additive inflation of pert (min10-pert). The results for the run are shown in figure (7.25c). With the longer time between DA, the ensemble has a lot of time to diverge but looking at the contour plot it seems that DA has done a good job of keeping the ensembles around a similar state space. This behaviour might be unique to the results at 1 rad/s as the more chaotic flow observed at higher rotation rates should cause the ensemble to diverge at these long gaps between DA. Overall the ensemble is not able to track the observation perfectly, falling slightly behind. This is not as bad as what was observed in figure (7.25a) when DA was done every minute.

For the second test, the additive inflation was increased to 2.5*pert (min10-2.5pert). The results for the run is shown in figure (7.25d). Even with the higher inflation, the tracking of the observation is still slightly behind as observed in the U velocity graph. It seems that given the long gap and the fact that the drift rate of the low resolution system is less then the high resolution run the system will always fall behind the truth even with higher inflation.

Figure (7.26) shows how the difference between the ensemble and the truth evolves in the runs. When DA was done every minute the 2pert study shows larger gap then pert. Both 1 minute runs seem stable but this again might just be due to the frequency of DA which is keeping the ensemble around the same state space. Looking the at the longer 10 minute gap, the difference looks very small indicating that the system remains very close to the truth. But this might be just coincidence that the ensemble drift rate and truth line up. A longer run might be needed to confirm this.

7.6.1.6 Changing the number of observed points

The number of observations used for DA can also have a big impact on the accuracy of EnKF. The higher the number of observations the better the correction. But a high number of observations means a bigger covariance matrix that needs to be solved. This can use up a lot of computer memory and slow the calculations down. A smaller number of observations reduces the computer load, but then EnKF needs to be tuned using inflation (multiplicative and additive), etc, to get an accurate prediction.

From the previous studies, it has become apparent that using EnKF at 5 minute intervals with an additive inflation of 2pert can lead to the ensemble system predicting and tracking the observations quite precisely. Using these settings, a new series of studies were done where the number of observed points was limited. In previous sections 576 grid points were observed with 4 variables being observed at each point. Limiting these observed points and variables will push EnKF and see if it will be able to track the observations with an even lower number of observations.

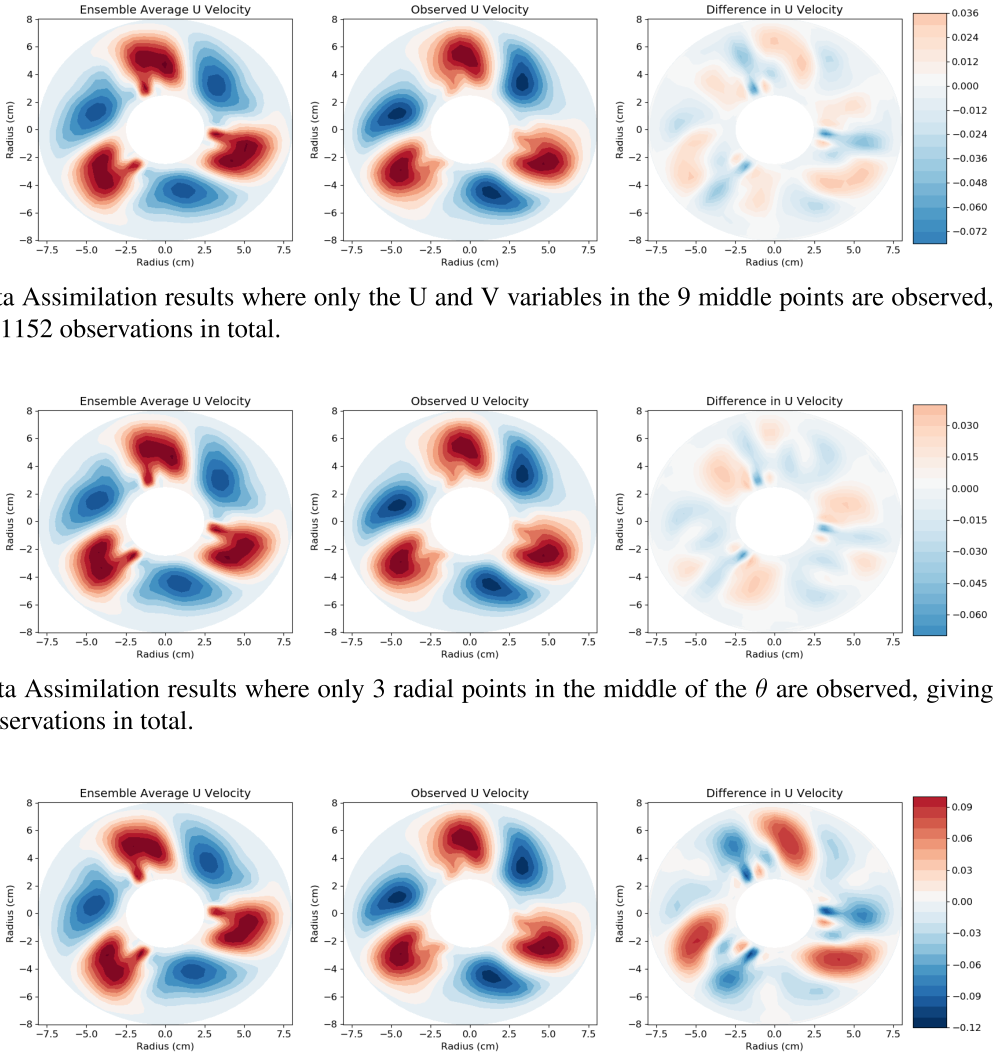
In the first study, the 9 middle points in the r and z directions for each θ-value are observed as usual giving 576 grid points but only the U and V variables are used for DA (P9-UV). This halves the number of components in the observation vector from 2,304 (9*64*4) in a normal study to 1,152 (9*64*2) components. The results at the end of the run can be viewed in figure (7.27a) which should be compared with the results in figure (7.20b). Looking the at the plots, the ensemble average is very close to the truth. This is encouraging as only half the variables are observed and no multiplicative inflation was applied.

In the next study, the number of observed points was decreased from 9 points in the middle of the θ-plane to just 3 middle points (P3) for each θ-value, i.e points (r_i, z_j) with $i = 11, 12, 13$ and $j = 12$. This decreases the number of observed points to 192, giving 768 observations altogether. Figure (7.27b) shows the results for this study. Even with the reduction in the observed points the system does very well at tracking the observation. This seems to indicate that observations in the middle of the model are important when it comes to correcting the model and tracking the observations.

The observations are reduced for the next study with only the U and V velocity variables used at the 3 radial points (P3-UV) for each θ-value giving only 384 observations to be used in DA. The results of this run is plotted in figure (7.27c). With fewer observations the system starts to struggle as seen in the difference contour plot. The differences is much clearer when compared to the previous two studies. This indicates that the ensemble is not tracking the truth accurately.

Bringing the observed points down even more to just the middle point (P1) in each θ-value i.e. the point (r_i, z_j) with $i = 12$ and $j = 12$. This gives only $4 \times 64 = 256$ observations that are used for DA. The system at the end of the run is shown in figure (7.28a). Even with only 1 point being observed the system does a better job when compared to the previous section where 3 points and only U and V velocity were observed. This seems to indicate that the observation of the different variables leads to better results.

In the last study the number of observations is cut to 1 point with only U and V

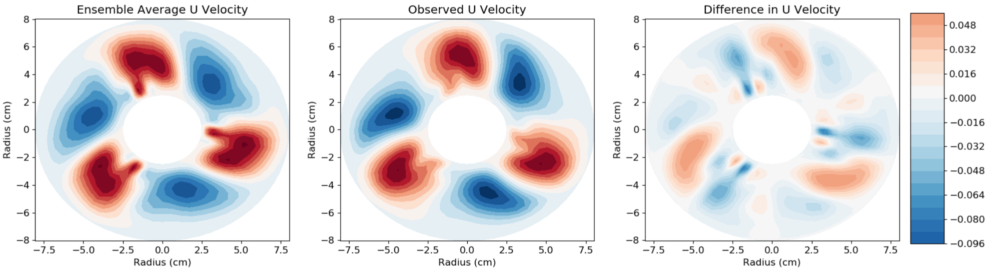


(a) Data Assimilation results where only the U and V variables in the 9 middle points are observed, giving 1152 observations in total.

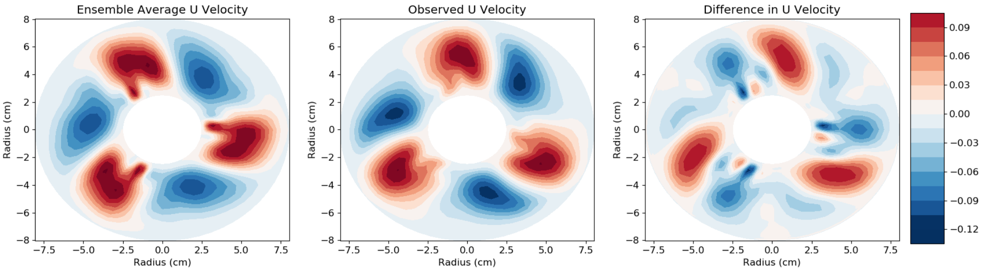
(b) Data Assimilation results where only 3 radial points in the middle of the θ are observed, giving 768 observations in total.

(c) Data Assimilation results where only U and V variables in 3 radial points in the middle of the θ are observed, giving 384 observations in total.

Figure 7.27: Results for an ensemble of 10 with DA using different number of observed θ. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.



(a) Data Assimilation results where only 1 radial points in the middle of the θ are observed, giving 256 observations in total.



(b) Data Assimilation results where only U and V variables in 1 radial point in the middle of the θ are observed, giving 128 observations in total.

Figure 7.28: Results for an ensemble of 10 with DA using different number of observed θ. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

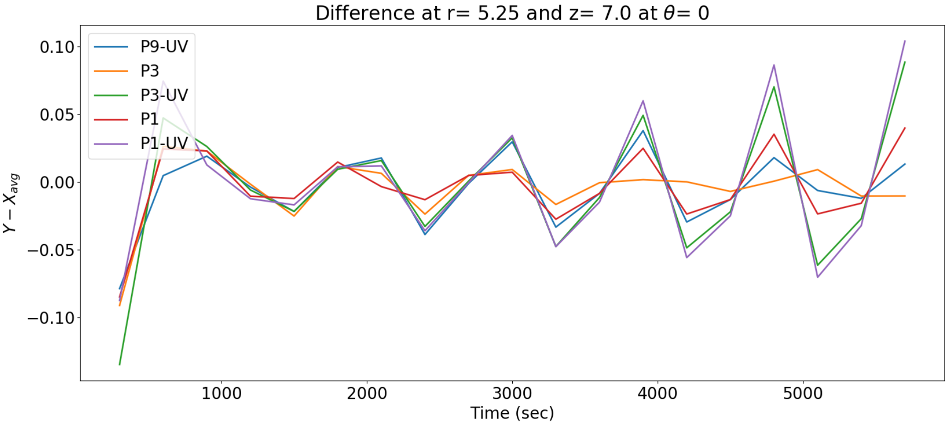


Figure 7.29: The difference between the ensemble average and the observations when using 1 rad/s and different observation values.

velocities (P1-UV) observed for every θ. This gives only 128 observations to be used for DA. The system at the end of the run is shown in figure (7.28b). With this low number of observations it is expected that the system will not do a good job of keeping track of the truth and looking at the difference plot this seems to be the case. The ensemble is not able to track the truth accurately.

Looking at figure (7.29) and how the gap between the ensemble and the truth evolve, it's clear that observing all the variables leads to better tracking. All studies with where only U and V variables were used lead to an increase in the difference, whereas when all the variables were observed the difference remained stable. This stability was also true with lower number observed points.

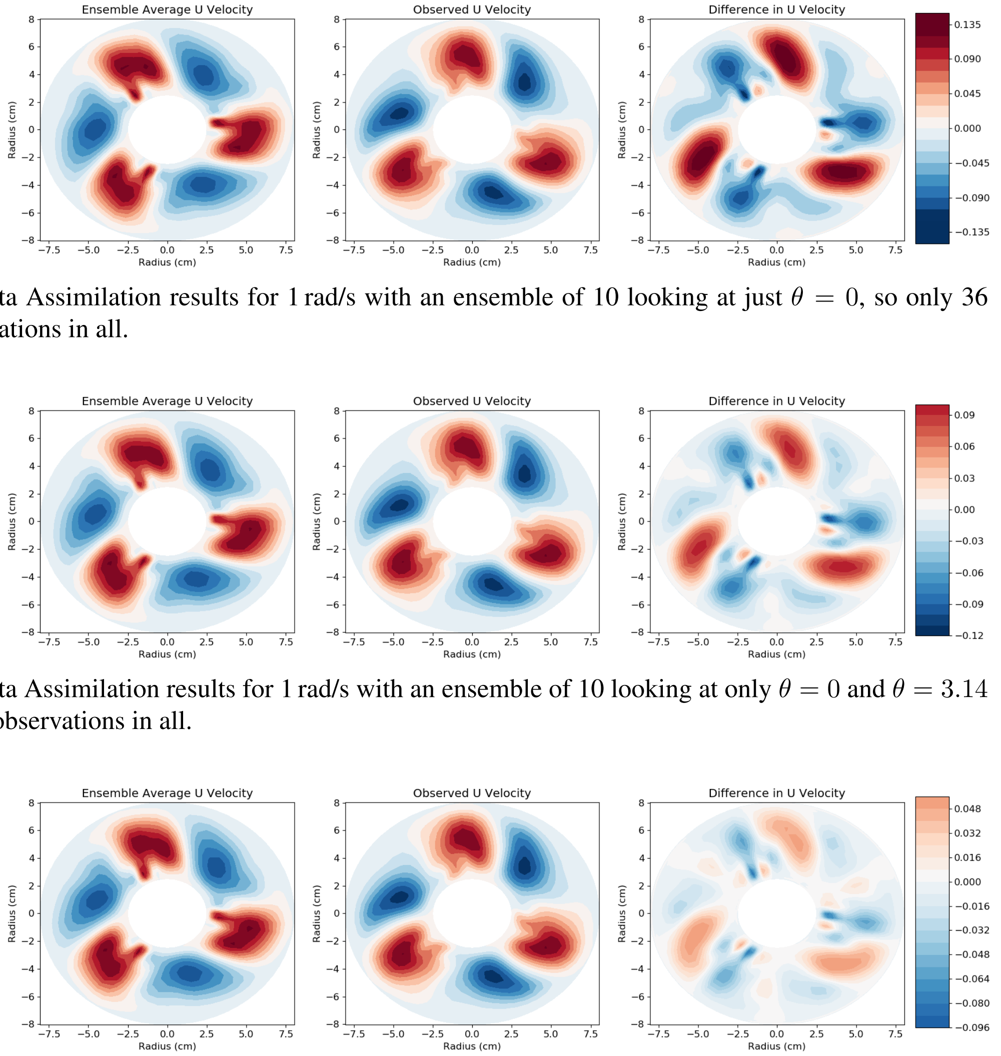
Overall as the number of observations were reduced, the accuracy of EnKF falls off. This is simply to do with the system not having enough information to give an accurate correction to the system. The results for the higher observation numbers, P9-UV and P3, are still encouraging with the system being close to the truth. With fewer observations, it becomes clear that observing all the variables is important to give accurate predictions. A good example of this is seen when only 1 point was observed per θ, when all the variables were observed a better result was obtained compared to when the 3 points were observed per θ and only U and V velocities were used. Maybe at this lower end of observation numbers a more frequent DA would help the system get a more accurate prediction.

7.6.1.7 Changing the number of observed θ-values

An interesting way of attacking the problem of how much observation is needed for accurate prediction is to look at how much information around the azimuth of the model is needed to accurately predict the truth. In the previous studies the number of observed points and variables were changed, but in all cases points were observed all around the azimuth. In this section a similar study is done but now the number of observed θ points around the azimuth is changed. Two approaches are presented, the first where slices of θ are observed at different points along the model, and second where a continuous section of the θ is observed. In all studies presented in this section an additive inflation of 2pert was used.

The first study looked at just the 9 Mid points at $\theta = 0$ (One θ), so this gives only 36 observations (9*4) for DA. This is drastically low compared to any other study presented here. The results for this run are shown in figure (7.30a) and as expected with this low number of observations the system does not do well. As observed in the difference contour plot, the system lags behind the truth.

Continuing to use the 9 Mid points, in the second study, two θ positions (Two θ) are observed at $\theta = 0$ and $\theta = 3.14$, i.e. the beginning and middle θ in the model giving 72 observations for DA. The results of the run are shown in figure (7.30b). As expected, the system does better than in the previous case, where only 36 observations were used. The system is closer to the truth when compared to the previous study but still has some ways to go.



(a) Data Assimilation results for 1 rad/s with an ensemble of 10 looking at just $\theta = 0$, so only 36 observations in all.

(b) Data Assimilation results for 1 rad/s with an ensemble of 10 looking at only $\theta = 0$ and $\theta = 3.14$ so 72 observations in all.

(c) Data Assimilation results for 1 rad/s with an ensemble of 10 looking at $\theta = 0$, $\theta = 1.57$, $\theta = 3.14$ and $\theta = 4.7$ so 144 observations in all.

Figure 7.30: Results for an ensemble of 10 with DA using different number of observed θ. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

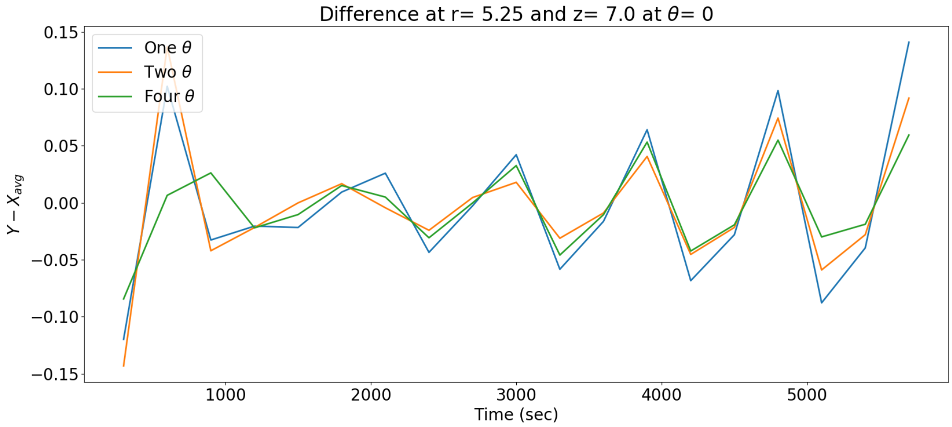


Figure 7.31: The difference between the ensemble average and the observations when using 1 rad/s and different θ slices.

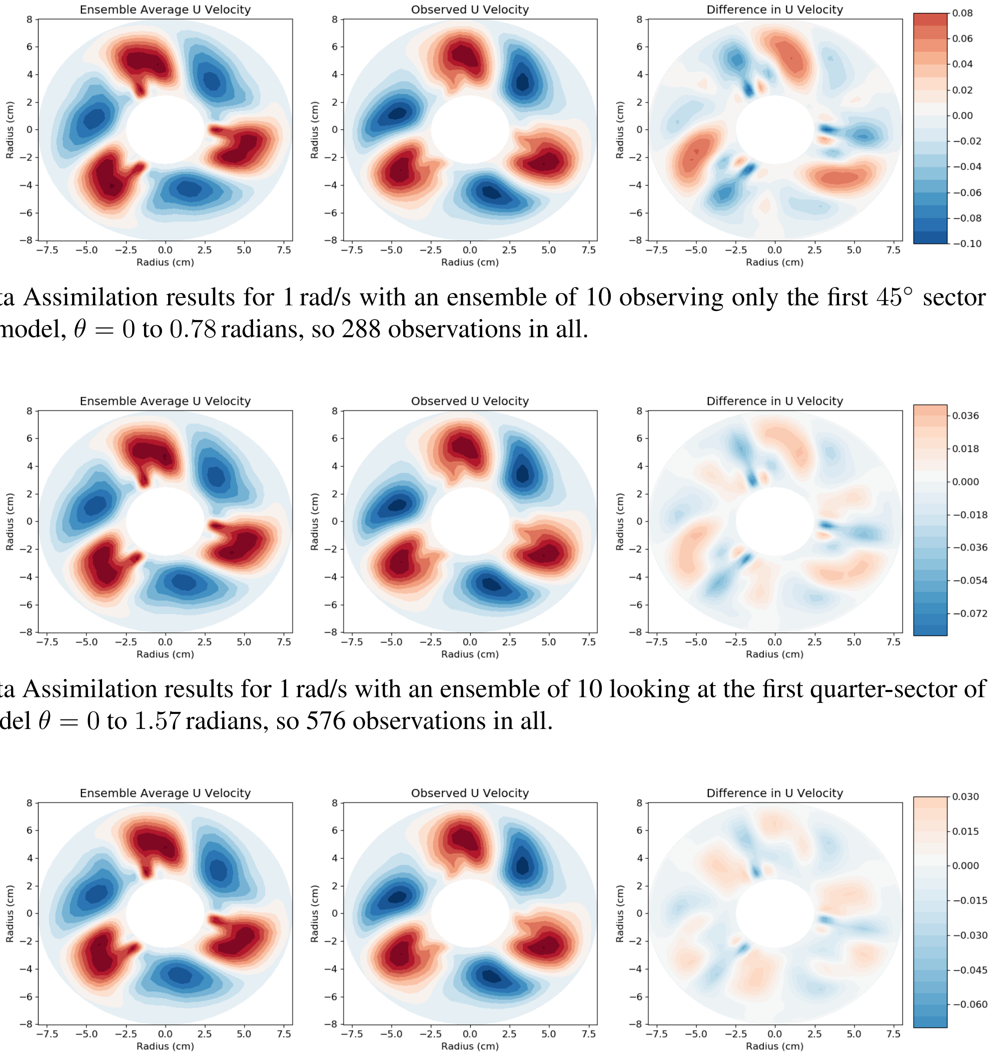
The observed number of θ was increased to 4 with $\theta = 0$, $\theta = 1.57$, $\theta = 3.14$ and $\theta = 4.7$ (Four θ) being observed i.e. every quarter around the model. This increases the number of points observed to 36 and so 144 observations (36) are to be used for DA. The results for the run are shown in figure (7.30c). As expected, the system has improved. With more points being used for observation, the correction from EnKF is becoming better.

The number of observations used here is equivalent to the P1-UV study where 128 observation was used for DA. The results here are better than what is observed in figure (7.28b) with the ensemble closer to the truth. This again seems to show how important it is to observe all the variables. But looking at the how the gap between the ensemble and the truth evolves in figure (7.31), it is clear that the system is not stable and the difference actually increases as the run propagates for all cases.

A second approach to testing how the choice of observation points θ can affect EnKF is to see how well a continuous sector of observations performs. Rather than distributing the θ points sparsely or regularly, we now use all the θ points that lie in a sector. In the first test a sector of the first eighth of the model is observed i.e. in $0 \leq \theta \leq 0.78$ radians (0-8 θ), again using the 9 Mid points at each θ-value, giving $8 \times 9 \times 4 = 288$ observations that are used for DA.

The results for this run are shown in figure (7.32a) and even with the higher number of observed points, the system gives similar results to when only 4 slices of θ were observed in the previous study. The ensembles are more spread than before but overall it seems that EnKF still cannot accurately predict the corrections using only a small part of the model. Observing other parts of the model should help with better prediction.

In the second test a quarter-sector of the model is observed i.e. $0 \leq \theta \leq 1.57$ radians (0-16 θ), giving 576 observations for the DA. The results for this run are shown in figure (7.32b). The system is now doing a good job of tracking the truth. It seems that the crucial difference between the $45°$ sector and the $90°$ sector is that the $90°$ sector



(a) Data Assimilation results for 1 rad/s with an ensemble of 10 observing only the first $45°$ sector of the model, $\theta = 0$ to 0.78 radians, so 288 observations in all.

(b) Data Assimilation results for 1 rad/s with an ensemble of 10 looking at the first quarter-sector of the model $\theta = 0$ to 1.57 radians, so 576 observations in all.

(c) Data Assimilation results for 1 rad/s with an ensemble of 10 looking at the first half of the model $\theta = 0$ to 3.14 radians, so 1152 observations in all.

Figure 7.32: Results for an ensemble of 10 with DA using different number of observed θ sectors. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

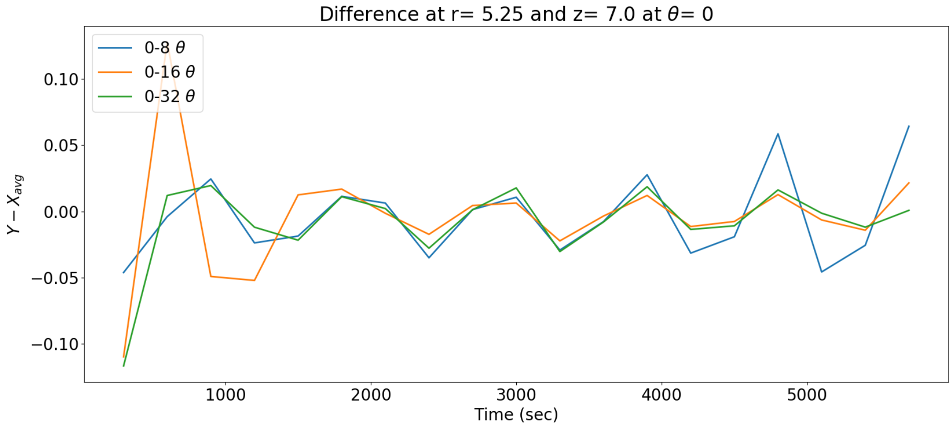


Figure 7.33: The difference between the ensemble average and the observations when using 1 rad/s and different θ sections.

captures almost a whole azimuthal wave, whereas the $45°$ sector could not locate the position of the wave sufficiently accurately.

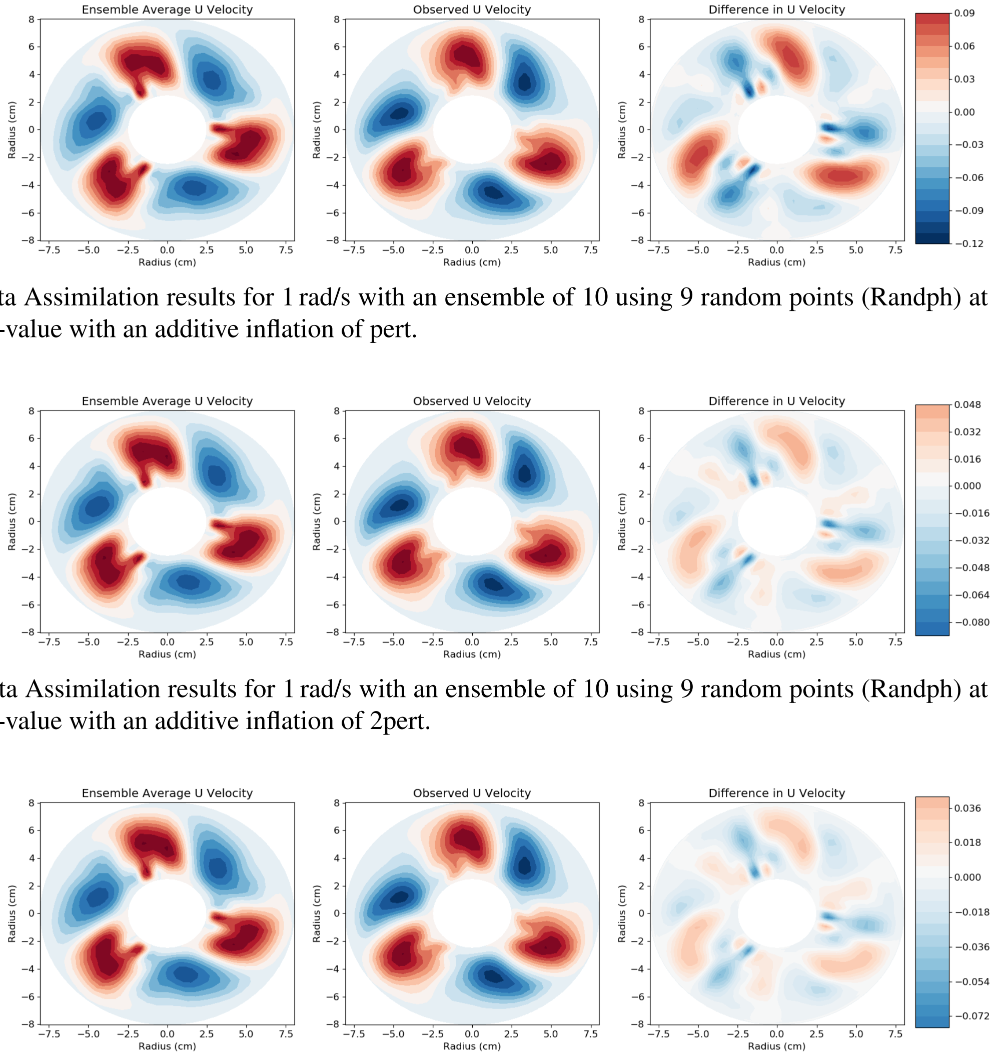
In the last study, half of the model is observed i.e. $\theta = 0$ to 3.14 radians (0-32 θ), giving $32 \times 9 \times 4 = 1152$ observations for DA. The results for the run are shown in figure (7.32b) and as expected after the previous results the system is still doing a good job of tracking the truth. Increasing the number of observed points has made the system more accurate as expected. But it is encouraging to see that the system can track the truth accurately by just observing half the azimuth.

Looking at figure(7.33) it is clear that more observation leads to better prediction. The ensemble gap to the truth decreases as the the number of observation increases. The system remains stable for larger number of observations while becoming more unstable with smaller number of observations. It is interesting to see that using 16 θ gives reasonable results for the gap but a longer run needs to be done to confirm this stability.

7.6.2 Randph Results

Now that a baseline has been set of how the system behaves with regularly spaced observations in the middle of the annulus, a bit more experimentation can be done on where in the θ-plane observations can be taken from. In this section, observation points were chosen randomly in the r and z directions at each θ-value, each time DA is done. In all cases, an ensemble of 10 was used along with an additive inflation of 2pert, unless specified, with the system running for a total of 100 minutes.

To create a baseline similar to Mid, the first test was to look at nine random points at each θ-value (P9). This gives a total of 2304 $(9 \times 64 \times 4)$ observations to be used for DA same as in Mid. In the first test an additive inflation of pert was used to see if the system behaves differently from before. The results of the run are shown in figure (7.34a). The results are very similar to what has been observed in previous sections

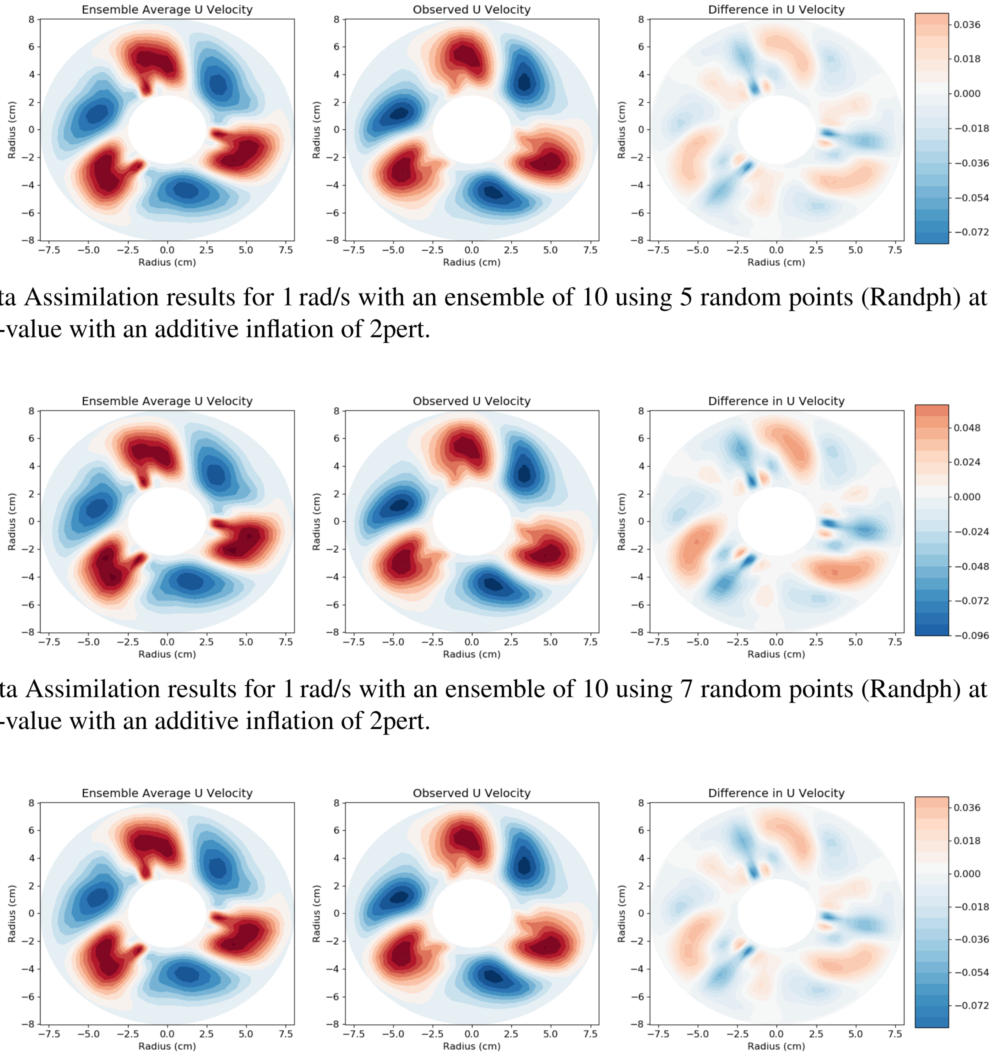


(a) Data Assimilation results for 1 rad/s with an ensemble of 10 using 9 random points (Randph) at each θ-value with an additive inflation of pert.

(b) Data Assimilation results for 1 rad/s with an ensemble of 10 using 9 random points (Randph) at each θ-value with an additive inflation of 2pert.

(c) Data Assimilation results for 1 rad/s with an ensemble of 10 using 3 random points (Randph) at each θ-value with an additive inflation of 2pert.

Figure 7.34: Results for an ensemble of 10 with DA using different random points (Randph). The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.



(a) Data Assimilation results for 1 rad/s with an ensemble of 10 using 5 random points (Randph) at each θ-value with an additive inflation of 2pert.

(b) Data Assimilation results for 1 rad/s with an ensemble of 10 using 7 random points (Randph) at each θ-value with an additive inflation of 2pert.

(c) Data Assimilation results for 1 rad/s with an ensemble of 10 using 12 random points (Randph) at each θ-value with an additive inflation of 2pert.

Figure 7.35: Results for an ensemble of 10 with DA using different random points (Randph). The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

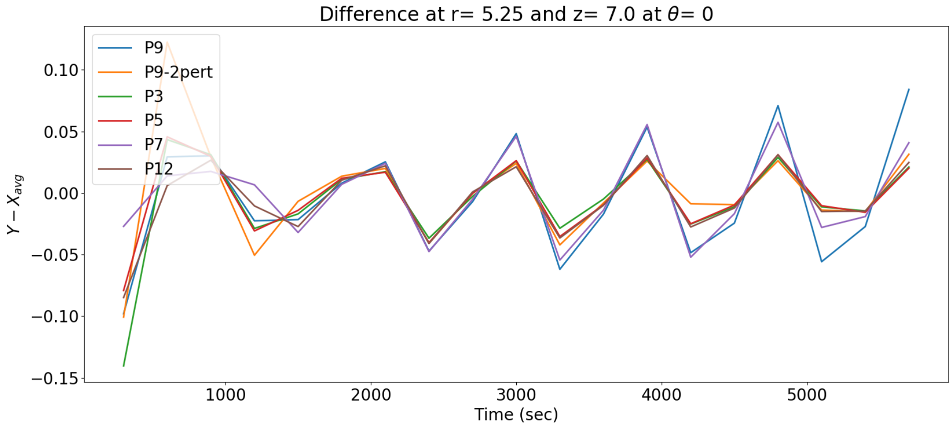


Figure 7.36: The difference between the ensemble average and the observations when using 1 rad/s and Randph observation method.

when pert additive inflation was used with the system lagging behind the truth.

The additive inflation was now increased to 2pert (P9-2pert) for the second test, with the results of the run shown in figure (7.34b). This can be compared with figure (7.20b) where the observed points are concentrated in the middle of the θ-plane. As expected the increase in inflation has helped the system track the truth better. It is still slightly behind but is doing a much better job than the previous study. The spread of the ensemble is also higher here, helping to get better correction and prediction. It is, however, not doing quite as well at tracking the truth as the Mid case where the middle 9 points where observed.

For the second test the number of observed points is lowered to see how the system copes. Here the number of observed points was reduced to just three in each θ-plane (P3), randomly chosen, giving 768 observations. The results for the run are shown in figure (7.34c). The system does very well in tracking the observation even at this low number of observation points. This shows the strength of random points being chosen in combination with the inflation being used. But caution needs to be taken as maybe a repeat experiment might not pick such favourable points.

The number of observed points was increased to five for each θ-value (P5) for the next test. This gives 1280 observations that can be used for DA. The results for the run is shown in figure (7.35a). The system again does very well in tracking the observation but there is no improvement in what was observed with only 3 random points.

The number of observed points is increased further to seven in each θ (P7) location, giving 1792 observations. The results for the run are shown in figure (7.35b). The system is tracking the truth but there is no improvement, this might just be due to what points were randomly chosen to be observed. Points observed near the boundaries might not be helpful as the lower resolution model will not be able to accurately recreate the same complex behaviour that is observed is the higher resolution truth.

For the last test the number of observed points is increased to twelve for each θ-value

(P12), giving 3072 observations. The results for the run are shown in figure (7.35c). Again there does not seem to be an improvement on the tracking. Again this might be due to the random points that were chosen not contributing towards an optimal correction.

Looking at the evolution of the ensemble-truth gap in figure (7.36), the pert system is unstable and the 2pert systems are more stable as was also observed previously. What is also interesting to see is that changing the observation size does not seem to have much affect on the accuracy of the ensemble system. This seems to indicate that in many cases the random points being chosen don't help with the correction of the system. Maybe the region where the points are chosen from needs to be restricted more to get more optimal corrections. This is very much true in the cases where points from the boundary are chosen. The low resolution ensemble might not be able to accurately recreate the same behaviour observed at boundaries in the high resolution truth. Hence EnKF will not be able to correct the system accurately.

7.6.3 Randpph Results

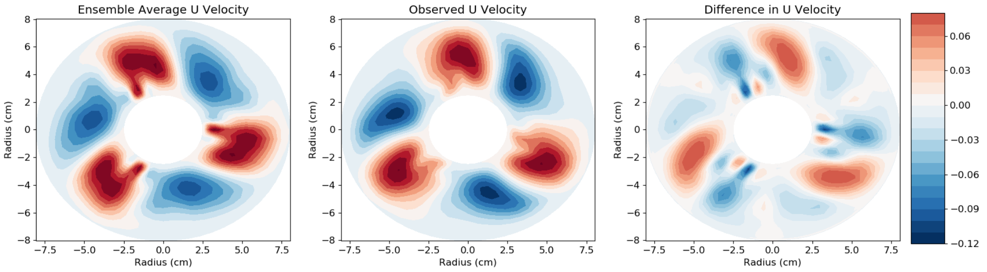
In this section the observation method is slightly changed, with the same random points chosen for observation at the first DA being used in all the other DA as well. Again all simulations are done with an ensemble of 10, an additive inflation of 2pert and run for 100 minutes, unless stated otherwise.

The first test is the baseline test of choosing 9 random points (P9) in each θ-plane and running the model with pert additive inflation. The results for the run is shown in figure (7.37a), and the results are as expected, with the ensemble lagging behind the truth. This lag can be observed looking at the difference between the ensemble average and the truth. The clear structures indicate a differences in the positions of the structures in the ensemble average compared to the truth.

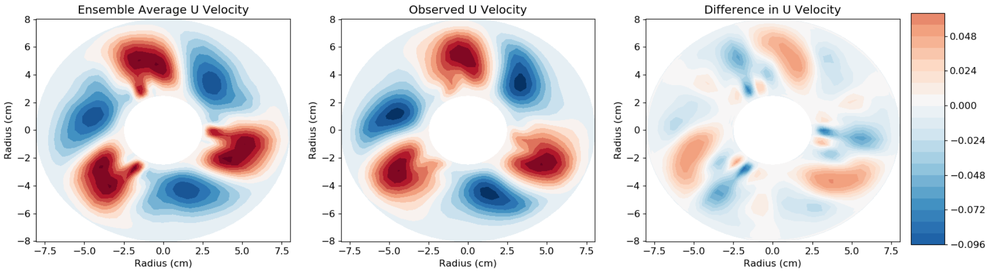
The next step is to increase the inflation to 2pert (P9-2pert) for the next run. The results for the run are shown in figure (7.37b). As expected the results are better, but not as good as what was observed in Randph. With the bigger ensemble spread with the inflation the correction is improved, but this has not helped the system track the observation perfectly. This might be because the points chosen for DA do not lead to an optimal correction. As these suboptimal points are used for the rest of the run, the system is never corrected properly, leading it to lag behind the truth as seen in other studies with suboptimal correction.

For the second test the observed points were reduced to just three points in each θ (P3), giving 768 observations. The results for the run are shown in figure (7.38a), the system does surprisingly well in tracking the observation even at this low number of observed points. The results are better than when nine points were observed with pert additive inflation. This shows how important it is to get the correct observation points and additive inflation for DA.

For the next test the observed points were increased to five points in each θ (P5). The results for the run are shown in figure (7.38b), where the system does better, but the

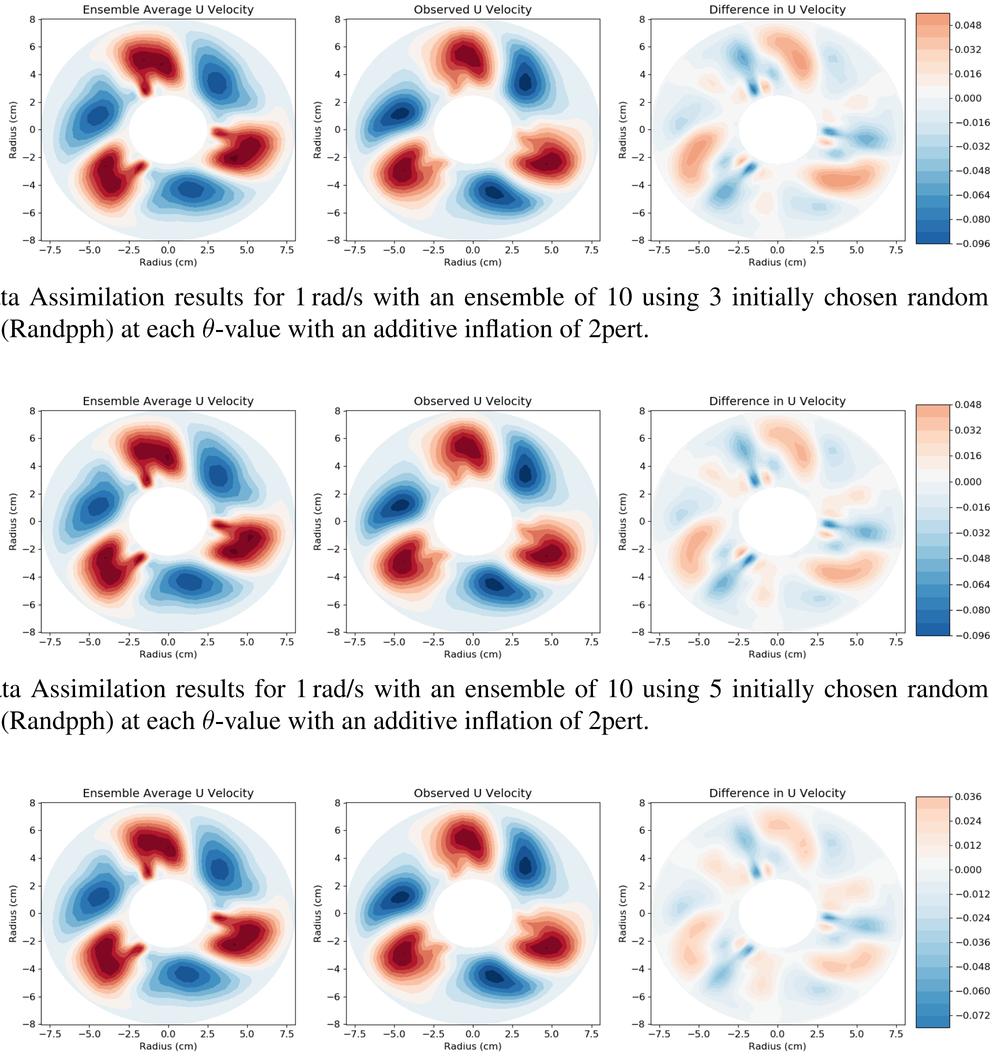


(a) Data Assimilation results for 1 rad/s with an ensemble of 10 using 9 initially chosen random points (Randpph) at each θ-value with an additive inflation of pert.



(b) Data Assimilation results for 1 rad/s with an ensemble of 10 using 9 initially chosen random points (Randpph) at each θ-value with an additive inflation of 2pert.

Figure 7.37: Results for an ensemble of 10 with DA using different random points (Randpph) at each θ-value. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.



(a) Data Assimilation results for 1 rad/s with an ensemble of 10 using 3 initially chosen random points (Randpph) at each θ-value with an additive inflation of 2pert.

(b) Data Assimilation results for 1 rad/s with an ensemble of 10 using 5 initially chosen random points (Randpph) at each θ-value with an additive inflation of 2pert.

(c) Data Assimilation results for 1 rad/s with an ensemble of 10 using 7 initially chosen random points (Randpph) at each θ-value with an additive inflation of 2pert.

Figure 7.38: Results for an ensemble of 10 with DA using different random points (Randpph) at each θ-value. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

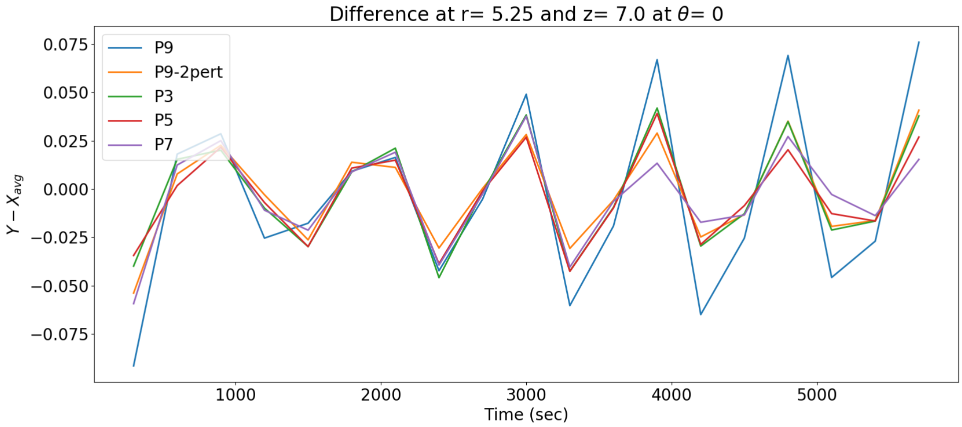


Figure 7.39: The difference between the ensemble average and the observations when using 1 rad/s and Randpph observation method.

results are not substantially better than when three points were used. The results are on par with its Randph counterpart with the system getting very close to the truth.

For the next test the observed points were increased further to seven in each θ (P7). The results for the run are shown in figure (7.38c) the system does very well in tracking the observation. The results here are better than their Randph counterpart. The increase in the number of observations means there is a better chance the system will get points which lead to a more optimal correction, and this seems to have happened here with the system being closer to the truth.

Finally looking at the difference evolution for the systems in figure (7.39), the instability of the pert system can be clearly observed in the P9 plot. The studies with 2pert additive inflation show very similar results to each other and appear stable. A longer run will show if these systems remain stable or if they start to become more unstable as the run progresses.

Overall there were no substantial changes as the number of observations was increased. As the observation points chosen for DA are permanent it becomes important that these lead to optimal correction as there won't be a chance to change them. The importance of this is clear in some of the studies such as P7 and P3 which were better than their Randph counterpart and p9 where it was worse. Again as mentioned in the Randph more restrictions on where the random points are chosen from could lead to more optimal corrections.

7.6.4 Randp Results

The random approach to choosing observation points is pushed further and now instead of a set number of random points being chosen from each θ-value, points can be chosen from anywhere in the model every time DA is done. This is a good test to see if points need to be observed regularly around the model as in Randph and Randpph or can be observed at random points around the model and still give just as good a result. Again all simulations are done with an ensemble of 10 and an additive inflation of 2pert and run for 100 minutes, unless stated otherwise.

For the first test for a baseline, the same number of random points were used as when 9 points (P9) were observed in each θ-plane; 576 points giving an observation vector with 2304 components (since U, V, W, and T are observed). For the first run an additive inflation of pert was used. The results for the run are shown in figure (7.40a). The results are as expected given the previous results using pert and the system lags behind the truth.

The number of observed points was reduced for the next test to the equivalent of just 3 points (P3) being observed in every θ-plane giving $4 \times 192 = 768$ observations. Using an additive inflation of 2pert the results for the run is shown in figure (7.40b). The system does very well at tracking the observation at this low number of observation points. The results are very similar to previous studies in Randph and Randpph. It might be that the system in these studies ended up with very similar random points which lead to favourable corrections.

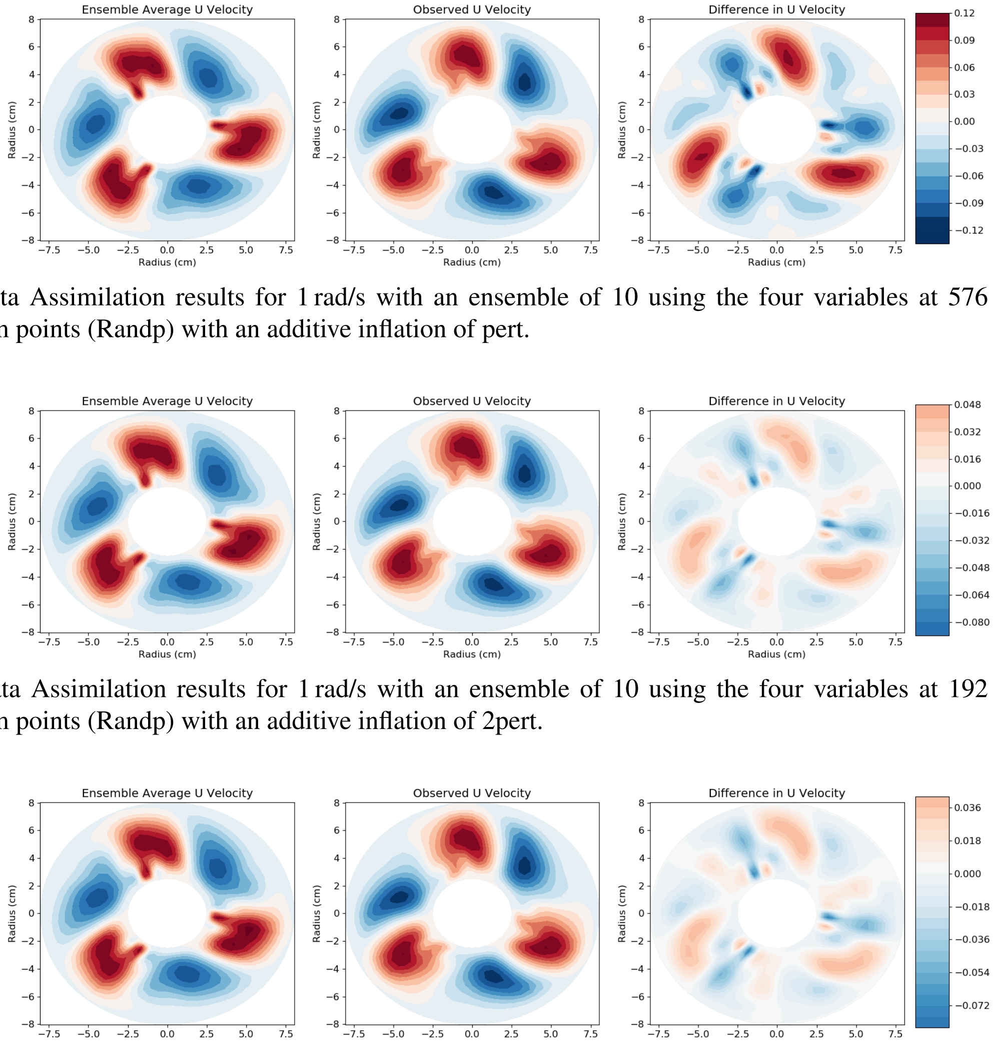
In the next study the number of observations increased to the equivalent of 5 observations (P5) per θ giving 320 points or 1280 observations. The results for the run is shown in figure (7.40c). The system improves upon the P3 results and track the truth closer then before but it is not significantly better.

For the next study the number of observations increased to the equivalent of 7 observation (P7) per θ giving 448 random points 1792 observations that are used for DA. The results for the run is shown in figure (7.41a). The system improves again on the previous study of P5 and is nearly on top of the truth.

In the last study the number of observations were increased to the equivalent of 12 observation (P12) points per θ giving 768 random points or 3072 observations. The results for the run are shown in figure (7.41b). The system has done a good job of tracking the observation but there does not seem to be much improvement from the previous study of P7. This indicates that once there is a sufficient number of observed points, adding further points does not help that much.

The difference evolution in figure (7.42) also shows similar behaviour as previously observed. The P9 run is very unstable as only an additive perturbation of pert was used. But the other system show signs of the system stabilizing as the run progresses with smaller gaps as the run progresses. Overall an increase in the number of observed points leads to better results, unlike in Randph and Randpph where improvements in tracking the truth slowed down significantly with increasing numbers of observation points. Using random points around the model for DA provides a better solution than observing every θ.

The problem with Randpph is that an unfortunate choice of radial point can affect every assimilation step. With Randp, as random points were chosen throughout the model, the small number of unhelpful point near the boundaries could be outnumbered by more helpful points in other parts of the model. And as random points are chosen at every assimilation step, suboptimal correction at one step can be corrected with better

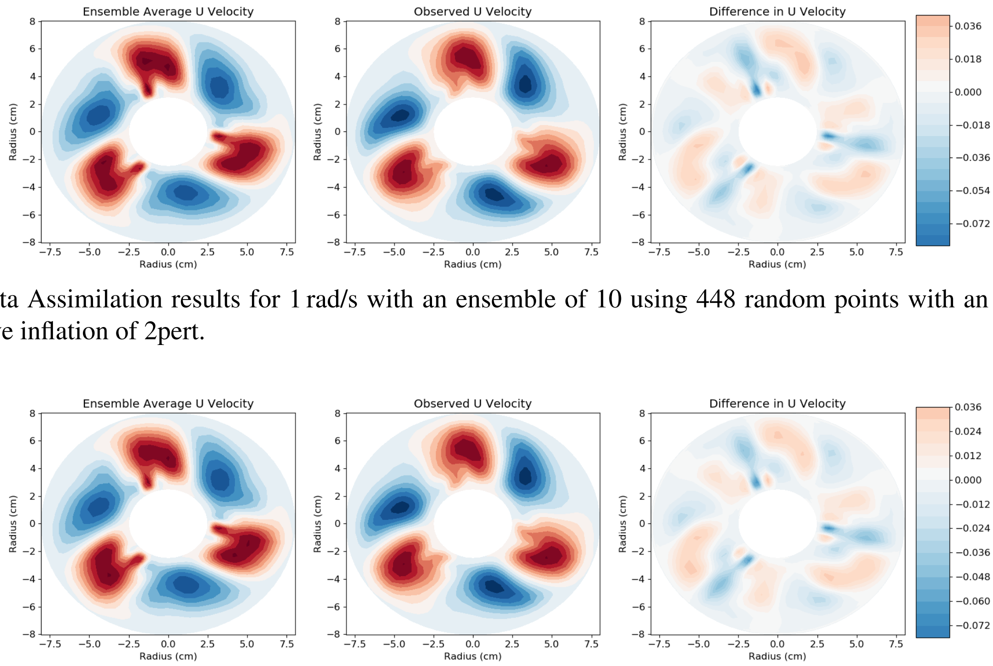


(a) Data Assimilation results for 1 rad/s with an ensemble of 10 using the four variables at 576 random points (Randp) with an additive inflation of pert.

(b) Data Assimilation results for 1 rad/s with an ensemble of 10 using the four variables at 192 random points (Randp) with an additive inflation of 2pert.

(c) Data Assimilation results for 1 rad/s with an ensemble of 10 using the four variables at 320 random points (Randp) with an additive inflation of 2pert.

Figure 7.40: Results for an ensemble of 10 with DA using different random points (Randp) at each θ-value. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.



(a) Data Assimilation results for 1 rad/s with an ensemble of 10 using 448 random points with an additive inflation of 2pert.

(b) Data Assimilation results for 1 rad/s with an ensemble of 10 using 768 random points (Randp) with an additive inflation of 2pert.

Figure 7.41: Results for an ensemble of 10 with DA using different random points (Randp) at each θ-value. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

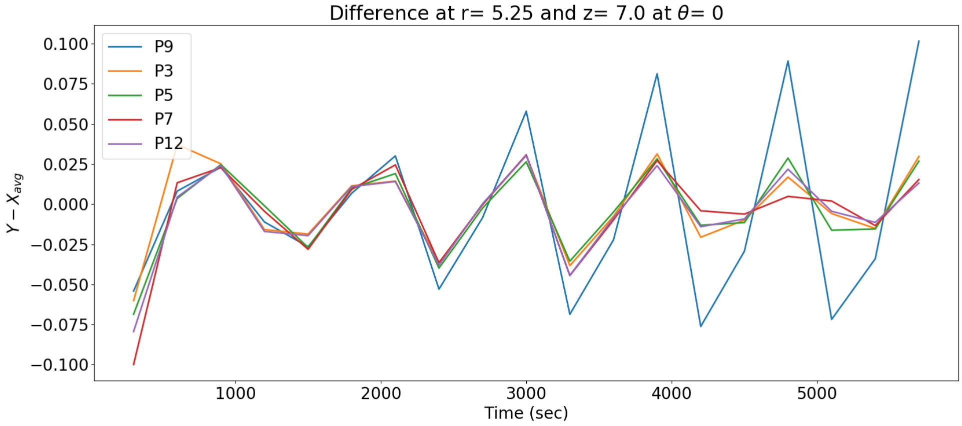


Figure 7.42: The difference between the ensemble average and the observations when using 1 rad/s and Randp observation method.

chosen random observed points and better prediction at the next assimilation step. But a bit more study on where the best place to observe needs to be done, such as was done in Mid, to refine the Randp approach.

7.6.5 Randpp Results

In the second approach to using random points in the model for DA, the random points are chosen at the first DA and are then used throughout the run for DA. This is similar to what was done in the previous Randpph study.

For the first test as before 576 random points (P9) were chosen in the first DA, and these points were used throughout the run, giving 2304 observations for DA. Here results are presented using an additive inflation of 2pert in figure (7.43a). The system tracks the observations very well and the ensemble seems to be well spread at this last stage in the run. This is expected as these parameters have given suitable results throughout the studies presented in this chapter.

A slightly different approach can also be taken here where 576 random points (P9UV) are observed but only the U and V velocity variables are used for DA giving 1152 observations. The results for the run are shown in figure (7.43b). The results are very similar to what was observed when all four variables were used for DA. It is very encouraging to see that the system can cope with tracking the truth with only half the variables. It could also be that the point randomly chosen lead to a more accurate corrections.

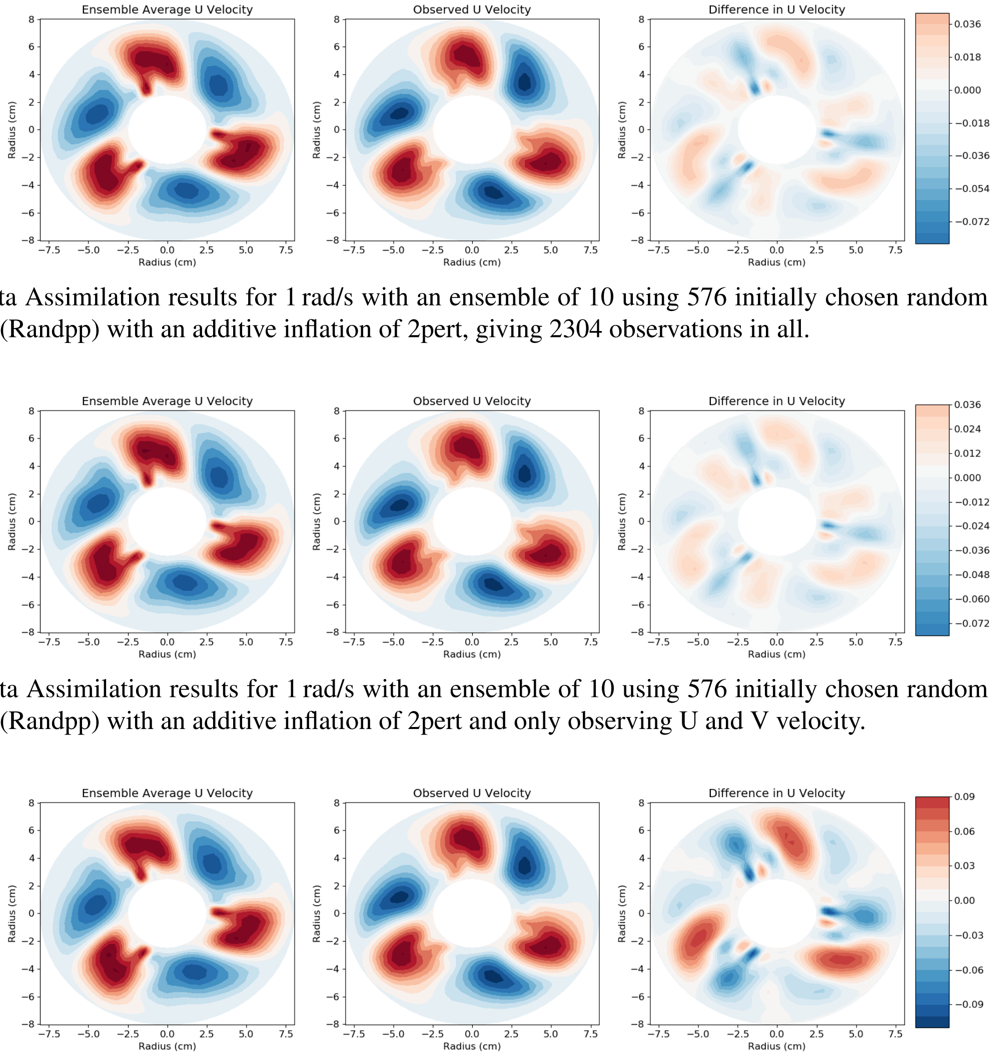
In the next study the number of observations is cut down to 64 random points (P1UV) and only U and V velocity are observed giving 128 observations. The results for the run is shown in figure (7.43c) where the system lags behind the truth. This is understandable with the low number of observed points but it is encouraging that the system is not lagging too far behind.

In the next study the number of observations increased to the equivalent of 3 observation per θ (192 points) using all 4 variables, giving 768 observations (P3). The results for the run is shown in figure (7.44a). The system understandably does better then where only 64 points were used but not drastically even with the number of observations increasing significantly.

The number of observations is increased further to the equivalent of 5 observation per θ (320 points) giving 1280 observations (P5). The results for the run is shown in figure (7.44b). The system tracks the observation well with the results being on par with what is observed in Radnp where 320 points were used for DA.

For the last study the number of observations increased to the equivalent of 7 observation per θ (448 points) giving 1792 observations (P7). The results for the run is shown in figure (7.44c). The system does well of tracking the truth but no gains are made when compared to when 320 points were used indicating that the observed points don't bring much to tracking the truth.

Looking at the difference evolution in figure (7.45), the systems with low number of observed points such as where 64 points and U and V velocity (P1UV), and where 192

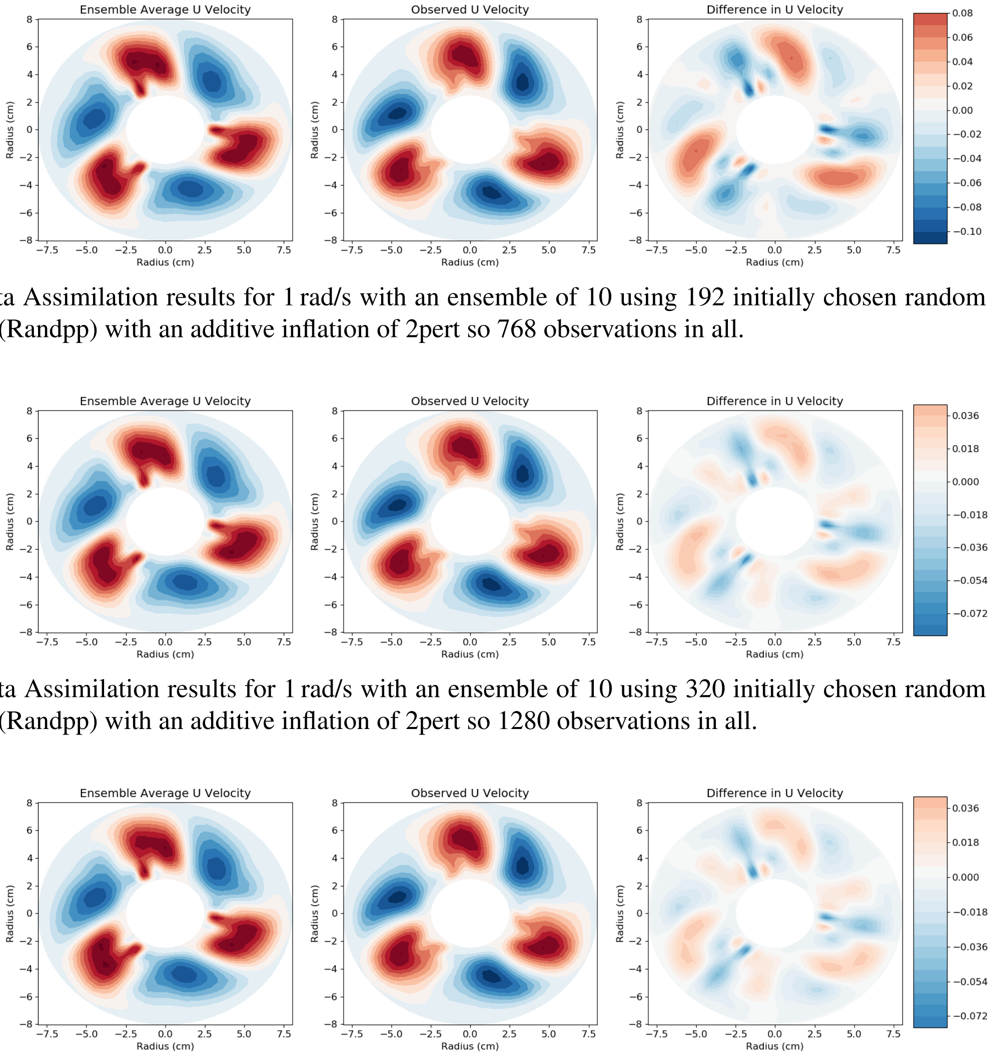


(a) Data Assimilation results for 1 rad/s with an ensemble of 10 using 576 initially chosen random points (Randpp) with an additive inflation of 2pert, giving 2304 observations in all.

(b) Data Assimilation results for 1 rad/s with an ensemble of 10 using 576 initially chosen random points (Randpp) with an additive inflation of 2pert and only observing U and V velocity.

(c) Data Assimilation results for 1 rad/s with an ensemble of 10 using 64 initially chosen random points (Randpp) with an additive inflation of 2pert and only observing U and V velocity.

Figure 7.43: Results for an ensemble of 10 with DA using different random points (Randpp) at each θ-value. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.



(a) Data Assimilation results for 1 rad/s with an ensemble of 10 using 192 initially chosen random points (Randpp) with an additive inflation of 2pert so 768 observations in all.

(b) Data Assimilation results for 1 rad/s with an ensemble of 10 using 320 initially chosen random points (Randpp) with an additive inflation of 2pert so 1280 observations in all.

(c) Data Assimilation results for 1 rad/s with an ensemble of 10 using 448 initially chosen random points (Randpp) with an additive inflation of 2pert, so 1792 observations in all.

Figure 7.44: Results for an ensemble of 10 with DA using different random points (Randpp) at each θ-value. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

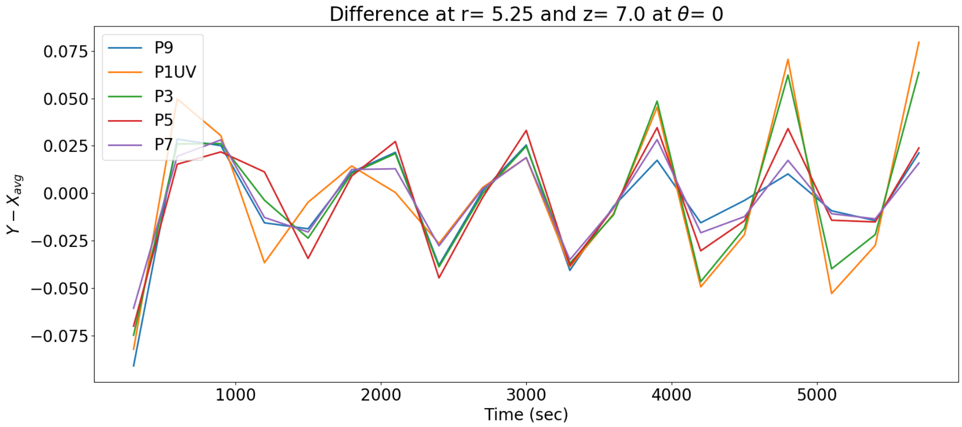


Figure 7.45: The difference between the ensemble average and the observations when using 1 rad/s and Randpp observation method.

points (P3), show unstable behaviour. As the number of observed points are increased the stability of the system increases as also seen in previous sections.

In most cases, the results are comparable to what was observed in the previous sections, for example, the results in 320 and 448 points were similar to Randp 320 and 448 points. But using 192 points the system did less well in tracking the observation than its counterpart in Randp. Overall, it seems that the system does well in tracking the truth for the steady flow of 1 rad/s, provided the larger inflation perturbation 2pert is used and provided sufficient points are being observed. It is important to note that observations need to be spread around the model for accurate correction. When choosing random points can lead to data rich and data poor areas which can lead to suboptimal corrections. There is a significant improvement when all four variables are observed rather than just U and V.

7.7 Results for 2 rad/s

As the limits of EnKF's capability has been understood to an extent at 1 rad/s, the rotation rate can now be increased to 2 rad/s to see how EnKF handles a higher rotation rate. The working fluid structures at this higher rotation rate are more complex with the wavenumber for the high resolution simulation increasing from a $m = 3$ to a $m = 4$ solution, as observed in section (7.4.2). In the low resolution ensemble on the other hand a selection of $m = 4$ and $m = 3$ wavenumber structure are observed in the ensemble. There is therefore uncertainty at the low resolution model as to which wavenumber is dominant. This along with the slower drift rate of the low resolution of the ensemble, as reported in table (7.2), will make it interesting to see how EnKF system copes when the ensemble is composed of two fundamentally different types of solution.

7.7.1 Mid results

To start the study a simple Mid setup was used as discussed in section (7.3.2). The middle nine points of every θ were observed, using all the 4 variables (U, V, W velocity and Temperature) at each of the 576 points giving a total of 2304 observations being used for DA.

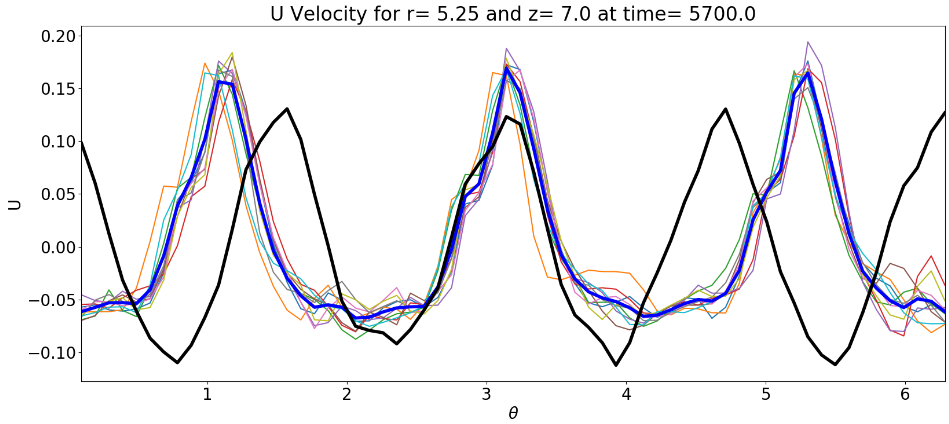
For the first test the system is run with an additive inflation of pert (P9-pert), with DA every 5 minutes. The U velocity results for the simulation are shown in figure (7.46a). The ensemble average is an $m = 3$ which is different from the $m = 4$ wavenumber observed in the truth. This means that the ensemble follow the truth accurately as it has completely different structure. This difference in wavenumber could be due to DA destabilising the system making it collapse from an $m = 4$ to $m = 3$ wavenumber.

To see if this is a problem of ensemble collapse due to correction, the additive inflation was increased to 2pert (P9-2pert). This will spread the ensemble out more, hopefully make the ensemble track the truth more accurately and help maintain the $m = 4$ wavenumber. The U velocity results for this simulation are shown in figure (7.46b), and it seems that having the higher additive inflation has solved the problem of the wavenumber. But it has also led to the problem of the ensemble being more chaotic as observed in the differing behaviour of the ensemble members. With the slower drift rate of the low resolution ensemble the system always found it hard to track the truth accurately.

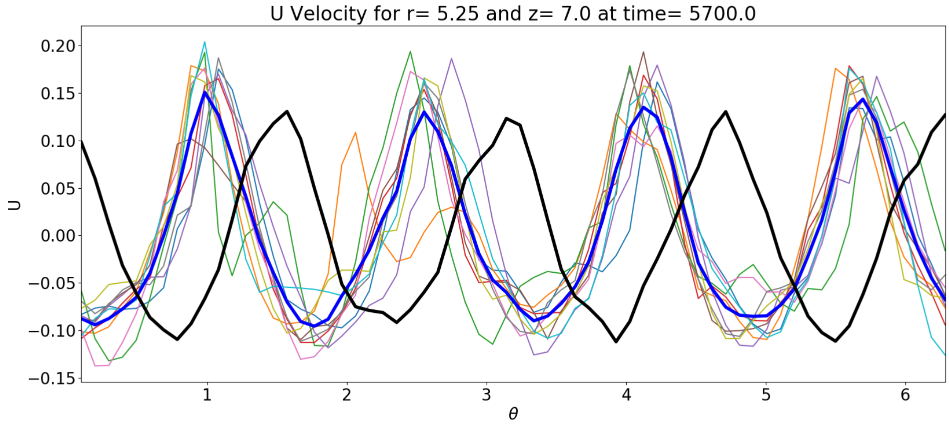
To take into account the more chaotic nature of the system at higher rotation rate, the inflation was reduced to pert but the rate of DA was increased to every 1 minute (1min-pert). The results are shown in figure (7.47a) and it seems that the wavenumber problem has been solved and the ensemble average is an $m = 4$ solution. But the combination of more frequent DA and lower inflation has caused the ensemble spread to be much smaller. Even though the wavenumber is correct, the ensemble lags behind the truth and the correction are not able to help the system track the truth.

Using the higher 2pert as observed before leads to a more chaotic results, a way to overcome this issue might be to increase the gap between DA to 10 minutes (10min-2pert). This should give time for the ensemble members to get to a more stable state before DA is done. The results for this study are plotted in figure (7.47b). With the longer gap between DA, the ensemble members have been given more time to evolve and have drifted apart. Furthermore, it is interesting that looking at the individual ensemble members there is some variation in wavenumber with some showing $m = 3$ (red lines) and others $m = 4$ (blue, green lines). This variation in wavenumber might be the reason for the problems this study has in trying to track the truth.

Keeping with DA at every 5 minutes and additive inflation of 2pert and the ensemble size is increased to 20 to help stabilise the system (m20-2pert). The results for this study are plotted in figure (7.48). With the larger ensemble size the system seems more stable and produces an ensemble average with an $m = 4$ wavenumber. The variation observed between ensemble members in the previous study has gone, with all members exhibiting an $m = 4$ wavenumber, but this system still lags behind the truth. Maybe an even higher ensemble size with more frequent DA will lead to a even better results.

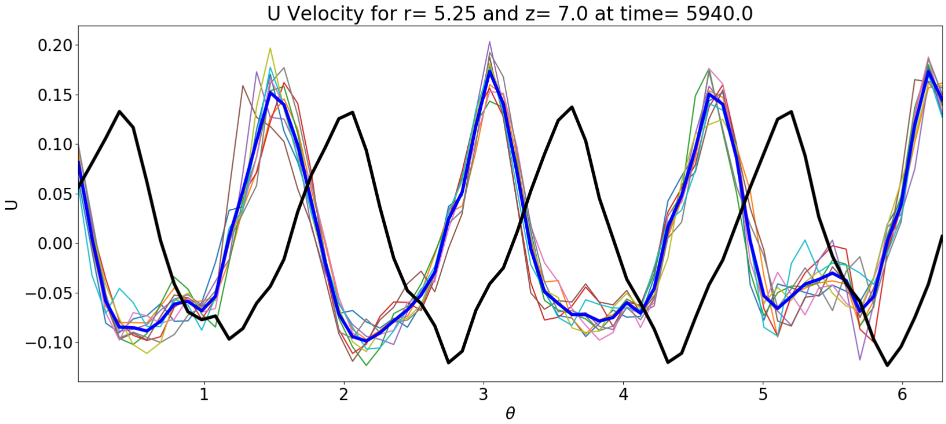


(a) The graph shows the U velocity results with an additve inflation of pert. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

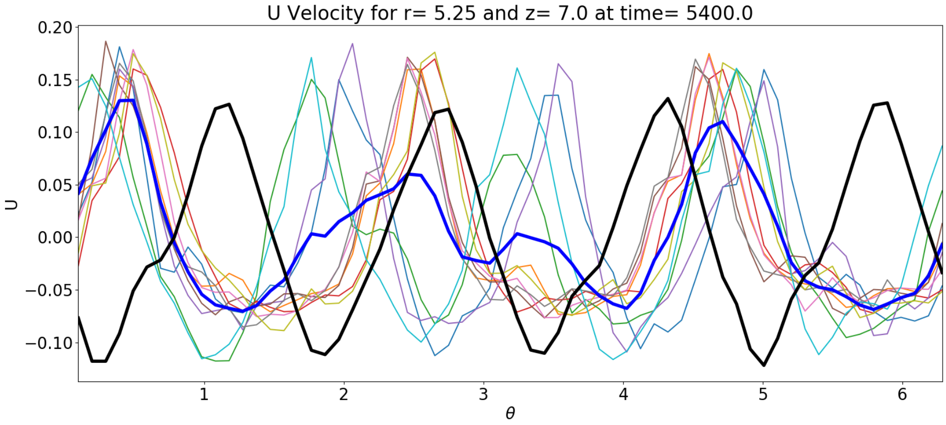


(b) The graph shows the U velocity results with an additve inflation of 2pert. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

Figure 7.46: Data Assimilation results at 2 rad/s for an ensemble of 10 looking at nine points in the middle of each θ (Mid) with different additive inflations.



(a) The graph shows the U velocity with an additive inflation of pert and with DA every 1 minute. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.



(b) The graph shows the U velocity with an additive inflation of 2pert and with DA every 10 minutes. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

Figure 7.47: Data Assimilation results for 2 rad/s with an ensemble of 10 looking at nine points in the middle of each θ (Mid) with different additive inflations and different assimilation rates.

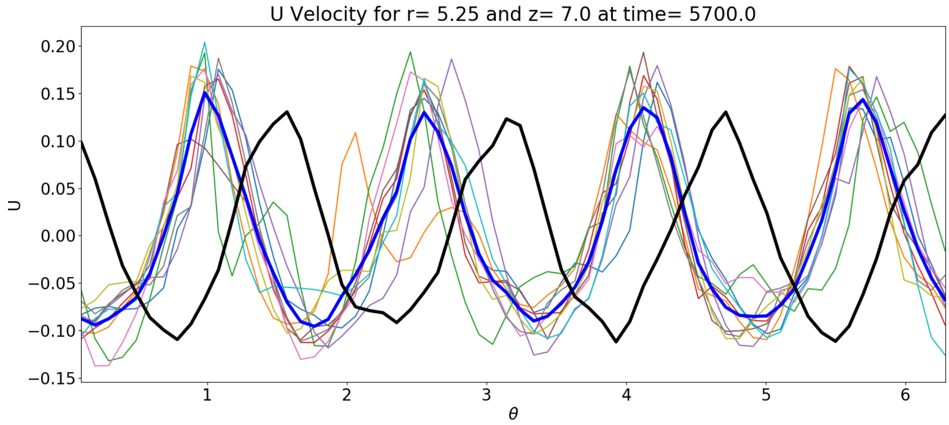


Figure 7.48: Data Assimilation results at 2 rad/s for an ensemble of 20 looking at U velocity using the Mid observations. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

Looking at the difference of the system in figure (7.49), all the runs show instability with large difference observed for all runs. All the runs fluctuate at similar times indicating that all systems seems to have a similar drift rate irrespective of the DA time, ensemble size or additive inflation. This shows how hard it is for EnKF to change the behaviour of the ensemble system when is comes to drift rate.

7.7.2 Randph Results

For the next study, 9 random points (randph) at each θ were observed for DA giving 2304 observations. DA was done every 5 minutes and a new set of random 9 points were used for every DA totalling 576 grid points. Using an additive inflation of 2pert the results for the simulation are shown in figure (7.50a). The ensemble average is an $m = 3$ wave, compared to the $m = 4$ wave observed in the truth, indicating that DA is suboptimal with these settings/observed points. A repeat study with better observed points might help get better tracking. And if a few of the ensemble members showed an $m = 4$ wave, a better solution could be obtained. Conducting DA more often i.e. every minute, might also help solve some of the issues. Difference plot for the studies presented here are shown in figure (7.51) and discussed in the next section.

7.7.3 Randpph Results

For this study 9 random points (randpph) were observed in each θ giving 2304 observations for the first DA and these points were subsequently used in every DA. In the first test an additive inflation of 2pert was used along with DA every 5 minutes. The results of the simulation are shown in figure (7.50b). Here the system has again collapsed to an

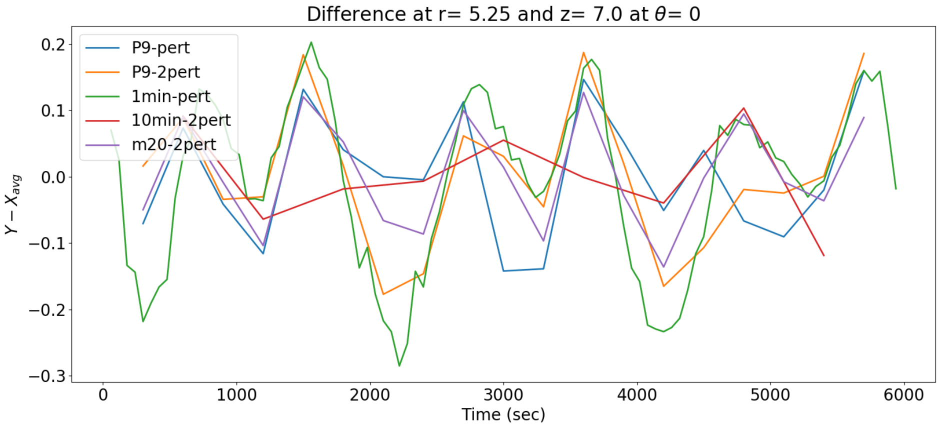


Figure 7.49: The difference between the ensemble average and the observations when using 2 rad/s and mid observation method.

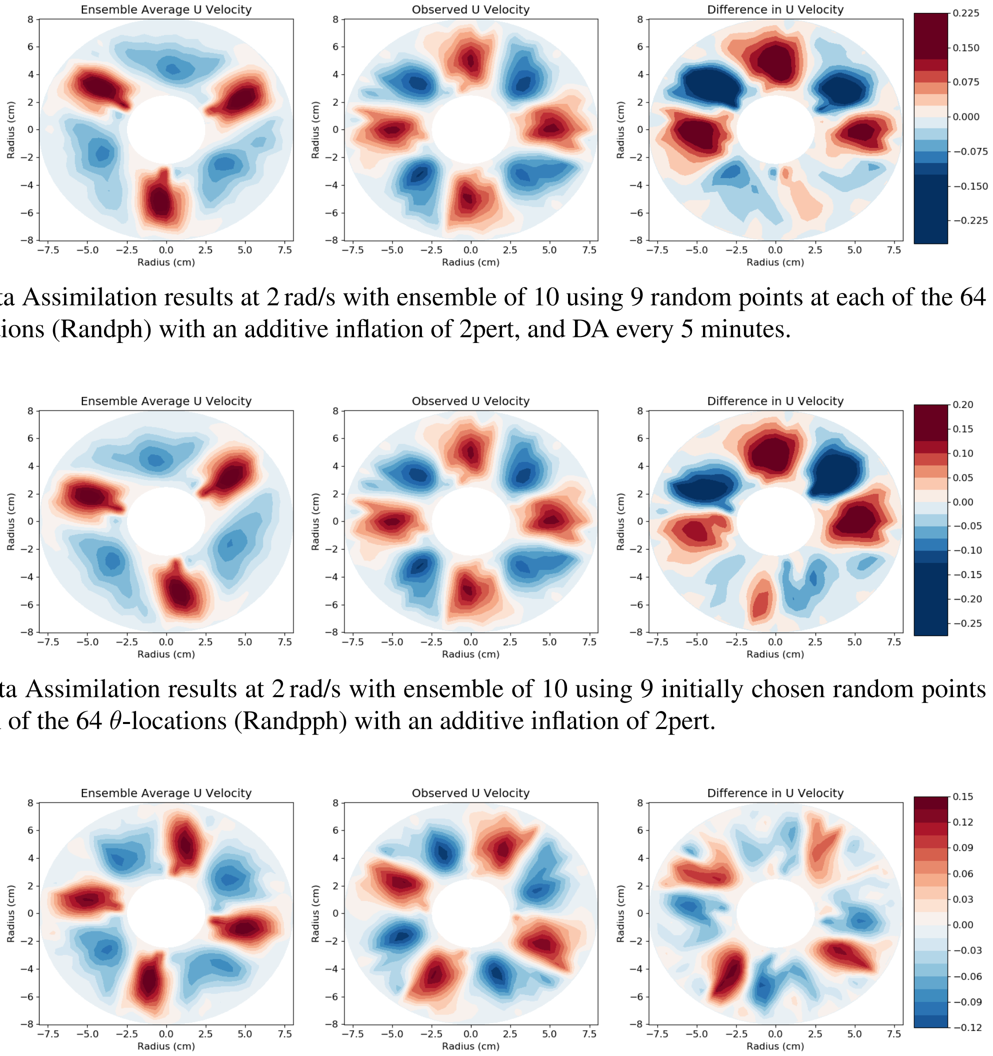
$m = 3$ wavenumber. This seems to indicate that the random points chosen for DA were not optimal.

To see if the tracking can be improved the time between DA was reduced to 1 minute while using the 2pert additive inflation (randpph-min1). The results for this are shown in figure (7.50c) and shows the $m = 4$ wavenumber. Although the wavenumber is correct the system itself is not yet tracking the truth accurately. There is still a large difference between the ensemble average and the truth. Maybe a similar test can be done in the previous studies to test for the correct wave number.

Figure (7.51) shows how unstable the systems are, with all three runs showing large changes. Even looking at the 1 minute run which showed a $m = 4$ wavenumber, there are points where the gaps decreases but this is just followed by a very large change. This indicates the system is drifting in and out of phase with the truth.

7.7.4 Randp Results

For the first test, 576 random points (randp) are chosen from throughout the model at every DA giving 2304 observations at each DA. Using an additive inflation of 2pert DA was done every minute. The results for the run are shown in figure (7.52). The structure of the system is an $m = 4$ wave which is encouraging. The ensembles have a good spread and each member of the ensemble seem to show slightly different behaviour while maintaining the $m = 4$ structure. Even with the encouraging results the ensemble is not able to track the truth accurately, drifting at a different rate then the truth, similar to what was observed in Ranpph. This is clearly observed in figure (7.52b), where the drift of the ensemble is slower when compared to the truth. Difference plot for the studies presented here are shown in figure (7.54) and discussed in the next section.



(a) Data Assimilation results at 2 rad/s with ensemble of 10 using 9 random points at each of the 64 θ-locations (Randph) with an additive inflation of 2pert, and DA every 5 minutes.

(b) Data Assimilation results at 2 rad/s with ensemble of 10 using 9 initially chosen random points at each of the 64 θ-locations (Randpph) with an additive inflation of 2pert.

(c) Data Assimilation results at 2 rad/s with ensemble of 10 using 9 initially chosen random points at each of the 64 θ-locations (Randpph) with an additive inflation of 2pert and DA every 1 minute.

Figure 7.50: Data Assimilation results at 2 rad/s with ensemble of 10 with an additive inflation of 2pert. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

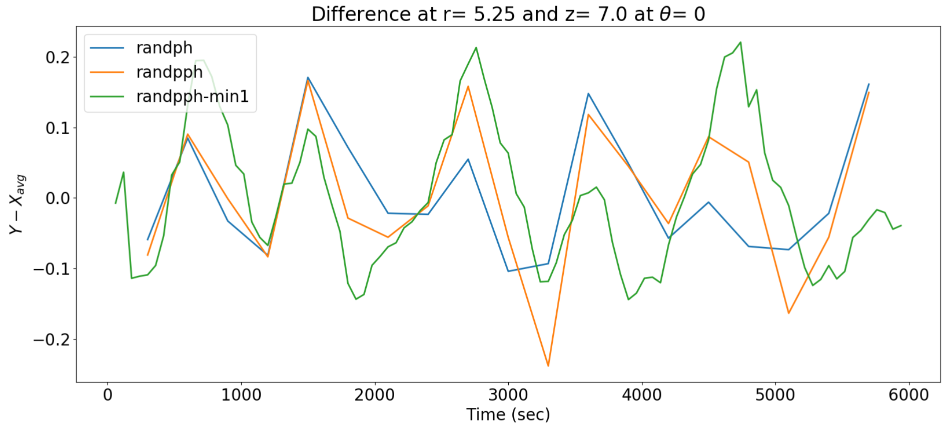


Figure 7.51: The difference between the ensemble average and the observations when using 2 rad/s and Randph observation method.

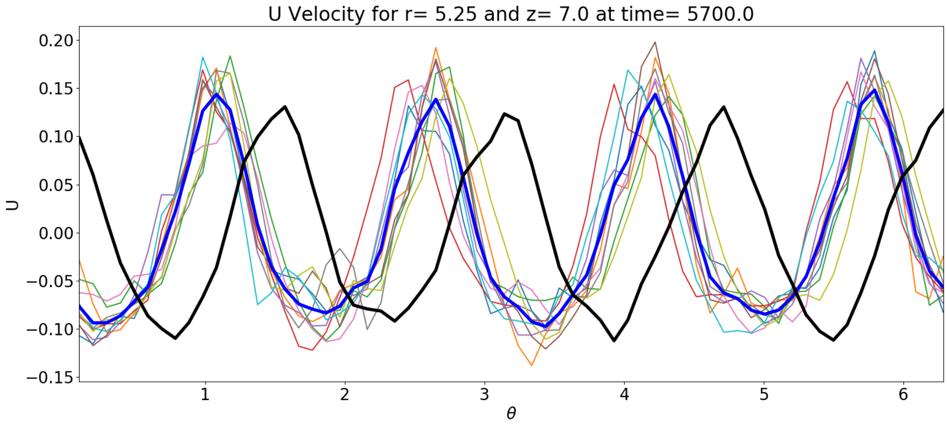
7.7.5 Randpp Results

The test is now changed to the first set of random point being used for all DA done during the run. Using 576 random grid points (randpp) giving 2304 observations with an additive inflation of 2pert. With DA every 5 minutes the results of the run are shown in figure (7.53a). The system has collapsed to an $m = 3$ wave and is not able to track the truth accurately. After the formation of an $m = 3$ wave EnKF is no longer able to correct the system and reform a $m = 4$ wave. The reason for this failure is probably a combination of what points were randomly chosen for observation, the time between DA and the inflation used.

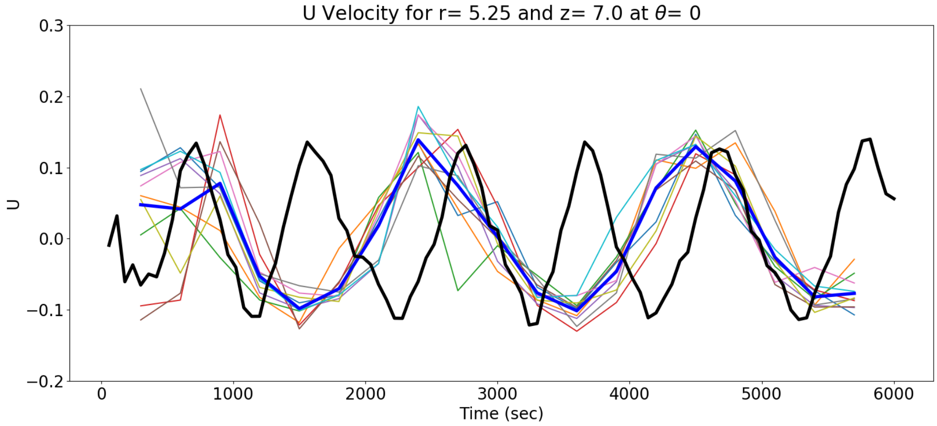
The DA time is now reduced to 1 minutes to see if the problem of the wavenumber can be fixed (randpp-min1). Using a additive inflation of 2pert the results for the run are shown in figure (7.53b). Here the problem of the incorrect wavenumber has been resolved giving a wavenumber $m = 4$ instead of $m = 3$. But the system is still not able to track the truth. The smaller time between DA is leading to the correct wavenumber, but possibly because a suboptimal set of points was selected for DA at the beginning, the system has failed to track the truth accurately. This only exacerbates the problem of the slower drift rate of the ensemble.

It is not a surprise that since no runs were close to the truth, the difference graph in figure (7.54) also show this result. All the runs are unstable with large changes that persist as the system propagates. In most cases, the low resolution ensemble system struggles to replicate this $m = 4$ wavenumber and stays at an $m = 3$ wavenumber. It was possible to induce the ensemble members to lock on to $m = 4$ by decreasing the time between successive DA to 1 minute. However, even when $m = 4$ is achieved, the system usually lagged behind the truth. Overall DA was not successful as tracking the truth.

Another problem that plagues the system is the slower drift rate of the low resolution ensemble. This can be observed in the results of Randpph in figure (7.50b), where the

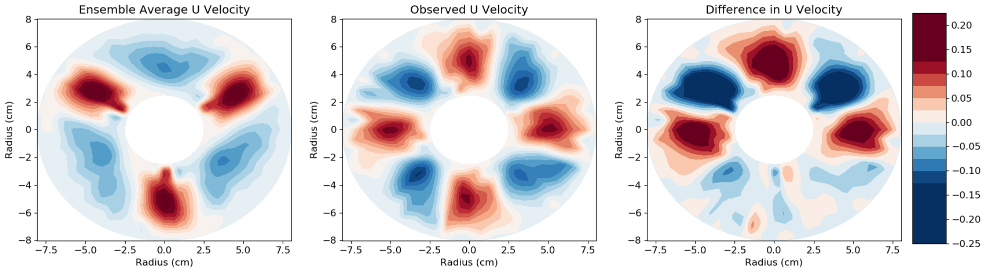


(a) The graph shows the U velocity across the annulus with DA every 5 minutes. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

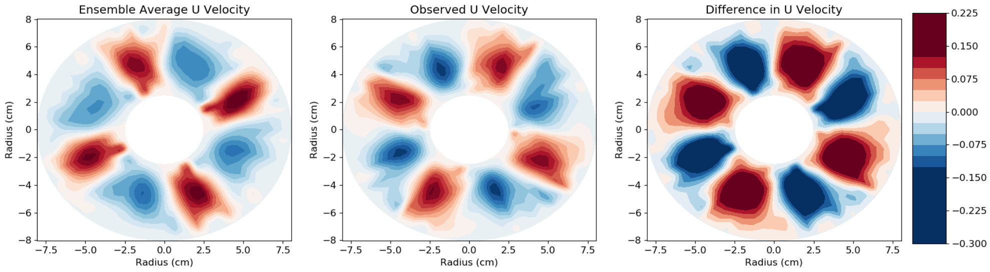


(b) The graph shows the U velocity through out the run of the simulation. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

Figure 7.52: Data Assimilation results at 2 rad/s with ensemble of 10 using 576 random points (Randp) with an additive inflation of 2pert.



(a) The graph shows the U velocity with 2304 observations used for DA every 5 minutes.



(b) The graph shows the U velocity with 2304 observations used for DA every 1 minutes.

Figure 7.53: Data Assimilation results at 2 rad/s with ensemble of 10 with random points (Randpp) with an additive inflation of 2pert. The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

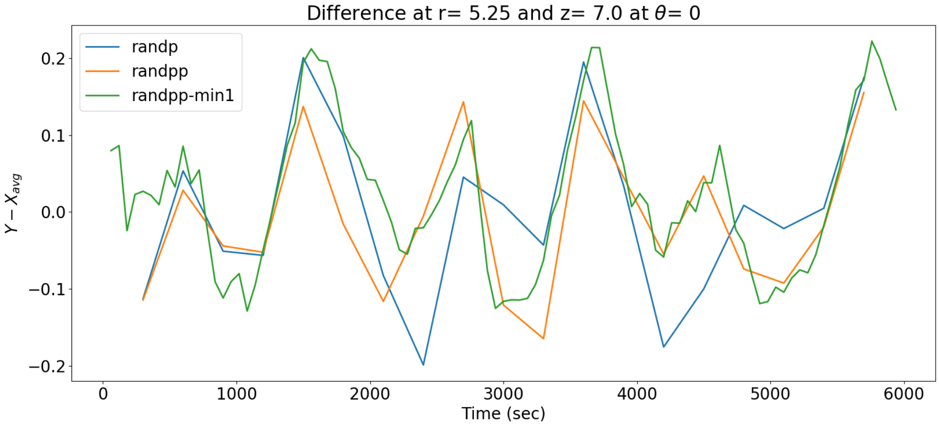


Figure 7.54: The difference between the ensemble average and the observations when using 2 rad/s and Randp observation method.

system looks like it is accurate towards the end of the run but looking at how the system behaves throughout the simulation shows how the system lags behind the truth. It might be interesting to add multiplicative inflation to the study along with the additive inflation already in use to see if the system can improve the drift rates for the ensemble and help track the truth. These problems might also be solved by increasing the resolution in the θ which might lead to better transfer of information around the annulus.

7.8 Results for 2.5 rad/s

The rotation rate was now is increased further to 2.5 rad/s leading to more chaotic structures. As seen in section (7.4.3) the structures at this rotation rate are unstable and can collapse from an $m = 4$ to an $m = 3$ wavenumber. This volatile situation will prove a tough test for EnKF. The drift rate of the working fluid structures are also very slow as seen in section (7.4.3), this will add to challenges faced by EnKF in this section.

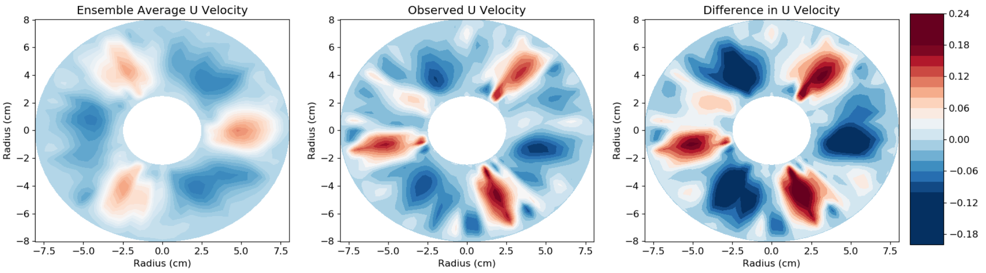
7.8.1 Mid results

The results presented here are where the 9 middle points are observed in each θ, for all 4 variables. All studies were done using an additive inflation of 2pert with the system running for 100 minutes.

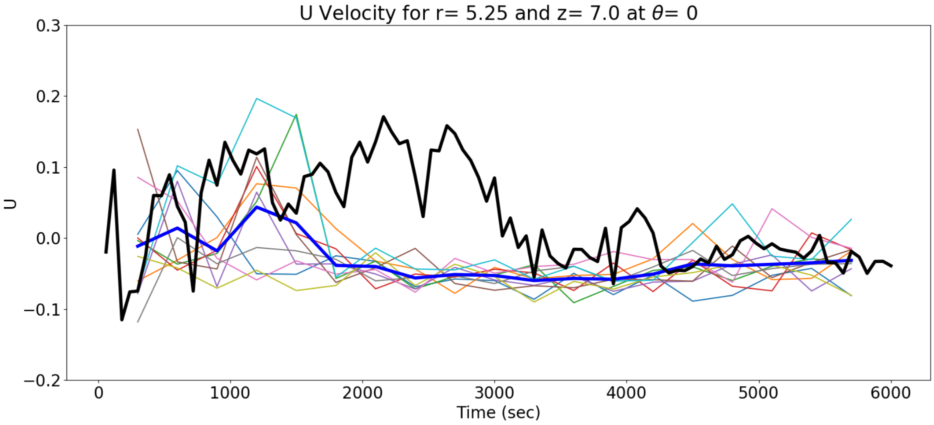
With DA done every 5 minutes (P9) the results of the simulation are shown in figure (7.55). As observed the system is having a lot of trouble tracking the truth. Although both the ensemble and truth display an $m = 3$ wave, the ensemble system is unable to track the truth. The ensemble is quite spread out leading to a smaller ensemble average. The spread is due to the combination of the unstable nature of the fluid at the higher rotation rate and the additive perturbation that was used for DA that spread the ensemble and add noise to the system.

Looking at how the system evolves in figure (7.55b), the drastic difference between the behaviour of the truth and ensemble can be clearly observed. There are no clear structures that passes through the point as for both the ensemble and the truth as were observed at lower rotation rates. The truth has a very unstable structure and eventually ends up around $U = 0m/s$ towards the second half of its simulation. The ensemble shows a lot of variation in the individual members but they all also end up around the $U = 0$ mark. This seems to indicate that the structures are either stuck or are rotating very very slowly around the annulus.

For the second study DA is done every minute (P9-min1) and the results for this run are shown in figure (7.56), and it seems that the ensemble is now able to track the truth towards the end of the run. Both the truth and ensemble have an m=3 wavenumber, but looking at figure (7.56b) one can clearly see that the truth and ensemble evolve very differently. At the beginning of the run both the ensemble and truth show very similar behaviour, but they both diverge, with the truth displaying positive velocity and

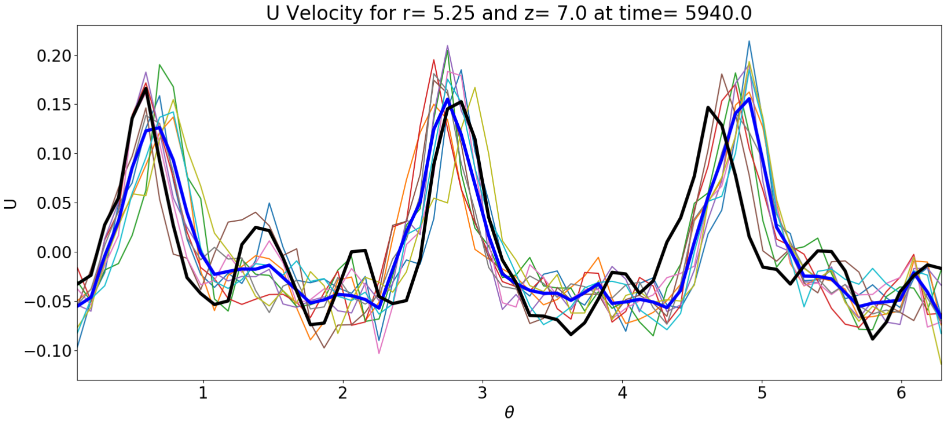


(a) The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

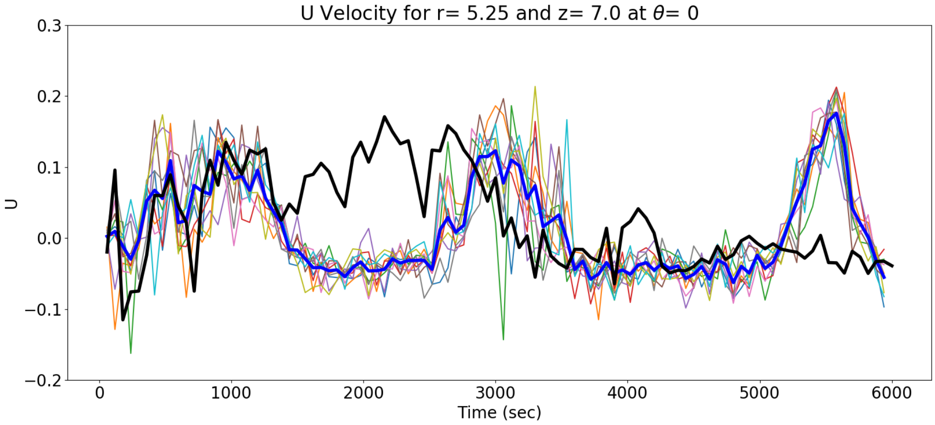


(b) The graph shows the U velocity through out the run of the simulation. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

Figure 7.55: Data Assimilation results for 2.5 rad/s with ensemble of 10 with an additive inflation of 2pert, with DA using Mid points every 5 minutes.



(a) The graph shows the U velocity across the annulus with DA every 1 minutes. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.



(b) The graph shows the U velocity through out the run of the simulation. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

Figure 7.56: Data Assimilation results for 2.5 rad/s with ensemble of 10 with an additive inflation of 2pert, with DA using Mid points every 1 minutes.

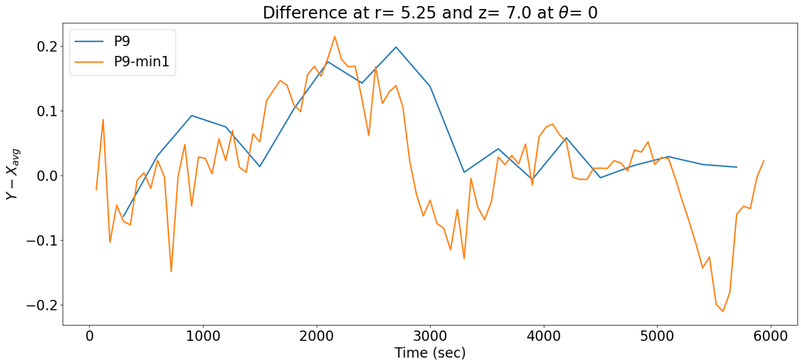


Figure 7.57: The difference between the ensemble average and the observations when using 2.5 rad/s and mid observation method.

the ensemble displaying negative velocity. Both come back together and the ensemble does a good job of tracking the truth. But towards the end of the simulation a fluid wave structure clearly passes the observed point and then is on top of the truth as observed in figure (7.56a). No fluid structures is observed passing the point for the truth and as this was the end of the simulation, it is not known that if the ensemble would have continue to track the truth if the simulation was continued.

Looking at the difference in figure (7.57) the systems where DA was done every minute is very unstable, while the when DA was done every 5 minutes the system seems to have stabilized. But this might just be due to the static behaviour observed in the system as seen in figure (7.55b). Hence these result might not be an accurate representation of the behaviour of the system at this rotation rate. More studies need to be done to get conclusive answers.

7.8.2 Randpph Results

The next few tests were done by looking at how the system behaves when 9 random points (PPHm-10) are chosen in the $r - z$ plane using all the 64 θ locations resulting in 576 random grid points for the truth. Observing all 4 variables gives 2304 observations in all. To help the system cope with the unstable nature of the fluid structure at this rotation rate, the frequency of DA is decreased to 1 minute. Using an additive inflation of 2pert the results for the system are shown in figure (7.58a). Here the ensemble has stayed at wavenumber $m = 4$ and not collapsed to $m = 3$ unlike the truth which collapsed to an $m = 3$ wave. This problem might be due to random points that are being observed for DA.

To try and fix this issue the ensemble size is increased to 20 (PPHm-20). The results for this simulation are shown in figure (7.58b), where the system has been able to correct itself from the $m = 4$ to an $m = 3$ wavenumber. The system does lag behind the truth but nevertheless this improvement is encouraging and maybe using an even higher ensemble size can lead to better results. Difference plot for the studies presented here are shown in figure (7.59c) and discussed in the next section.

7.8.3 Randpp Results

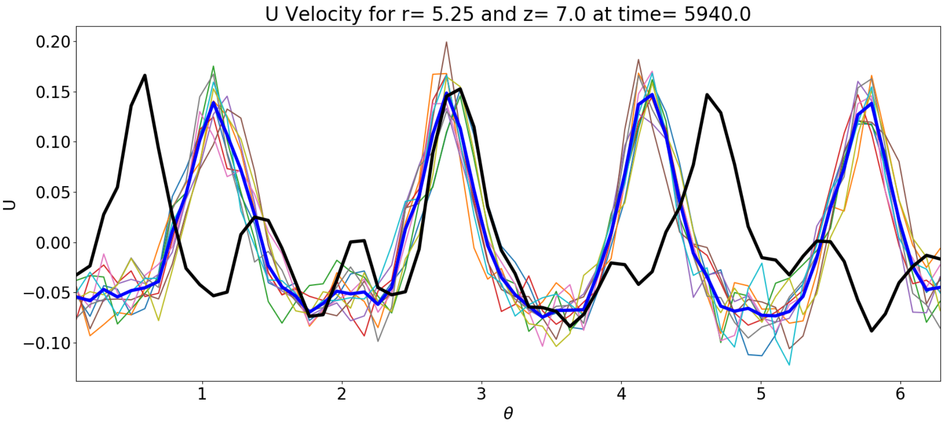
For the last test 576 random points (PPm-10) were chosen throughout the model giving 2304 observations for DA. These observations points were used at every DA for the study. Using an additive inflation of 2pert DA was done every minute, the results for the run are shown in figure (7.59). Here the ensemble has got an $m = 4$ wave while the truth has an $m = 3$ wave. Hence the ensemble is not able to track the truth accurately. The evolution of the ensemble at $r = 5.25cm$, $z = 7.0cm$ at $\theta = 0$ is shown in figure (7.59b). As observed towards the end of the simulation the ensemble shows very similar behaviour as the truth. But as there is a difference in the wavenumber of the ensemble and the truth, the low resolution ensemble will not be able to track the truth accurately. Maybe a simulation with larger ensemble size can help solve some of these issues.

Figure (7.59c) shows the difference plot for the studies done using Randph and Randpp methods. The plots corroborates what is mentioned in those studies with the system failing to track the system accurately. The runs observed clearly don't line up with the truth early one. The low difference observed towards the end of the runs are a result of the system coincidently having similar values. It seems that the systems end up at the similar area of the sate space.

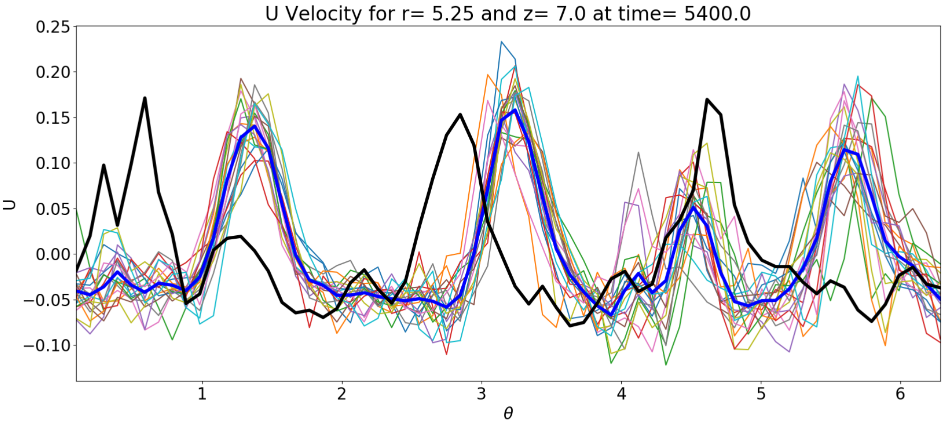
7.8.4 Summary

This section presented results for simulations at 2.5 rad/s. The structures observed at this rotation rate are very volatile with the fluid going from $m = 4$ to $m = 3$. Overall the ensemble found it hard to track the truth. With the 5 minute gap between the observations the ensemble usually gives an $m = 4$ wave which is different to what the is observed in the truth as the end of the run. This makes it hard for EnKF to get an optimal correction for the system. When the time between DA is decreased to one minute, the system does give a stable structure of $m = 3$ wave in most cases, but the systems still behaves differently to what was observed in the truth in terms of rotation rates, etc.

Increasing the ensemble size from 10 to 20 helps with getting the correct structure but the ensemble average still has difficulty tracking the truth, lagging behind it. An even higher ensemble number might help with getting a better tracking. Maybe the number of θ-points should have been increased from 64 to 128 to account for the faster rotation rate leading to information travelling faster around the model. Using this both the higher resolution truth and the low resolution ensemble should be run again to get more accurate data.

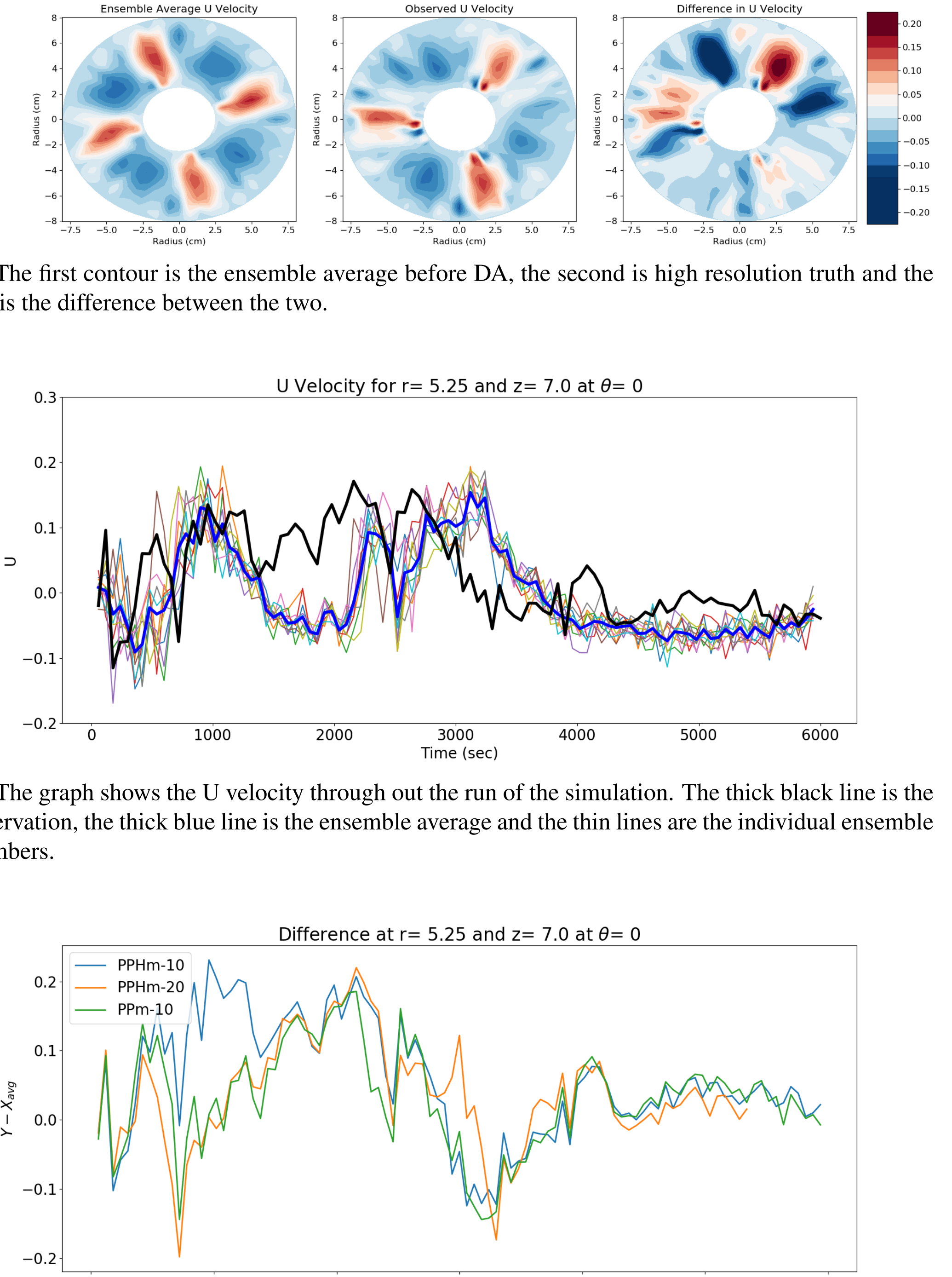


(a) The graph shows the U velocity for an ensemble of 10 with 2304 observations and with DA every minute. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.



(b) The graph shows the U velocity for an ensemble of 20 with 2304 observations and with DA every minute. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

Figure 7.58: Data Assimilation results for 2.5 rad/s using 9 initially chosen random points at each of the 64 θ-locations (Randpph) with an additive inflation of 2pert.



(a) The first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

(b) The graph shows the U velocity through out the run of the simulation. The thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

(c) The difference between the ensemble average and the observations when using 2.5 rad/s and, Randpph and Randpp observation method.

Figure 7.59: Data Assimilation results for 2.5 rad/s with an ensemble of 10 using 576 initially chosen random points (Randpp) with an additive inflation of 2pert, giving 2304 observations in all.

7.9 Results for 3 rad/s

For the last set of tests the rotation rate for the system was increased to 3 rad/s. At this high rotation rate the fluid structures are very chaotic, so the resolution in θ for the high resolution model was increased to 128 points from 64. As seen in section (7.4.4) when no assimilation was done the system shows a stable $m = 3$ wavenumber despite the small scale chaos. But the lower resolution model showed an $m = 4$ wavenumber, so it will be interesting to see whether EnKF can get the correct wavenumber.

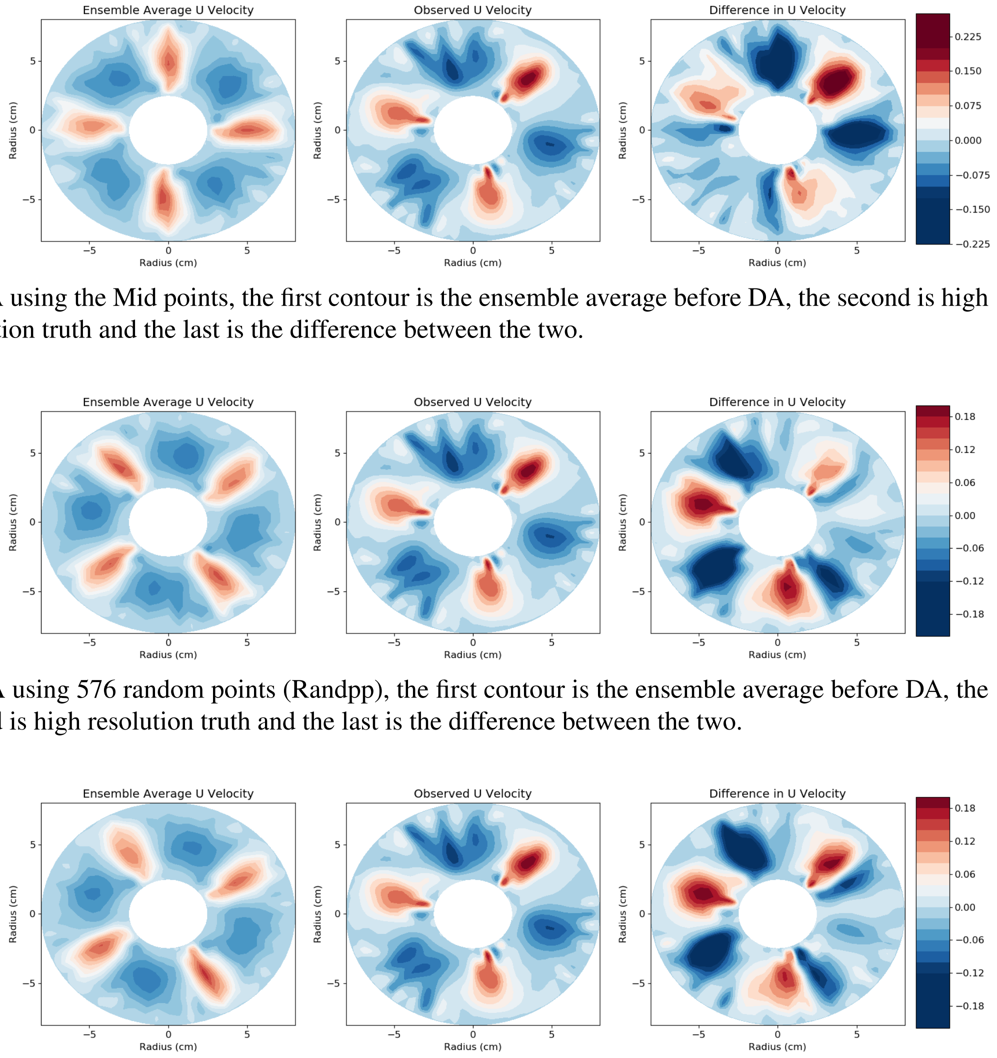
7.9.1 Mid results

For the first test the usual 9 middle points (mid) are observed giving 2304 observations. DA was done every 1 minute and an additive inflation of 2pert was used. The results of the simulation are shown in figure (7.60a). The ensemble system settles at wavenumber $m = 4$ but the truth has a $m = 3$ wavenumber. Even with conducting DA done every one minute the system cannot track the observation correctly and cannot get to the correct wavenumber. To get towards the correct answer the ensemble needs more tuning, with maybe an increase in the ensemble size to give EnKF more data to work with. The only other viable option is an increase in resolution to better model the behaviour of the fluid at this high rotation rate. Difference plot for this study is presented in figure (7.61) and discussed in the next section.

7.9.2 Randpp Results

For this test 576 random grid points (PP-9), giving 2304 observations are used for DA which was done every 1 minute. Looking at the U velocity contour plots in figure (7.60b) the ensemble system again goes to a $m = 4$ wave. This again hampers EnKF from tracking the observation accurately and leads to suboptimal corrections. For the second test the number of observed points was increased to 1920 random points (PP-30), giving 7680 observations in total. This was done to see if the increase in the number of observations could lead to better prediction. Looking at the U velocity contour plots in figure (7.60c), even with this higher number of observations the system still goes to an $m = 4$ wave and cannot track the truth. These issues might be solved with a bigger ensemble sizes and higher resolution. As both cases the ensemble is $m = 4$ wavenumber while the truth is a $m = 3$ system this might be a more stable wavenumber for the low resolution ensemble system.

With all the ensemble runs ending up with a wavenumber of $m = 4$, it is not surprising that the difference graph in figure (7.61) shows an unstable system. It is interesting to note that the mid study has slightly different fluctuating behaviour when compared to the Randpp results. The mid study shows large negative difference around the 1000 and 5000 second mark while the Randpp studies show a large negative difference around the



(a) DA using the Mid points, the first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

(b) DA using 576 random points (Randpp), the first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

(c) DA with 7680 random points (Randpp), the first contour is the ensemble average before DA, the second is high resolution truth and the last is the difference between the two.

Figure 7.60: Data Assimilation results for 3 rad/s with an ensemble of 10 with an additive inflation of 2pert and DA every minute.

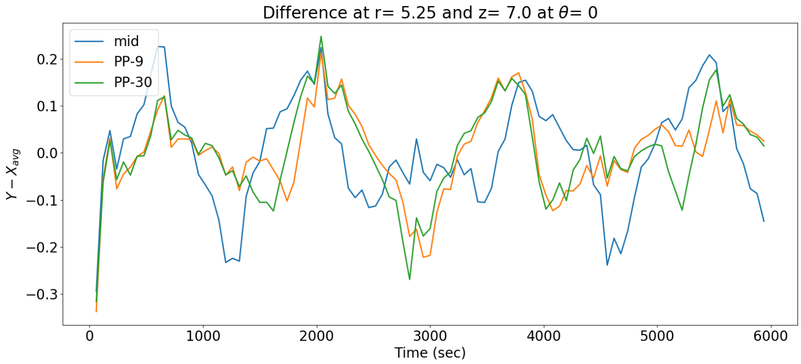


Figure 7.61: The difference between the ensemble average and the observations when using 3 rad/s and, mid and Randpp observation method.

3000 seconds mark. This seems to indicate that the two studies have different behaviour during the run.

7.9.3 Summary

The last rotation rate to be looked at was 3 rad/s. Here the number of θ points was increased to 128 for the high resolution model for better accuracy. But with the low resolution model still having only 64 points in θ it always gave the wrong wavenumber. Looking at the standard deviation periodic increases are observed. It seems as the simulation progress the ensemble spread and converge causing these waves of increase in the deviation. This behaviour might be just due to how the information is transferred around the annulus.

Maybe these issues might be solved with bigger ensemble sizes. The last resort would be to increase the number of θ-points for the low resolution model, to see if the system can recreate structures resembling the truth more accurately. This increase in resolution will increase the computational cost, but as the fluid system at this rotation rate shows a lot of chaotic structures an increase in resolution might be necessary to get accurate prediction.

7.10 Root mean squared error

A way to look at the effectiveness of the different approaches that have been presented in the chapter is to look at the root mean squared error (RMSE) for the tests. The difference plots used in the individual sections can be reused to calculate the RMSE for each run. Using the difference between the truth and ensemble average throughout the run RMSE can be calculated using the equation below.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \overline{X}_i)^2} \tag{7.2}$$

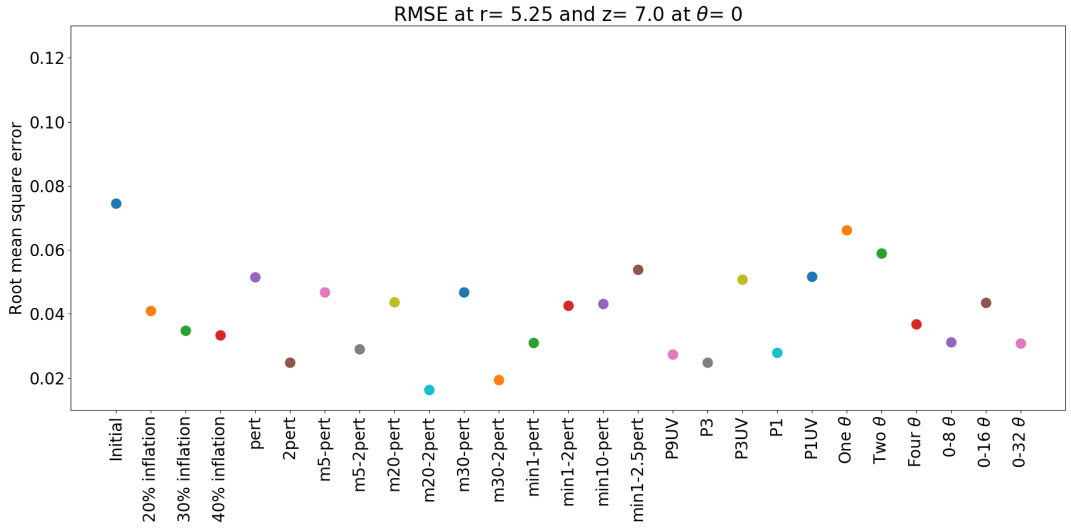
Here Y is the truth, $\overline{X}$ is the ensemble average, i is the DA number and N is the total number of times DA was done. The results for the RMSE for all the studies presented in this chapter are shown in figure (7.62) and (7.63).

Figure (7.62) shows the RMSE for all the runs done at 1 rad/s. Being a steady flow, the RMSE for this rotation rate is low. Figure (7.62a) shows the results of the various runs using the Mid method, and figure (7.62b) shows the results for the various runs using the random point selection methods. There is a lot of variation in the error results, but this seems mostly due to the additive inflation used for the study. In almost all cases the 2pert additive inflation runs show lower errors when compared to runs done using pert additive inflation. Using an ensemble of 20 with 2pert gave the lowest error when compared to all the studies done at various rotation rates and various approaches.

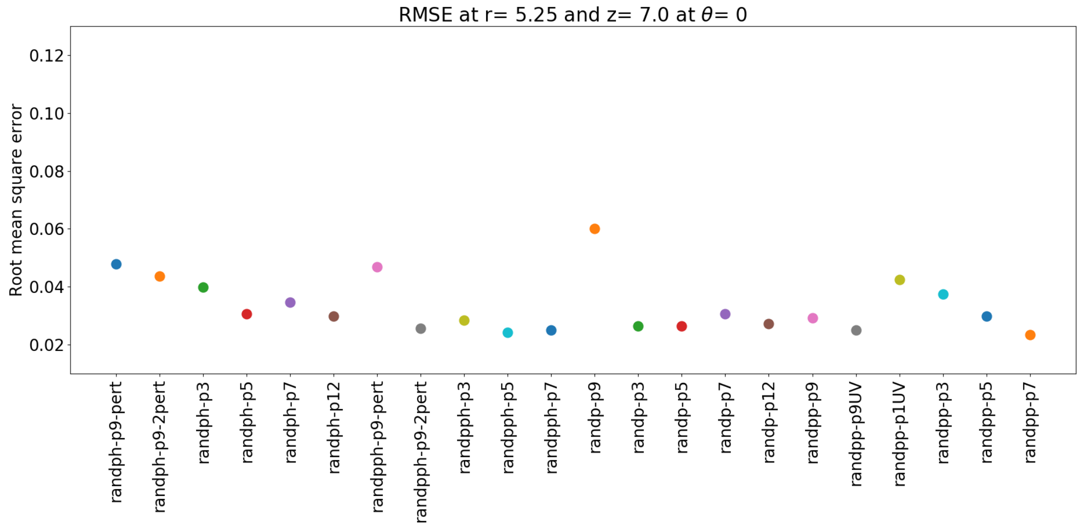
The random point selection results show lower variation as almost all runs were conducted using 2pert additive inflation. Pert runs (randph-p9-pert, randpph-p9-pert, randp-p9) show higher errors then their 2pert counterparts, same as observed in the mid results. In most 2pert cases the errors are quite similar, leading to the belief that no one random method is better than the other. Looking at the randpp results it seems that the number of observations is important when it comes to lower errors.

Figure (7.63) shows the RMSE for all the runs done at 2, 2.5 and 3 rad/s. In all cases the errors in the system are higher then what was observed at 1 rad/s. This seems to be down to the ensemble failing to accurately simulate the more complex flows at these higher rotation rates. Looking at the results for 2 rad/s in figure (7.63a) all the mid runs give a fairly higher error with the mid-min10-2pert giving the lowest error. Looking at the random methods results, the higher error are mostly due to the system not being able to replicate the correct wavenumber or not able to track the truth accurately.

These high error continue for results at 2.5 rad/s and 3 rad/s as plotted in figure (7.63b). As observed in their individual results sections, the ensemble is not able to replicate the same wavenumber observed in the truth, and when they do, they have trouble tracking the truth accurately. The lowest error observed in this set of runs is for 2.5rad rad/s randpp-min1, where the system did not replicate the correct wavenumber and the drift of the working fluid seemed to have stalled. These results re-enforce the idea discussed in the previous sections that more studies need to be done at these higher rotation rates to find ways to reduce the error in the system and get more accurate results.

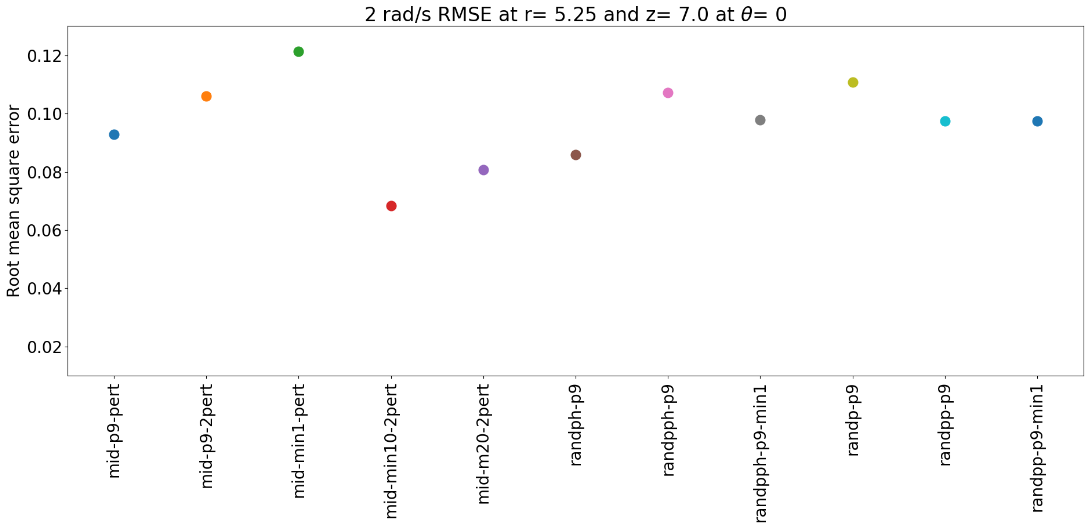


(a) Root mean squared error for mid results.

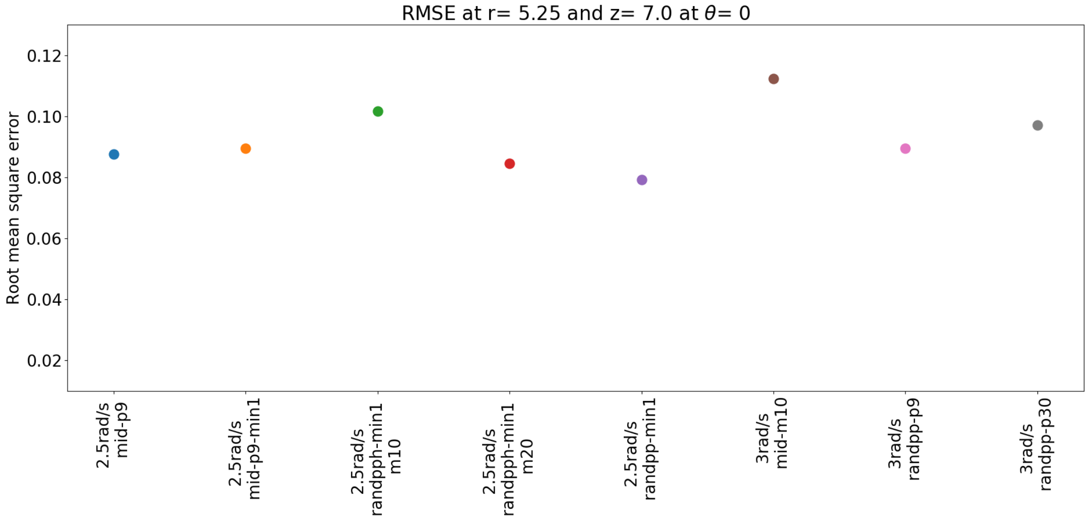


(b) Root mean squared error for random point selection method results.

Figure 7.62: Root mean squared error for 1 rad/s



(a) Root mean squared error for 2 rad/s results.



(b) Root mean squared error for 2.5 and 3 rad/s results.

Figure 7.63: Root mean squared error for 2, 2.5 and 3 rad/s

7.11 Conclusions

This chapter presented the results for Ensemble Kalman Filter twin experiments which were carried out using the MORALS code, which is a rotating annulus code that solves the Navier-Stokes equations in the context of a Boussinesq, viscous, incompressible thermally conducting liquid (see chapter 3 for more details on MORALS). The twin experiment was carried out to test different scenarios for EnKF with the aim to find suitable settings for the model and EnKF. These setting will then be applied for Data assimilation using observations obtained via experiments to test model prediction.

Before any data assimilation was carried out, a benchmark/control study was done, where using an ensemble of 10, the ensemble model was run without any data assimilation. These runs were done at 1 rad/s, 2 rad/s, 2.5 rad/s and 3 rad/s. In all the studies, there was a good spread of the ensembles at the end of the run. Interestingly, similar standard deviations were calculated at the different rotation rates indicating that increasing the rotation rate in the system doesn't affect the spread of an ensemble. There were some differences in wavenumber observed in the the ensembles at higher rotation rates such a 2.5 and 3 rad/s. This indicates that the fluid is unstable and the low resolution model might have trouble giving consistent results.

At 1 rad/s, a lot of studies were done, with slightly differing settings to find optimal parameters which can be used for DA, which lead to accurate results. It was concluded that an ensemble of 10 with DA every 5 minutes and an additive inflation of 2pert gave accurate results in most cases at this low rotation rate. These settings were carried forward when looking at studies at 2 rad/s, 2.5 rad/s and 3 rad/s.

At 2 rad/s, the higher rotation led to very complex behaviour with both $m = 4$ and $m = 3$ wavenumbers observed in the ensemble. In most cases the low resolution ensemble system struggled to replicate the $m = 4$ wavenumber observed in the truth and stayed at the $m = 3$ wavenumber. In many cases the system needed to be tuned to create an $m = 4$ wavenumber. Although decreasing the DA length to 1 minute gave the best result, in many cases the system still strayed away from the truth. Overall EnKF could not track the truth accurately as observed at 1 rad/s.

Going further to 2.5 rad/s, the structures observed at this rotation rate were very volatile with the system going from a $m = 4$ to a $m = 3$ wavenumber. In all cases the system found it hard to track the truth. This was the case even when the time between DA was decreased or the ensemble size was increased. Maybe the number of θ points should have been increased from 64 to 128 to increase the accuracy of the information travelling around the model at this higher rotation rate.

The last rotation rate to be looked at was 3 rad/s. Here the number of θ points was increased to 128 for the high resolution model for better accuracy. But the low resolution ensemble was kept at 64 points in θ. The ensemble always gave the wrong wavenumber of $m = 4$ while the truth was a $m = 3$. Maybe the number of θ-points should have also been increased for the low resolution ensemble, or the ensemble size increased to try and get to the correct wavenumber.

Comparing the results obtained in this chapter with Young and Read (2013), where experimental data was used, the results look promising. At the low rotation rates EnKF does a good job of tracking the truth with the correct tuning. This is in line with what was reported by Young and Read (2013) with analysis correction. More work needs to be done to confirm that the system can track vacillation. When increasing the rotation rate similar problems to Young and Read (2013) start to arise as well. Both system start having problems when tracking the more complex structure with even higher number of observed points, and shorter DA times. More tuning is needed using EnKF in the study presented here for the higher rotation rates to get accurate results along with other changes suggested in this section.

Chapter 8

EnKF using MORALS and PIV Observations

8.1 Introduction

Data assimilation (DA) has become very important in fields such as weather prediction and oceanography (Evensen, 2003). As such, the success of a DA method is based on if it can successfully predict and track observations that are seen in real life phenomena. In chapter (7), a twin experiment was conducted to understand how the Ensemble Kalman Filter (EnFK) method behaves under different conditions. This included scenarios where the number of observed vectors is changed or the time between DA is changed. The lessons learned in that chapter were used to setup a study where EnFK was applied to a set of data gathered from an experiment to try and see if the model can predict the same results.

Using EnKF on real life data provides new challenges, as now the amount and quality of the data used as the observation are subject to how the experiment was conducted and how the data was recorded. The amount of data gathered is subject what instruments can be used without hampering the behaviour of the fluid. Hence a probe cannot be used in the middle of the annulus to measure velocity data as it will change the flow of the fluid. The errors in the experimental data are associated with the instruments that were used to measure data and methods that were used to process the data to obtain velocity vectors. As we are dealing with images here to track movement, the sensitivity of the PIV software and finding the correct parameters to be used are important. The results of the experimental data set are discussed in chapter (6), along with how the data was obtained and processed. In this chapter, the processed results of the experiments were used without any further changes for DA.

To predict the behaviour seen in the experiments the Met Office/Oxford Rotating

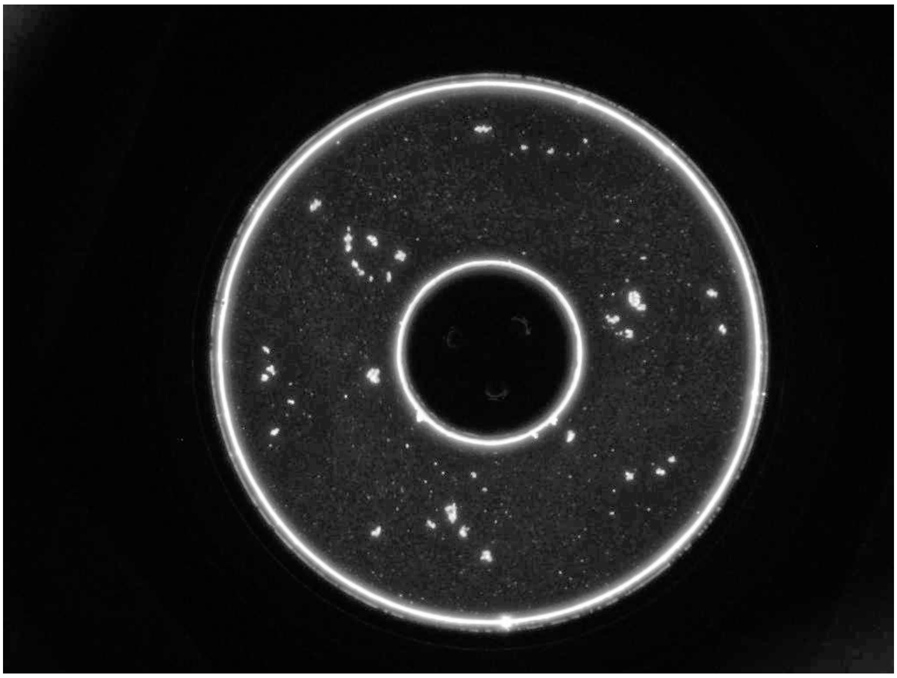


Figure 8.1: Typical image captured from the experiment used for Data processing. This image shows the fluid at level 5. The bright spots are the non-buoyant particles reflecting light.

Annulus Laboratory Simulation (MORALS) code was used, which solves the Navier-Stokes equations for a rotating annulus. This is the same code that was used in the twin experiment. More details on the working of the code can be found in chapter (3). In the study presented in this chapter, the experimental data were used as the truth and the MORALS code was used to create an ensemble of low resolution models which was used to predict the experimental results with the help of EnKF.

The method used for DA with the experimental data and MORALS is described in section (8.2), this includes what data were used for the studies presented in this chapter and how EnKF was implemented. The results of the studies done at 1 rad/s are presented in section (8.3.1), while the results for 2.5 rad/s are presented in section (8.3.2).

8.2 Method

This section explains the different aspects of the study that were combined for DA using a model and experimental data. In the first section, the experiment is described and the processing of the data is summarised. This is followed by a section on how the model works and, in the final section, how the two are combined and DA is done is explained.

8.2.1 The Experiment

The experiment was done using the thermal annulus which was rotated using a motor. The temperature of the inner wall was kept at $18.0 \pm 0.1C$ and the outer wall was kept at $22.0 \pm 0.1C$. The annulus has five slits at different levels, through which light can be projected to observe the behaviour of the fluid in the annulus. PIV was done on the images collected from the different levels and the data saved. For the studies done here, the working fluid was water with 17% glycerol mixture. A summary of the dimensions and parameters of the experiment is given in table (7.1), in the previous chapter. Chapter (6) provides more information about how the PIV was done in section (6.3) and section (6.2) provides more details on the experimental setup.

A typical image captured by the camera mounted on the rig is shown in figure (8.1). The resolution on the camera is 720 by 1080 pixels. After the image was processed a 84 by 63 grid was produced which contained data for the U and V components of velocity. This drastic reduction was due to the PIV software averaging over several points to obtain accurate predictions of the U and V velocity components. As observed in the figure, the camera also captures the area surrounding the annulus. Invalid velocity results calculated by the PIV software from these areas were present in these results. For DA the results outside the annulus have to be ignored. Fortunately, this is an easy task as the PIV software has already marked the data which was present inside the annulus. After discarding the points outside the annulus this gave 4032 velocity points at each level that could be used for DA.

As only one level of the experiment was observed at a time, this meant that unlike in the twin experiment (chapter 7), where the data from the whole model could be observed and could be used for DA, one level in z can be used for DA using the experimental data. To use data from other levels interpolation would needed to be done. Upon looking through the results from the different levels (chapter 6), it was decided that level 4 gave the most accurate results. Level 4 sits at $9.7\,\text{cm}$ from the bottom of the annulus. This is near the top of the annulus and hence has less fluid blocking the camera's view, so this leads to a more accurate picture for the software to calculate the U and V velocity components. Since the PIV software does not give an error for its calculations, it was decided that the same error used in the twin experiment should be used for a start (R=0.08,0.03,0.03,0.2) (section 6.4.1 for more info).

8.2.2 The Model

As mentioned in the introduction, the MORALS code is used to predict the behaviour observed in the experiments. The MORALS code solves the Navier-Stokes equations for a rotating annulus in axisymmetric 2D and full 3D. More details on the working of the code can be found in chapter 3 or on the AOPP website at MORALS (2020).

For the studies presented in this chapter, an ensemble of low resolution models was run to see if they can replicate and predict the behaviour observed in the experiments. The resolution of the low resolution model was 24 points in the r and z axis but the resolution in the θ axis varied between 64 and 128 points. The model uses a stretched grid in the r and z axis to account for boundary conditions near the inner and outer wall. Table (7.1) presents all the dimensions and parameters for the model which are the same as the experiment.

Due to the stretched grid, the points from the low resolution ensemble do not line up with the experimental data. In the low resolution model, the closest corresponding point to level 4 which will be used for DA is at $10.35\,\mathrm{cm}$ from the bottom of the annulus. This is a difference of $0.65\,\mathrm{cm}$ from the height of $9.7\,\mathrm{cm}$ which is what level 4 corresponds to in the experimental setup. This also means there will be a difference in the behaviour observed at level 4 between the low resolution model and the experimental data. No interpolation was done for either numerical or experimental data when applying DA, the observational error used for the DA should cover any difference in behaviour between points.

8.2.3 Data Assimilation

In the study presented here, EnKF was used for DA to bring together data from the thermal rotating annulus experiment and the ensemble of low resolution MORALS models. As the experimental data already exists, this will just be imported in the system when needed for DA.

When the DA code is initialised the timestamps of the experimental images are read into the system. This will be used as a basis to determine the length of the gap between each DA. As mentioned before each level was observed around every 30 seconds. But to compare the study with the twin experiment, the gap between each DA was kept at around 60 seconds. This was done by just looking at the time difference between 20 images. The starting image was given by the user, and this was used as the initial data for the first DA.

The ensemble is now spun-up for 60 seconds to start the DA. When the simulation reaches its designated point, the data from the ensemble and the corresponding experimental data is imported into the system. DA is then done and the model is updated using the corrections. Time is then calculated for the next DA point using the image time gap. This time is calculated after every DA to see how long the ensemble should run. The ensemble is then run till the next DA point. After the corrections are calculated by EnKF additive inflation using a Gaussian of 2pert (R=0.16,0.06,0.06,0.4), as in the twin experiment, was created for each of the variables and added to the results. This ensemble is now used to restart the models.

As mentioned before, after the image is processed, the software makes a 84 by 63 grid in which is stored the U and V velocity data. And since not all the points can be used, as described in section (8.2.1), DA was restricted to 500 random points within the annulus. The closest corresponding points in the model are calculated and read into the system to be used for DA. With both U and V velocity components being used for DA, a total of 1000 vectors are used for each DA.

8.3 Results

8.3.1 Results at 1 rad/s

The first results presented here are for a rotation rate of 1 rad/s. This lower rotation rate, as observed in the twin experiments (section 7.6), provides a very stable platform to test a lot of different parameters for DA. Comparing what is observed in the experimental results (section 6.4.2) and the twin experiments, the structures in the working fluid are similar, with both showing an $m = 3$ wave travelling around the annulus. This similarity indicates that the structures observed in the twin experiment are realistic and even the low resolution model gives structurally accurate results. This bodes well when applying DA to the system, the only worry is that the drift rate of the fluid structures will be different between the experiment and the model. This means DA may find it challenging to get the model to track the experiment accurately for long.

Study with 64 θ points

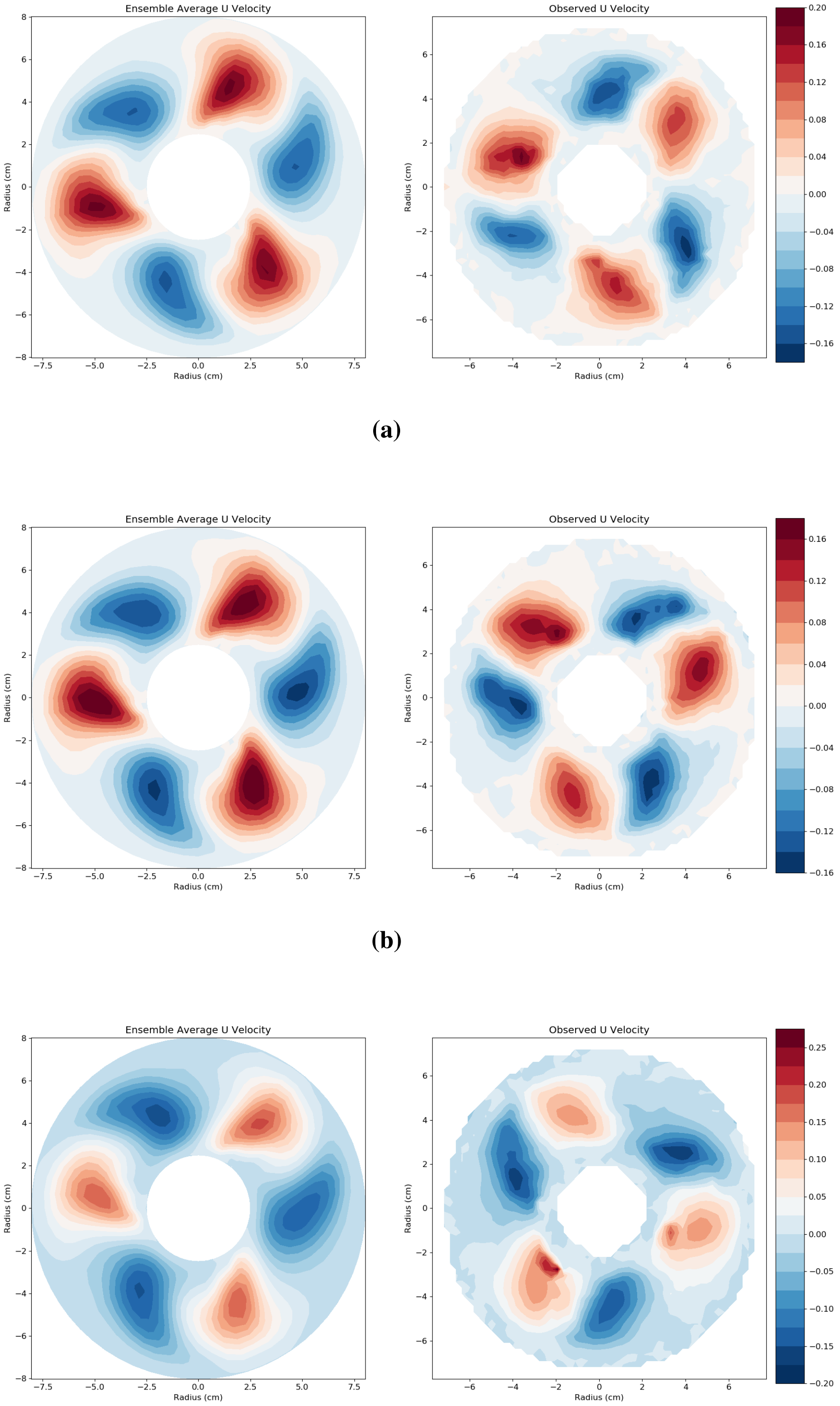
For the first study at 1 rad/s the ensemble had a resolution of 24 points in the radial and vertical direction (r, z) and 64 points the in θ direction. DA was performed every minute, with an ensemble of 10 members, and an additive inflation of 2pert, as described in section (8.2.3). These settings proved to give accurate results in the twin experiments. As mentioned in the methods section only level 4 is being used for DA where a random set 500 points are observed with only U and V velocity components being used for DA.

A contour plot of the simulation is shown in figure (8.2). The first plot in the figure is the ensemble average of the system before DA. The second plot is the observed truth which is used for DA. The figure shows the state of the system for 3 sequential DA. As observed the low resolution model shows an $m = 3$ wave travelling around the annulus, the same as the truth. Looking at the sequence of images the low resolution ensemble lags behind the truth.

Figure (8.3) shows a series of U velocity plots before DA is done, with the thick black line representing the truth and the thick blue line showing the ensemble average. The thin lines are the individual ensembles. As seen clearly for the plots, the system is never able to track the truth. The series shows that the wave in the experiment is travelling at a different rate around the annulus than the low resolution model. Another noticeable thing is that the ensemble amplitude is larger than what is observed in the truth. This might be because of the averaging between points that is done during PIV calculation to obtain the values of U and V. Figure (8.4) shows the standard deviation for the system. The ensemble standard deviation decreases in spread but holds its values after a 100 seconds have passed.

Study with 128 θ points

For the second study, the same simulation is done but the θ resolution is now increased to 128 points. This increase in points should help information to travel more accurately

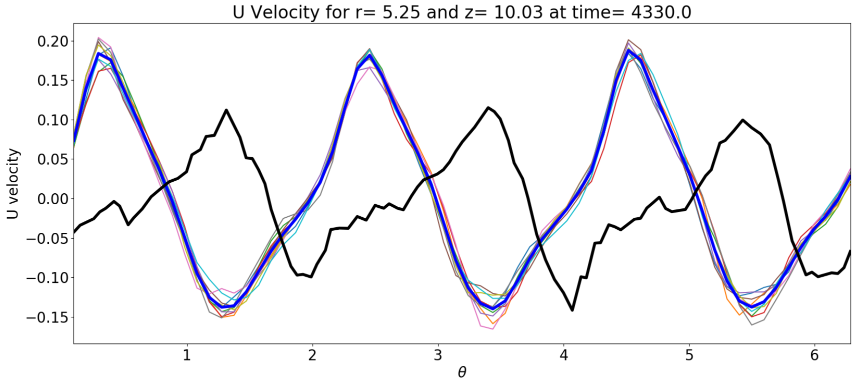


(a)

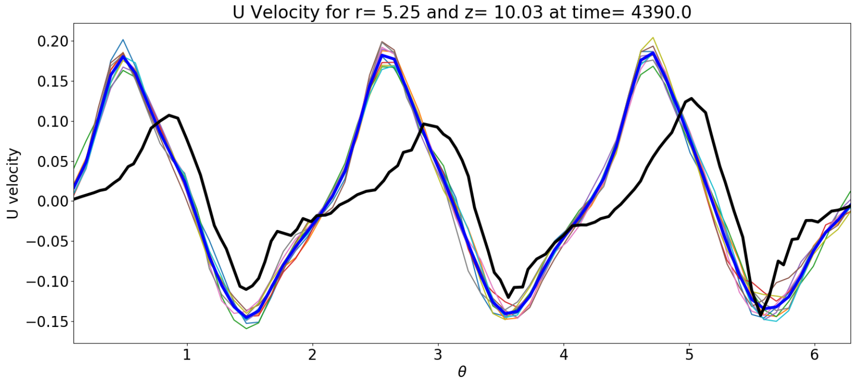
(b)

(c)

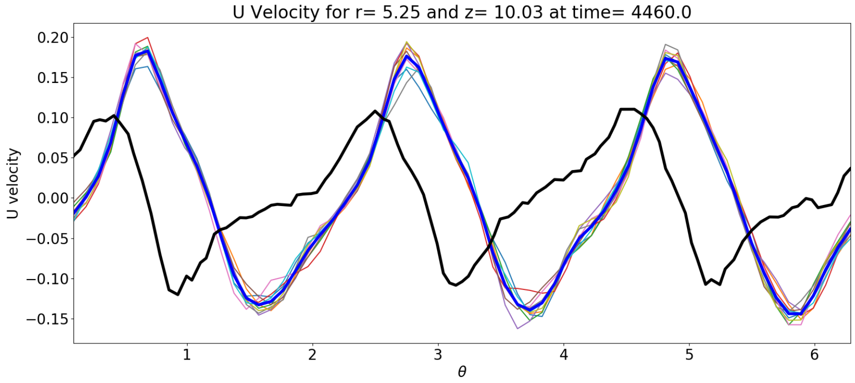
Figure 8.2: Contour plots for a system rotating at 1 rad/s for 3 sequential DA. The plots show the ensemble average before DA and the truth used for DA.



(a) The graph shows the U velocity, the thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.



(b) The graph shows the U velocity, the thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.



(c) The graph shows the U velocity, the thick black line is the observation, the thick blue line is the ensemble average and the thin lines are the individual ensemble members.

Figure 8.3: Data Assimilation results for a system at 1 rad/s with ensemble of 10 with an additive inflation of 2pert where DA is done every 1 minute and 64 θ points are used.

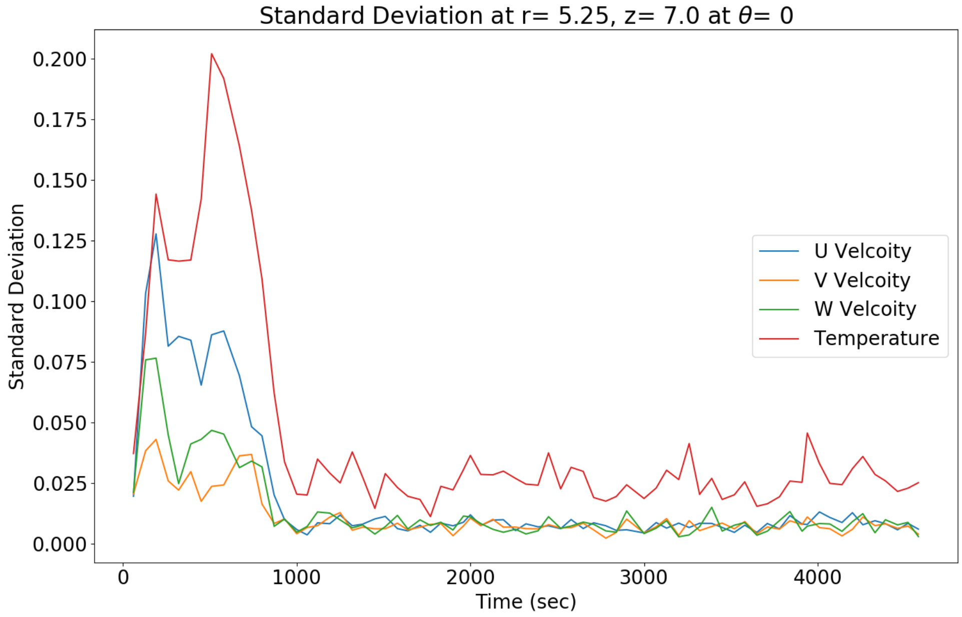


Figure 8.4: Standard deviation for a system rotating at 1 rad/s with 64 points the in θ direction.

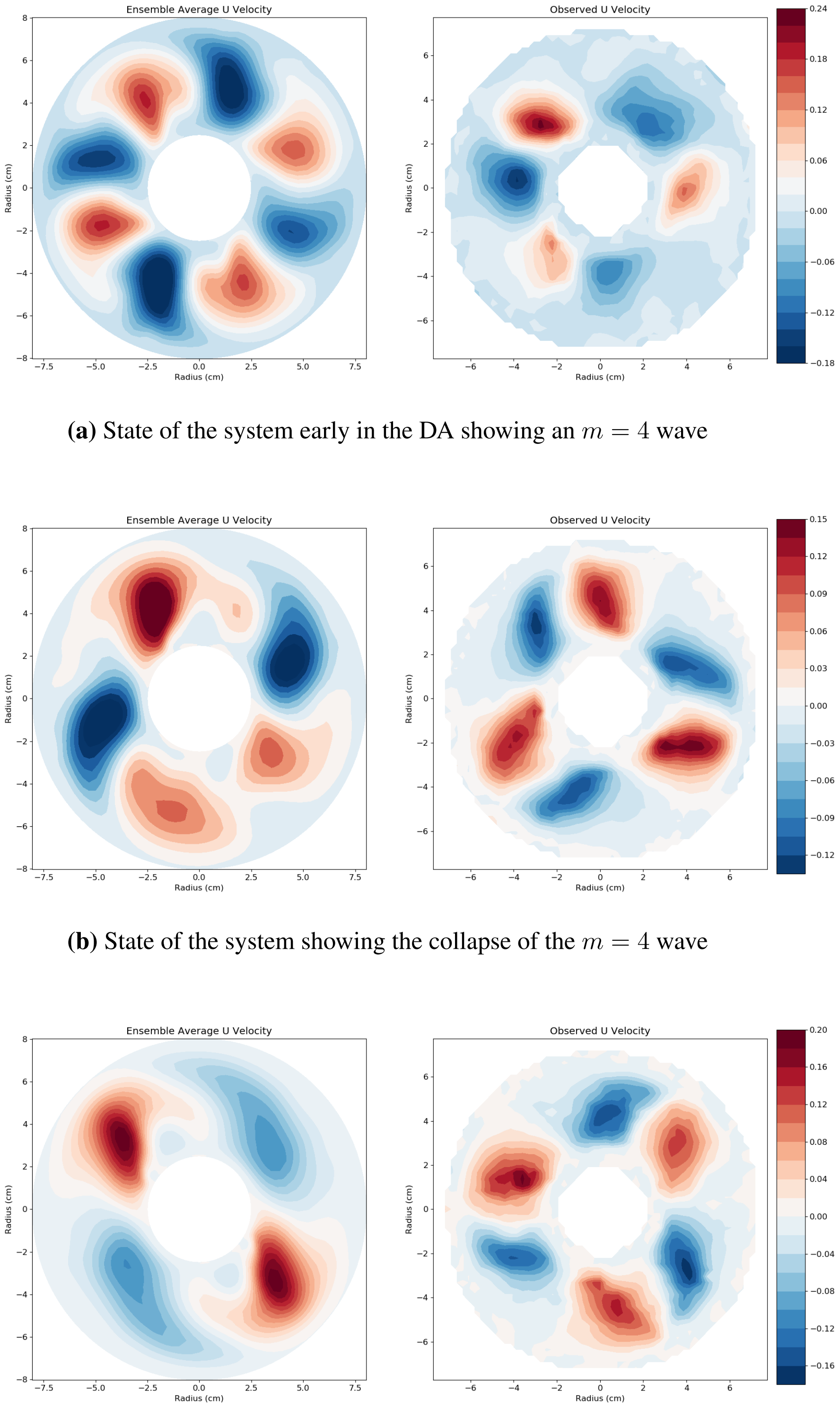
around the annulus. This should hopefully lead to better structures and better drift rates, leading to a more accurate prediction by the ensemble.

The contour plot for the system is shown in figure (8.5), where as before the ensemble average before DA is plotted along with the truth. Unlike the 64 θ point study, where the system was an $m = 3$ wave throughout the run, now the system starts as an $m = 4$ wave and collapses to an $m = 2$ wave. The system then keeps this $m = 2$ wave for the rest of the run. The reason for this collapse could be due to the corrections done by EnKF is making the ensemble collapse. Without DA the system might have settled on a $m = 3$ wave just like the truth and how is did when no DA was done in section (7.4.1). The corrections want to change the system from an $m = 4$ wave to an $m = 3$ wave, but by trying to correct this the system starts to collapse further and forms an $m = 2$ wave.

The U velocity graph for the system is shown in figure (8.6a). As observed in the contour plots the system is an $m = 2$ wave travelling around the annulus. As this never matches the $m = 3$ wave from the truth it will never be able to track the truth. As before the ensemble has a much larger amplitude than the truth and the ensembles have all converged to an $m = 2$ wave. The standard deviation of the system is plotted in figure (8.6b) and doesn't show much difference compared to what was observed previously with only 64 points in θ.

Sensitivity test with lower errors

The experimental value of U and V velocity obtained through PIV are lower then the

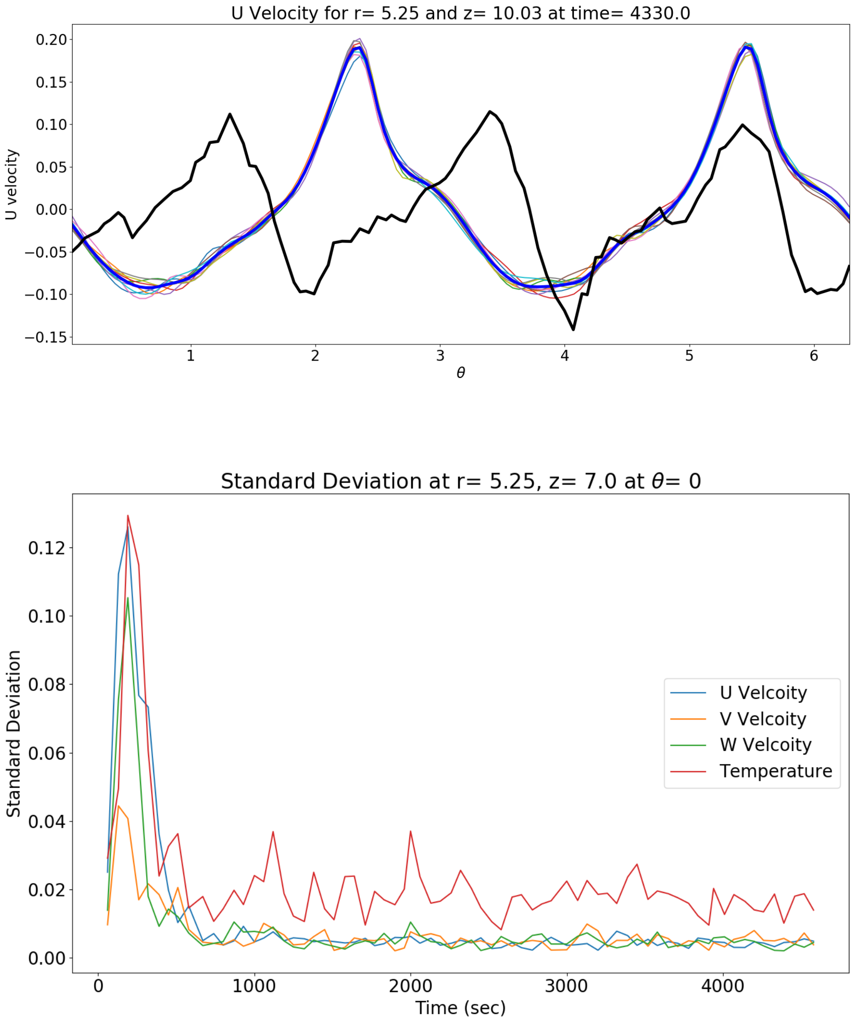


(a) State of the system early in the DA showing an $m = 4$ wave

(b) State of the system showing the collapse of the $m = 4$ wave

(c) State of the system showing an $m = 2$ system

Figure 8.5: Contour plots for a system rotating at 1 rad/s with 128 θ points. The plots show the ensemble average before and after DA followed by the truth used for DA.



(b) Standard deviation

Figure 8.6: Data Assimilation results for a system at 1 rad/s with ensemble of 10 with an additive inflation of 2pert where DA is done every 1 minute with 128 θ points.

ones obtained in the twin experiment due to the accuracy of the equipment and software. In some cases the PIV calculated values were smaller then the observation error used. This can lead to very suboptimal corrections as $Y - X^f$ will be very big. To compensate for this the Kalman gain needs to be modified. To see how the observation error affects the value of the Kalman gain a simple sensitivity test was carried out where the error values were reduced.

In the first test, the observation error was reduced to 10% of the value used. Figure (8.7) shows the results for this test. Here the ensemble is a $m = 3$ wave accurately representing the truth structurally but the system has a lot of problems tracking the truth. The drift rate of the structures are different compared to the truth and the ensemble is never able to stay in a similar space as the truth for long.

In the second test the observation error was reduced further to just 1% of the original. Figure (8.8) shows the results for this test. Here the ensemble has collapsed to a $m = 2$ wave. The behaviour here is similar to what was observed in the previous section where the resolution was increased (figure 8.5). The system starts with a $m = 3$ and collapses to a $m = 2$ wave and can never recover.

It was thought that reducing the observational error would mean that the ensemble covariance would play a bigger role in calculating the Kalman gain. But it seems that reducing the observational error alone does not have a positive effect on the system. These changes might have to be tested with using different multiplicative and additive inflation, and ensemble sizes to gain a better understanding of how EnKF behaves. Increasing the observational error might be the obvious next step, but as Kalman gain $K = PH^T \cdot (R + HPH^T)^{-1}$, a larger R will lead to a smaller Kalman gain which in turn will lead to a smaller correction. As the ensemble is far from the truth this small correction will not lead to more accurate model.

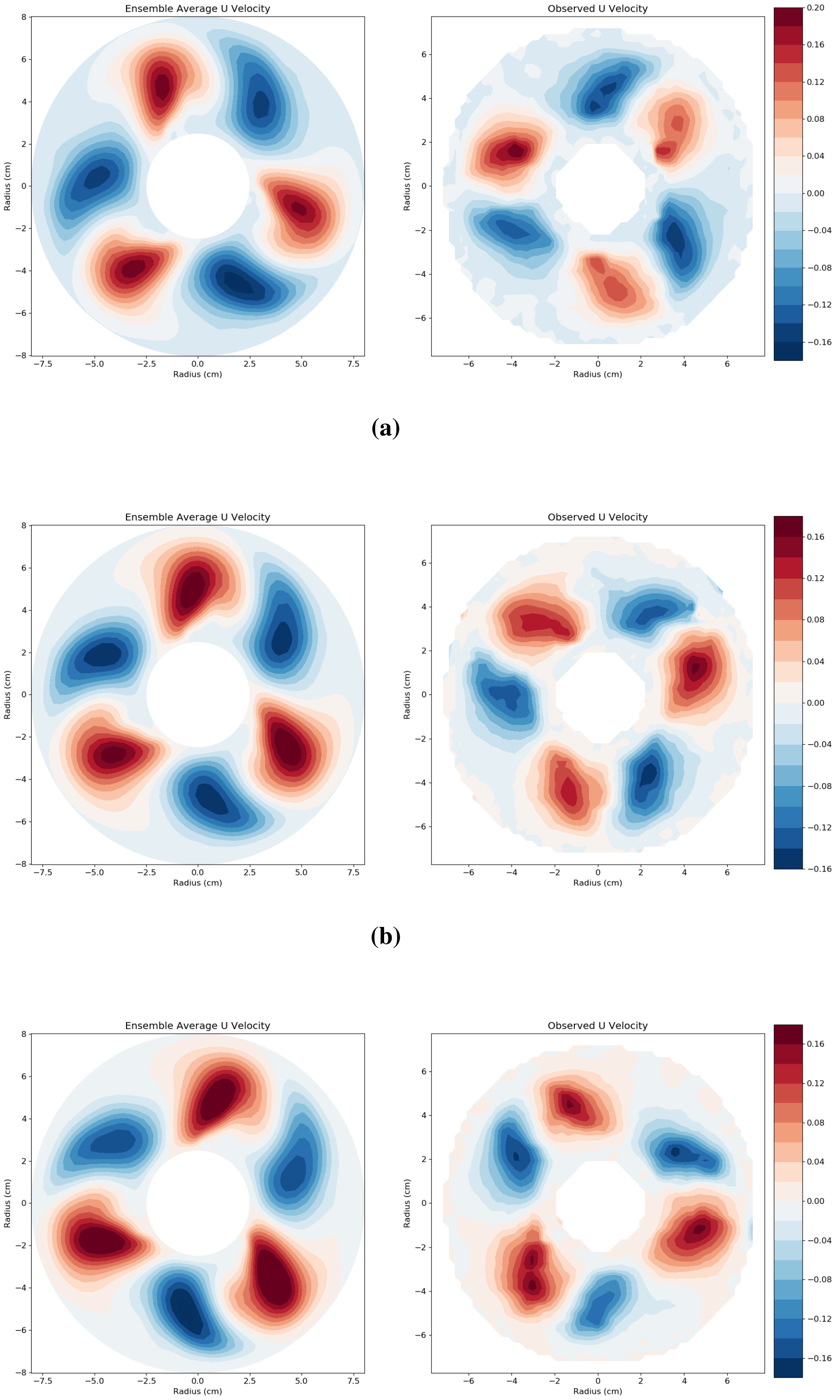
8.3.2 Results at 2.5 rad/s

The rotation rate of the system is now increased to 2.5 rad/s. As observed before in the twin experiments, this higher rotation rate can lead to unstable waves. Similar unstable behaviour is observed in the experiments as well, causing a lot of problems for the PIV software to accurately give values of U and V velocity components. As before, the simulation is run with an ensemble of 10 and with additive inflation of 2pert.

Study with 64 θ points

The contour plots for this run are shown in figure (8.9) where the ensemble before DA is plotted along with the truth. The plots clearly show that the ensemble is a $m = 3$ wave while the truth is a $m = 4$ wave. The system is not able to replicate the truth structurally, which may be down to the resolution in θ. This higher rotation rate might need higher resolution to pass information around the annulus more accurately.

Looking at the series of plots, the truth again seems to travel at a different rate around the annulus than the ensemble. There is also a lack of intensity in the colours

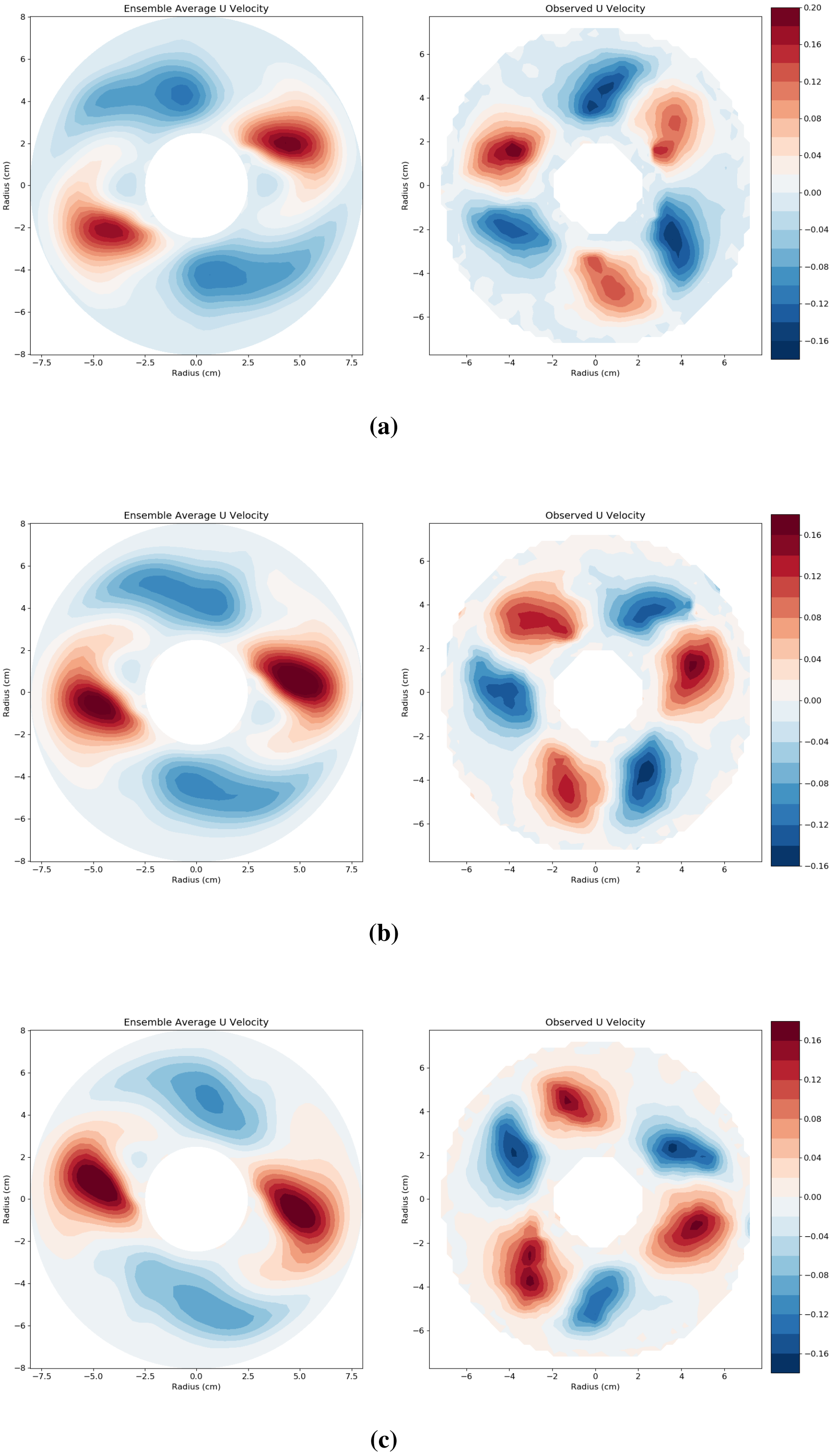


(a)

(b)

(c)

Figure 8.7: Contour plots for a system rotating at 1 rad/s for 3 sequential DA with 10% error. The plots show the ensemble average before DA and the truth used for DA.



(a)

(b)

(c)

Figure 8.8: Contour plots for a system rotating at 1 rad/s for 3 sequential DA with 1% error. The plots show the ensemble average before DA and the truth used for DA.

in the truth contour plots which indicates low values for the U velocity component. It seems the higher rotation rate causes problems for the PIV software when it comes to accurately tracking the buoyant particles to calculate the values of U and V velocity. Obviously different setting in the PIV software need to tested to obtain a better result. These less accurate values of U and V lead to less accurate corrections by EnKF, which might also be one of the reasons for the wavenumber not being accurate. This is also reflected in the structures disappearing and reappearing in the contour plots for the truth.

A plot of U velocity is shown in figure (8.10a), and as observed before, the values of the truth are much lower when compared to the ensemble. These low values are due to the PIV software having issues tracking the fluid accurately with the higher rotation rate. Looking at the graph it is not clear what the wavenumber of the truth is but as the graph corresponds to the contour plot in figure (8.9c) it should be an $m = 4$ wave. The standard deviation for the ensemble is plotted in figure (8.10b). Here the deviation shows regions where the ensemble spread became very small and places where is spread out again. These changes are probably due to the changes to the wave structure due to the higher rotation rates.

Study with 128 θ points

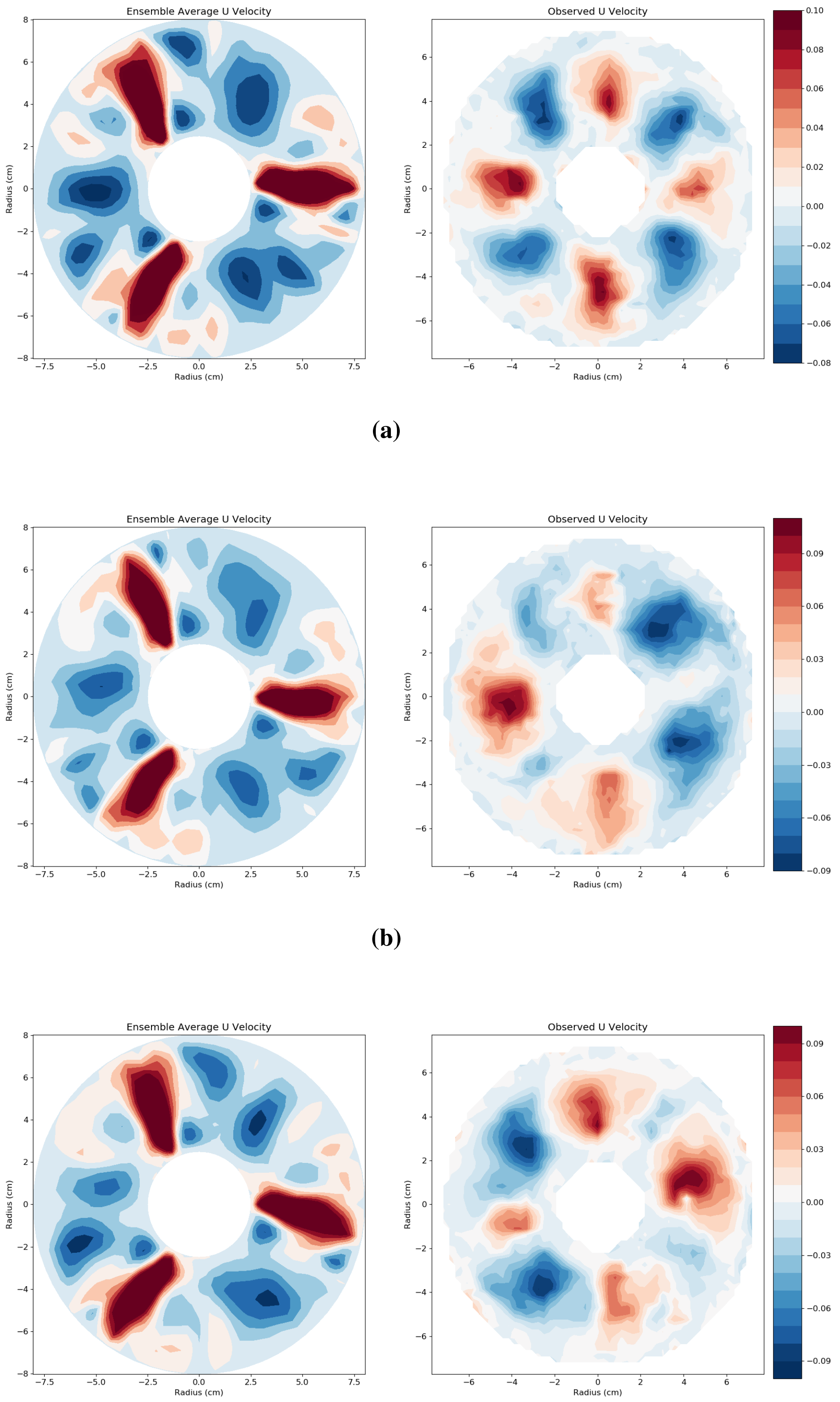
The study at this higher rotation rate was repeated but with an increased θ for 128 points to help cope with the higher rotation rate. The contour plots for the system are plotted in figure (8.11), showing the ensemble system before and after DA along with the truth. Unfortunately, even with the higher resolution in θ, nothing significant has changed in the ensemble system. It still has an $m = 3$ wave travelling around the annulus while the truth shows an $m = 4$ wave travelling around the annulus. The values of the U velocity in the truth are again much lower than the ensemble average. The reason for this is the same as mentioned before.

The U velocity graph is plotted in figure (8.12a) showing the ensemble U velocity average along with the ensembles and the truth. As seen before, the truth is very small compared to the ensemble and no clear wavenumber can be observed while the ensemble showsa $m = 3$ wave. The standard deviation for the system is given in figure (8.12b). Here a regular fluctuation is observed for all variables.

8.4 Conclusion

In this chapter, results were presented for DA done using data from PIV observations from experiments. DA was done for a thermal rotating annulus rotating at 1 rad/s and 2.5 rad/s with 500 random grid points being used to calculate and correct the low resolution ensemble. An ensemble of 10 was used with an additive perturbation of 2pert for all studies with DA being done every 1 minute. It is clear from the results that EnKF has failed for various reasons.

At 1 rad/s the ensemble can replicate the fluid structures observed in the PIV data with an $m = 3$ wave being observed in both studies. But the ensemble is never able to

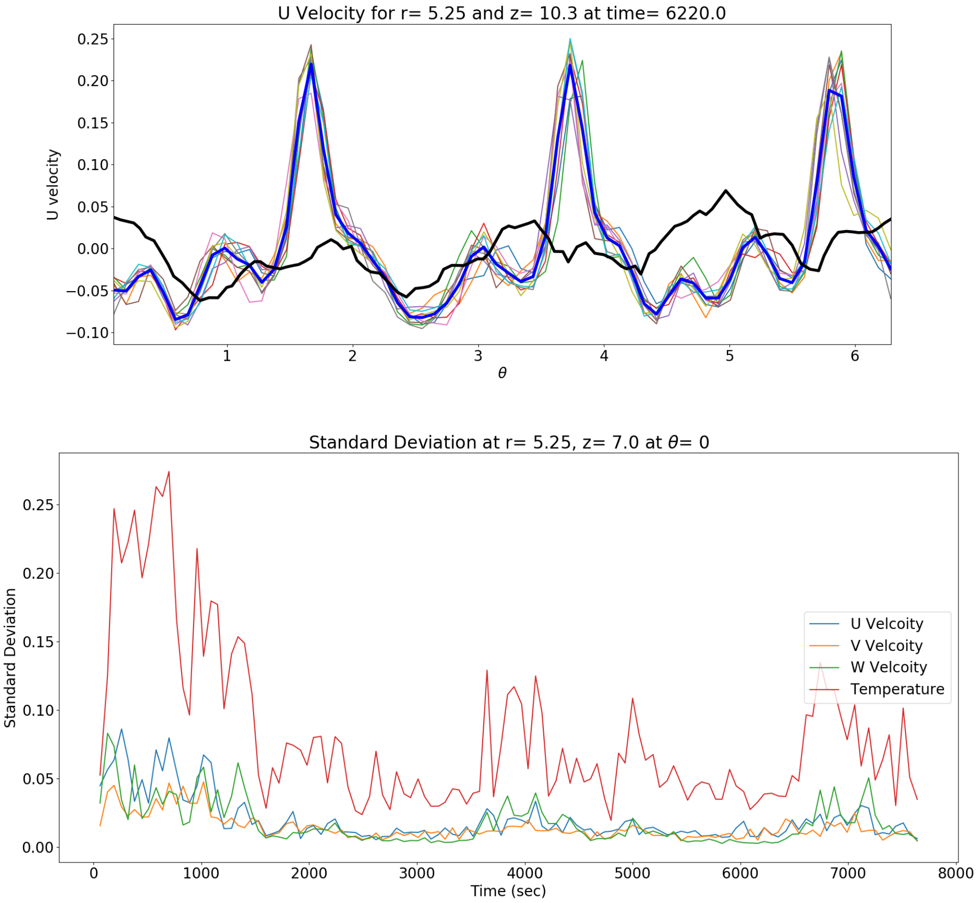


(a)

(b)

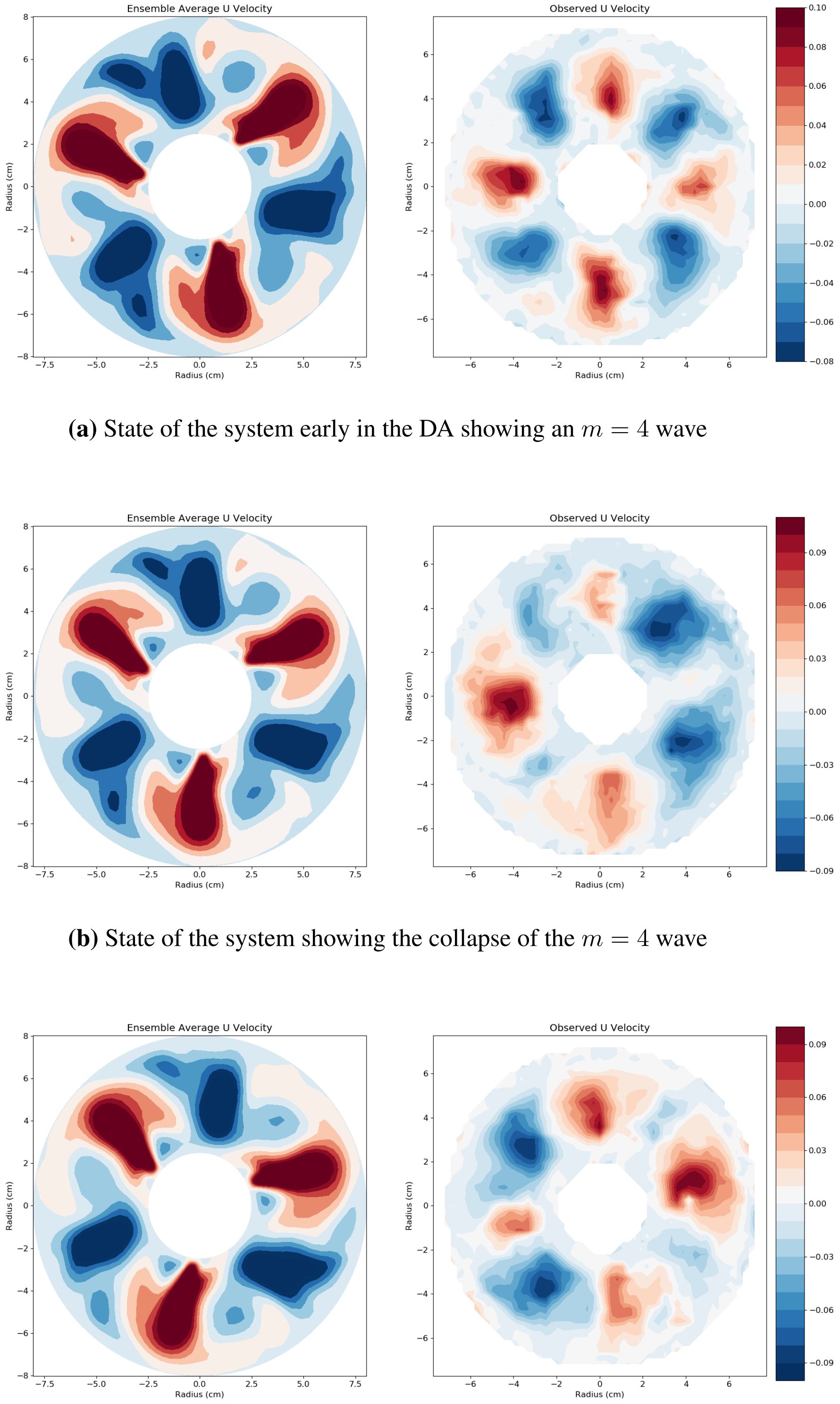
(c)

Figure 8.9: Contour plots for a system rotating at 2.5 rad/s. The plots show the ensemble average before and after DA followed by the truth used for DA. Here 64 θ points were used.



(b) Standard deviation

Figure 8.10: Data Assimilation results for a system at 2.5 rad/s with ensemble of 10 with an additive inflation of 2pert where DA is done every 1 minute with 64 θ points.

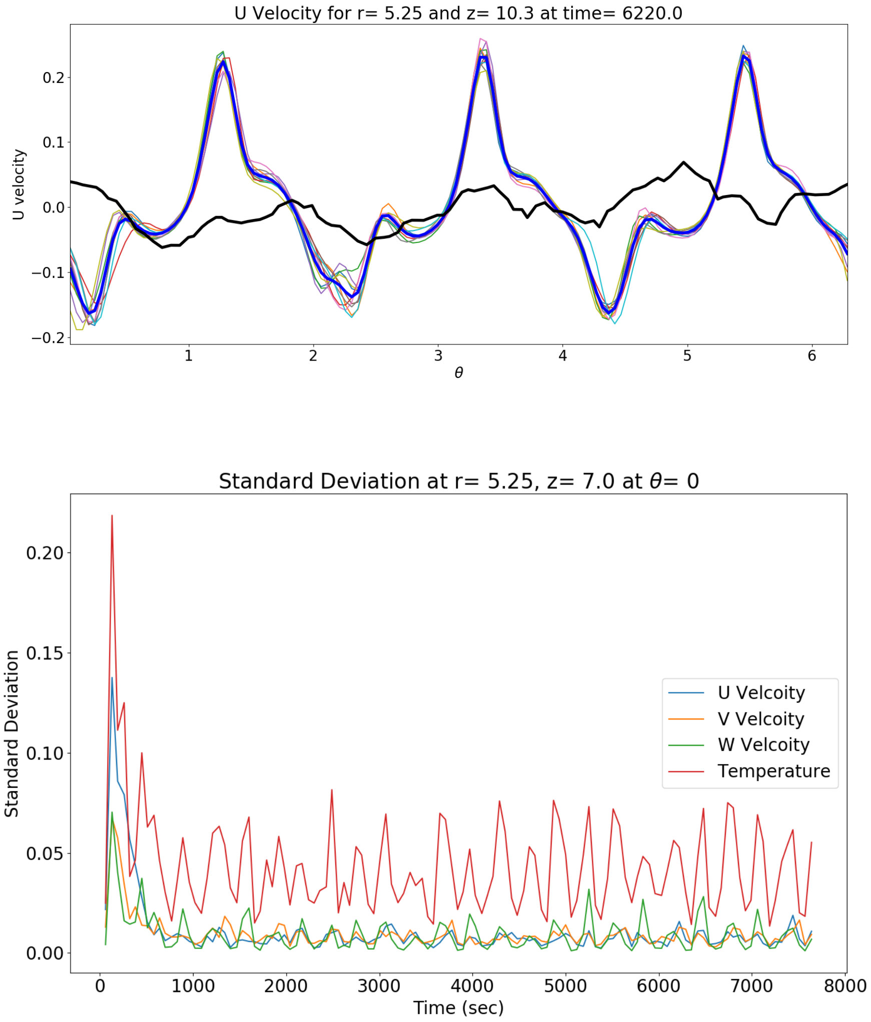


(a) State of the system early in the DA showing an $m = 4$ wave

(b) State of the system showing the collapse of the $m = 4$ wave

(c) State of the system showing an $m = 2$ system

Figure 8.11: Contour plots for a system rotating at 1 rad/s with $\theta = 128$. The plots show the ensemble average before and after DA followed by the truth used for DA.



(b) Standard deviation

Figure 8.12: Data Assimilation results for a system at 2.5 rad/s with ensemble of 10 with an additive inflation of 2pert where DA is done every 1 minute with $\theta = 128$.

track the truth with the latter travelling around the annulus faster. The slow drift velocity could be due to the low θ resolution of just 64 grid points. The test was repeated but with an increased θ resolution of 128 points. Here the ensemble starts with an $m = 4$ wave and then collapses to an $m = 2$ wave. This collapse of the wavenumber might be due to the DA corrections. With this difference in wavenumber between the ensemble and truth, the ensemble was never able to track and predict the truth.

The rotation rate was increased to 2.5 rad/s for the next study. Here the PIV data shows an $m = 4$ wave but in the first test with a θ resolution of 64 the ensemble shows an $m = 3$ wave. The ensemble is never able to move up to an $m = 4$ wave. Even when the θ resolution is increased to 128 points the ensemble average is still a $m = 3$ wave. In both cases, the ensemble is not able to track the truth.

One thing to note about the PIV data is that the amplitude of the values of the truth are smaller than what is observed in the ensembles. This is understandable, as when PIV is carried out it uses around 12 pixels to calculate the values of U and V velocity components so some accuracy is lost. This difference is small at 1 rad/s but at 2.5 rad/s the difference is too big. Looking at the U velocity graphs, the wavenumber of the truth is indistinguishable and the information can only be ascertained by looking at the contour plot. This shows the need for the PIV data to be reprocessed with different parameter settings in the hope of obtaining more accurate results. It is clear that the same settings cannot be used to calculate values for U and V velocity for different rotation rates.

When compared to the results presented by Young and Read (2013), a lot of shortcomings can be observed in the study presented here. In all studies the drift rates are different between the ensemble and the experimental data. In some cases the ensemble does not even show the same structure as the truth either. Other problems with the study seem to stem from the error used for the truth. As the PIV software did not provide an error is was decided to use the same error as used in the twin experiments. This certainly needs to change and a more accurate error for the PIV data needs to be calculated. Maybe the PIV calculations can be repeated to obtain a standard deviation of how its calculated values differ. Interpolation also needs to be done on the truth so that they line up with the low resolution ensemble.

Another twin experiment also needs to be carried out where only one level is observed. This will be a testing ground to see how EnKF copes with such limited data from the experimental data set. The parameters that work can then be used for testing EnKF with the experimental data. By the results seen in this preliminary study, the θ resolution should be kept at 128 points, especially for the higher rotation rate. This should help with reproducing a better structure formation and better drift rates for the ensemble. Another option is to increase the ensemble number to see if this can bring about the more accurate results. Tuning of the inflation parameters could be done, but this is more of an art than a science and can take a long time before reasonable results can be obtained.

Chapter 9

Conclusion

All computer models are imperfect and this problem is exacerbated further when studying complex fluid behaviour. Therefore it is important to reconcile the differences in the results produced by modelling and observations. Many studies have used data assimilation (DA) to bridge some of the shortcomings of models. DA has been used for variety of applications such as NWP and oceanography (Evensen, 2003) which have lead to better prediction.

This thesis presents results for a study where the Ensemble Kalman filter (EnKF) was used to try and accurately predict behaviour observed in the thermal rotating annulus. This was done using the MORALS numerical code and thermal rotating annulus experiments. The MORALS code is a Fortran code maintained at the Atmospheric, Oceanic and Planetary Physics (AOPP) group at the University of Oxford. The code solves the Navier-Stokes equation for a thermal rotating annulus for a 2D axisymmetric model and a full 3D model. The experiments that were used as observations were also done as the AOPP using a thermal rotating annulus.

The aim of the study was to apply the knowledge gathered about EnKF during this study to tokamak data obtained at the Culham Centre for Fusion Energy (CCFE). Here EnKF would be used to validate models which predict instabilities in tokamak plasma. The models would be used for parameter estimation on the data to get an insight on the parameters of the plasma and later to see if real time predictions can be made of these instabilities that are observed.

9.1 Summary

The first two results chapter were aimed at gathering the initial understanding of DA and rotating thermal annulus respectively. Chapter 5 used the Lorenz model as the test bed to better understand EnKF with chapter 6 presenting the results observed in the thermal rotating annulus for different rotation rates. The last two results chapters were all about applying EnKF the thermal rotating annulus. First in chapter 7 twin experiments was done where using accurate observations from a high resolution run EnKF was used along with a low resolution ensemble to try and accurately predict the high resolution behaviour. Different studies were done here to understand EnKF's behaviour when used with a 3D model. In chapter 8 EnKF was applied to experimental data to see if EnKF help predict real life behaviour with low resolution ensemble.

9.1.1 Testing with Lorenz

Before EnKF can be done using the experimental data a good understanding of the method's behaviour and limitation are needed. For this, the Lorenz model (Lorenz, 1963) was used as it has a very complex and chaotic structure at certain parameter values. A high resolution run was performed and its data stored to create a set of observations which were used for all the tests in the study.

In the first set of tests, the ensemble sizes were changed to test the predictive power of EnKF. Increasing the ensemble size dramatically increased the accuracy of the system. The ensemble has some issues with giving accurate prediction when the observation oscillated between the two fixed point regions. With larger ensemble sizes EnKF does a better job of tracking in these areas and the speed at which the system corrects when the ensemble divergence also increases with larger ensemble sizes.

In the second set of experiments, multiplicative inflation was used along with different ensemble sizes to test EnKF. Inflation leads to better tracking of the observations by the ensemble. Smaller ensembles needed higher inflation and larger ensembles needed smaller inflation when it came to better tracking the observations. This is because of a larger ensemble sample more of the state space, hence lead to a better covariance and Kalman gain. Inflation is more useful when used for smaller ensemble sizes as the user is saving on the computational cost of simulating additional ensemble members.

In the third set of experiments, the number of time steps between the DA points was changed. As theorised, with the larger interval between DA the system finds it harder to track the truth. This is because the ensemble members have more time to diverge. But in most cases using multiplicative inflation can help stop the system from diverging from the truth. With the longer interval between DA, the amount of inflation needed to accurately track the truth is higher than what was used with the shorter interval.

Finally, a study was done on the capabilities of EnKF for parameter estimation. Here the parameters of σ, r and b were predicted by EnKF along with the normal variables x, y and z. Overall EnKF gets the predictions close to the actual values used for the parameters. With smaller ensemble sizes inflation was needed to increase the accuracy of the estimated parameters. But as the ensemble size was increased the need for inflation to get an accurate estimate decreased. These results are comparable to previous studies such as Annan and Hargreaves (2004), where a much larger ensemble, up to 100 members, was used to obtain close predictions.

All results obtained here are equivalent to what has previously been observed in past studies such as Evensen (1997); Miller (1994) for studies on ensemble sizes and inflation and Annan and Hargreaves (2004) for study on parameter estimation.

9.1.2 Experimental results

The aim of these experiments were to test the new equipment that had be installed at AOPP and to get data that can be used for DA. Young and Read (2013) had used an older experiment for DA where the rotation rate of annulus changes every 20 minutes. But as this is the first time EnKF will be applied to this system it was decided to do a long run to get stable data at different rotation rates.

The experiments were done to create observations for EnKF and were done at a rotation rate of 1 rad/s, 2.5 rad/s and 3 rad/s. Overall the experiments showed behaviour that is expected at the various rotation rates. It was very difficult to observe the behaviour of the fluid at lower levels consistently. But much clearer behaviour was observed at higher levels. This was due to the fluid at higher levels obscuring the camera's view when it comes to observing the lower levels.

The experiment at 1rad/s showed a steady m=3 wave that travelled around the annulus. This could be observed in the images as well as being confirmed by the PIV data. At lower levels, the behaviour is hard to garner as observed in both the images and PIV but level 3 onwards the observations become clearer. At a rotation rate of 2.5rad/s increases the observed wavenumber to m=4 around the annulus. Observations at lower levels are still poor and get better with the height of the observed level. The Hovmöller plots for this run show the m=4 wave very clearly and the structures can be followed throughout the run.

The last experiment at 3rad/s caused a lot of problems for the PIV software to generate accurate observations. At lower levels the fluid showed minimal movement with only the higher levels showing clear fluid circulating around the annulus. This was confirmed by the PIV plot and the Hovmöller plots. A m=4 wave is observed at higher levels and the wave slowing drifting around the annulus. There were also issues with clumping of seeding particles along the inner cylinder. It seems that the high rotation rate was pushing the particles towards the inner and outer wall of the annulus.

Overall the one size fits all approach to PIV settings in DynamicStudio for analysing images does not seem to work well. Future work will have to try and fine tune settings when calculating U and V velocity values for observation taken at different levels and different rotation rates.

9.1.3 Twin Experiment

The next step was to couple EnKF with the MORALS code and conduct a twin experiment to understand the intricacies of DA on a 3D model. Here a high resolution model was used to create data for observation and then an ensemble of low resolution models were used to see if they can replicate the results using EnKF. Tests were conducted at a varying rotation rates of 1 rad/s, 2 rad/s, 2.5 rad/s, 3 rad/s.

A control study was done where the low resolution ensemble of 10 models was run without any data assimilation. Studies were done at 1 rad/s, 2 rad/s, 2.5 rad/s and 3 rad/s. In all the studies, there was a good spread of the ensembles at the end of the run. Interestingly, similar standard deviations are produced for the different rotation rates indicating that increasing the rotation rate in the system doesn't affect the spread of an ensemble.

At 1 rad/s a lot of studies were done with slightly different settings to find optimal parameters which can be used for DA which lead to accurate results. It was concluded that an ensemble of 10 with DA every 5 minutes and additive inflation of 2pert gave accurate results in most cases at this low rotation rate. These settings were carried forward when looking at studies at 2 rad/s, 2.5 rad/s and 3 rad/s.

At 2 rad/s the higher rotation led to a more complex behaviour with both $m = 4$ and $m = 3$ wavenumbers observed. In most cases, the low resolution ensemble system struggled to replicate this $m = 4$ wavenumber and stayed at the $m = 3$ wavenumber. In many cases, the system needed to be tuned to create a $m = 4$ wavenumber. Although decreasing the DA length to 1 minute gave the best results but in many cases, the system strayed away from the truth.

Going further to 2.5 rad/s, the structures observed at this rotation rate were very volatile with the fluid going from a $m = 4$ to a $m = 3$ wavenumber. In all cases, the system found it hard to track the truth. This was the case even when the time between DA was decreased or the ensemble size was increased. Maybe the number of θ points should have been increased from 64 to 128 to increase the accuracy of the information travelling around the model at this higher rotation rate.

The last rotation rate to be looked at was 3 rad/s. Here the number of θ points was increased to 128 for the high resolution model for better accuracy. But with the low resolution model still having only 64 points in θ, it always gave the wrong wavenumber. Maybe the number of θ-points should have also been increased for the low resolution model.

Young and Read (2013); Ravela et al. (2010) are one of the only few studies published about DA and thermal rotating annulus experiments. Young and Read (2013) used analysis correction as a DA method and Ravela et al. (2010) presented a setup to assimilate data using EnKF and showed it could be used to predict behaviour. The results presented in this thesis provides a variety of results for EnKF showing situation where it is able to help accurately predict the results. The results presented in the twin experiments show a promising start especially for low rotation rate of 1 rad/s. When increasing the rotation rate problems start to arise, which were also observed by Young and Read (2013) using analysis corrections. Both system starts having problems when tracking the more complex structure even when using higher number of observed points and shorter DA times. More tuning is needed using EnKF in the study presented here for the higher rotation rates to get accurate results along with other changes suggested in this section.

9.1.4 DA using Experimental Observation

Using the understanding gained from the previous experiments DA can finally be applied using the experimental data as the observations. Due to the quality of data and other restriction DA was done for system rotating at 1 rad/s and 2.5 rad/s with 500 random points from the experiments being used as observations. An ensemble of 10 was used with an additive perturbation of 2pert for all studies with DA being done every 1 minute.

At 1rad/s the ensemble can replicate the fluid structures observed in the PIV data with an m=3 being observed. But the ensemble is never able to tack the truth with the latter travelling around the annulus faster. The slow drift velocity might be due to the low θ resolution of just 64 points. The test was repeated but with an increased θ resolution of 128 points. Here the ensemble starts with an m=4 wave and then collapses to an m=2 wave. This collapse of the wave might be due to the DA corrections. With this difference in wavenumber between the ensemble and truth, the ensemble will never be able to track and predict the truth.

The rotation rate was increased to 2.5rad/s for the next study. Here the PIV data shows an m=4 wave but in the first test with a θ resolution of 64 the ensemble shows an m=3 wave. The ensemble is never able to move up to an m=4 wave. Even when the θ resolution is increased to 128 points, the ensemble average is still a m=3 wave. In both cases, the ensemble is not able to track the truth understandably.

Although no accurate results were achieved, a lot was learned about what kind of observations are needed to get more accuracy in prediction. Young and Read (2013) used around 2000 vector points in their studies and this study was only using around 1000 vector points. It is clear in the results presented in this thesis that the PIV data needs to be reprocessed to obtained more accurate values. In both cases the U and V velocity values were lower compared to values predicted by the ensemble. For the thesis results the resolution of the ensemble needs to be increased with at lest the 128 points in the θ for better resolution of rotation at higher rotation rates. Finally if the experiment is repeated, only one level should be observed to obtain continuous data and to not miss any important behaviour that might have been missed when switching between levels.

9.2 Future Work

This thesis has presented a way to use EnKF to predict the behaviour in a thermal rotating annulus. And although a lot of different studies have been presented here, improvements can be made to all aspects of the studies leading to a better prediction for the thermal rotating annulus.

Data Assimilation methods

The version of EnKF used for the studies present here was a very stochastic approach as aspects such the additive inflation and observation perturbations were random. This approach was deliberately chosen for two reasons, first to see how a simple DA method will behave when used for an increasingly complex system. Second, to see how difficult it is to implement EnKF for complex 3D system and gain experience in operating it.

As mentioned in the DA chapter there are other versions of EnKF such as Ensemble Square Root Filter (EnSRF) and Ensemble Transform Kalman Filter (ETKF) which are more deterministic in their approach. These approaches could help if there is any spurious correlation in the system. This could lead to a better prediction. The system can also be updated for asynchronous DA which will be useful for applications in real life problems such as predicting instabilities in tokamaks.

Experiments

This was the first time this set of equipment was used to conduct long duration experiments for a thermal rotating annulus at different rotation rates at the AOPP. Previous results (Young and Read, 2013) used an older experiment which was done using older equipment. As such a lot of improvement can be made on how the data is collected and analysed.

1. Data at level 4 seems to be the best when it comes to being useful for DA. A few experiments can be done while only level 4 is observed throughout the run to have a larger set of data that can be analysed. This could be done especially for the higher rotation rates to get more of a chance to obtain usable results.

2. Doing analysis using a continuous data set should also lead to better calculation of U and V velocity, here switching between levels hampered getting continuous data. This will also lead to a better understanding of the behaviour of the working fluid, leading to better Hovmöller plots, better calculations of the drift rates and better understanding of phenomenon such as vacillation.

3. More tuning is needed on DynamicStudio's PIV software, as it seems like a lot of data is lost when the images are processed to generate the values of U and V velocity. The images taken during the experiments was 1024x768 pixels while the processed data file contains just 83x62 data points. More data points might lead to clearer plots and more data points that can be used for DA. Maybe some sort of data interpolation can be used to create data that can be used for DA.

4. Only data from a 1 rad/s level 4 was used to find settings that generate reasonable results on DynamicStudio. Hence more tuning is also required to get a more optimal settings for both different levels and different rotation rates. With faster rotation rates the number of pixels needed to obtain accurate results will increase. This might decrease the resolution of the image further, leading to less usable data. One way to overcome this problem might be to increase the frequency of image acquisition for faster rotation rates. This should help PIV software to track particles more accurately leading to better prediction.

5. Another option is to use a different PIV software as DynamicStudio is not generally used to analyse data from a thermal rotating annulus. This meant that there was less of an idea on what setting might be optimal in practice to get good results.

6. With better experimental data, DA studies can be repeated to get better results.

Numerical model

A lot of studies were presented in this thesis using the twin experiment and, as reported, most of the time EnKF did not do a good job of tracking the observations. Hence a lot of improvements can now be made to see what situations are needed where EnKF can track the observations.

1. At higher rotation rates the ensemble has a lot of problems with accurately replicating the correct wavenumber and drift rates as observed in the high resolution model and experiments. Hence the θ resolution needs to be increased from 64 to 128 for a better transfer of information around the rotating annulus. This should lead to a better rotation of the working fluid in the ensemble.

2. Larger ensemble run needs to be done at higher rotation rates to make up for the more chaotic structures that are observed. This will be computationally more expensive but should lead to better results. Using the larger ensemble, studies done at lower rotation rates, such as looking at where around the annulus are the best places to observe can be repeated. These will help when creating/designing real life experiments when using a thermal rotating annulus. The ensemble size was deliberately kept low for all the studies to look at the extreme conditions where EnKF could still work.

3. All the studies done here mostly looked at the qualitative side of DA i.e. looking at how well EnKF did at helping the ensemble track and predict the observation. But this can be expanded to look at the quantitative side as well. This will mean start looking at things such as how the covariance changes with time and if there are any spurious correlation in the system. This will help in determining if localisation is needed in the system. With localisation, the ensemble size needed for accurate prediction could be decreased which will save computational costs.

CCFE

The final aim of this project was to apply what was learned about EnKF to helping better predict better behaviour in a tokamak in collaboration with CCFE. Some preliminary tests were done at CCFE such as looking at what data can be extracted from tokamaks and what the look like. A few simple models were also look at to see if they produce similar results as extracted. But no final decisions were made on the data or the model and due to the time spent doing steadies on the thermal rotating annulus, it was not possible to fully apply what was learned from a thermal rotating annulus on a tokamak. Given how easy it is to apply EnKF to another setting this should not take too much time to setup knowing what model is being used and what data is being extracted.

Bibliography

Anderson, J. (2009), 'Spatially and temporally varying adaptive covariance inflation for ensemble filters', *Tellus A* **61**, 72–83.

Annan, J. D. and Hargreaves, J. C. (2004), 'Efficient parameter estimation for a highly chaotic system', *Tellus A: Dynamic Meteorology and Oceanography* **56**(5), 520–526.

Arakawa, A. and Lamb, V. R. (1977), 'Computational design of the basic dynamical processes of the ucla general circulation model', *Methods in Computational Physics: Advances in Research and Applications* **17**, 173–265.

Bannister, R. (2017), 'A review of operational methods of variational and ensemble-variational data assimilation', *Quarterly Journal of the Royal Meteorological Society* **143**, 607–633.

Barnes, S. L. (1964), 'A technique for maximizing details in numerical weather map analysis', *Journal of Applied Meteorology* **3**, 396–409.

Bergthórsson, P. and Döös, B. R. (1955), 'Numerical weather map analysis', *Tellus* **7**, 329–340.

Bowler, N. E., Arribas, A., Mylne, K. R., Robertson, K. B. and Beare, S. E. (2008), 'The mogreps short-range ensemble prediction system', *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography* **134**(632), 703–722.

Burgers, G., Balmaseda, M. A., Vossepoel, F. C., van Oldenborgh, G. J. and Van Leeuwen, P. J. (2002), 'Balanced ocean-data assimilation near the equator', *Journal of physical oceanography* **32**(9), 2509–2519.

Burgers, G., Jan van Leeuwen, P. and Evensen, G. (1998), 'Analysis scheme in the ensemble kalman filter', *Monthly Weather Review* **126**, 1719–1724.

Chandrasekhar, S. (1953), 'The instability of a layer of fluid heated below and subject to coriolis forces', *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **217**(1130), 306–327.

Dalziel, S. (1994), ‘Image processing for fluid dynamics’, *Cambridge Enviromental Reserch Consultants Ltd* .

Eady, E. (1949), ‘Long waves and cyclone waves’, *Tellus* **I**, 33–52.

Einieke, G. A., ed. (2012), *Smoothing, Filtering and Prediction - Estimating The Past, Present and Future*, InTech.

Evensen, G. (1994), ‘Sequential data assimilation with a nonlinear quasigeostrophic model using monte carlo methods to forecast error statistics.’, *J. Geophys. Res. – Oceans* **99(C5)**, 143–162.

Evensen, G. (1997), ‘Advanced data assimilation for strongly nonlinear dynamics’, *Mon. Weather Rev.* **125**, 1342–1354.

Evensen, G. (2003), ‘The ensemble kalman filter: Theoretical formulation and practical implementation’, *Ocean Dynamics* **53**, 343–367.

Evensen, G. (2009), ‘The ensemble kalman filter for combined state and parameter estimation’, *IEEE Control Systems Magazine* **29**(3), 83–104.

Farnell, L. and Plumb, R. A. (1975), ‘Numerical integration of flow in a rotating annulus i: axisymmetric model’, *Occasional Note Met O 21 75/3, Geophysical Fluid Dynamics Laboratory, UK Meteorological Office. Unpublished technical report, 32 pages* .

Farnell, L. and Plumb, R. A. (1976), ‘Numerical integration of flow in a rotating annulus ii: three dimensional model’, *Occasional Note Met O 21 76/1, Geophysical Fluid Dynamics Laboratory, UK Meteorological Office. Unpublished technical report, 24 pages* .

Fein, J. S. (1973), ‘An experimental study of the effects of the upper boundary condition on the thermal convection in a rotating differentially heated cylindrical annulus of water’, *Geophysical Astrophysical Fluid Dynamics* **5**, 213–248.

Früh, W.-G. and Read, P. L. (1997), ‘Wave interactions and the transition to chaos of baroclinic waves in a thermally driven rotating annulus’, *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* **355**(1722), 101–153.

Fultz, D., Long, R. R., Owens, G. V., Bohan, W., Kaynor, R. and Weil, J. (1959), ‘Studies of thermal convection in a rotating cylinder with implications for large-scale atmospheric motions’, *Meteorological Monographs, Boston, American Meteorological Society* **4, No. 21**, 1–104.

Gaspari, G. and Cohn, S. (1999), ‘Construction of correlation functions in two and three dimensions’, *Quarterly Journal of the Royal Meteorological Society* **125**, 723–757.

Gilchrist, B. and Cressman, G. P. (1954), ‘An experiment in objective analysis’, *Tellus* **6**, 309–318.

Gill, A. E. (1982), *Atmosphere-Ocean dynamics (International Geophysics Series)*, academic press.

Harlander, U., von Larcher, T., Wang, Y. and Egbers, C. (2011), ‘Piv-and ldv-measurements of baroclinic wave interactions in a thermally driven rotating annulus’, *Experiments in fluids* **51**(1), 37–49.

Hide, R. (1958), ‘An experimental study of thermal convection in rotating fluid’, *Philosophical Transactions of the Royal Society* **A250**, 441–478.

Hide, R. (2010), ‘A path of discovery in geophysical fluid dynamics’, *Astronomy & Geophysics* **51**, 4.16–4.23.

Hide, R. and Mason, P. J. (1975), ‘Sloping convection in a rotating fluid’, *Advances in Physics* **24**(1), 47–100.

Hignett, P. (1985), ‘Characteristics of amplitude vacillation in a differentially heated rotating fluid annulus’, *Geophysical and Astrophysical Fluid Dynamics* **19**, 293–299.

Hignett, P., White, A. A., Carter, R. D., Jackson, W. D. N. and Small, R. M. (1985), ‘A comparison of laboratory measurements and numerical simulations of baroclinic wave flows in a rotating cylindrical annulus’, *Quarterly Journal of the Royal Meteorological Society* **111**(467), 131–154.

Houtekamer, P. and Mitchell, H. (1998), ‘Data assimilation using an ensemble kalman filter technique’, *Monthly Weather Review* **126**, 796–811.

Houtekamer, P., Mitchell, H. and Deng, X. (1999), ‘Model error representation in an operational ensemble kalman filter’, *Monthly Weather Review* **137**, 2126–2143.

Houtekamer, P. and Zhang, F. (2016), ‘Review of the ensemble kalman filter for atmospheric data assimilation’, *Monthly Weather Review* **144**, 4489–4532.

James, I. N., Jonas, P. R. and Farnell, L. (1981), ‘A combined laboratory and numerical study of fully developed steady baroclinic waves in a cylindrical annulus’, *Quarterly Journal of the Royal Meteorological Society* **107**(451), 51–78.

Judd, K., Reynolds, C. A., Rosmond, T. E. and Smith, L. A. (2008), ‘The geometry of model error.’, *J. Atmos. Sci.* **65**, 1749–1772.

Kalman, R. (1960), ‘A new approach to linear filtering and prediction problems’, *Journal of Fluids Engineering* **82**, 35–45.

Kalman, R. and Bucy, R. (1961), ‘New results in linear filtering and prediction theory’, *Journal of Fluids Engineering* **83**, 95–108.

Kalnay, E. (2003), ‘Atmospheric modeling, data assimilation, and predictability’, *Cambridge University Press* .

Kalnay, E., Li, H., Miyoshi, T., Yang, S.-C. and Ballabrera-Poy, J. (2007), ‘4-d-var or ensemble kalman filter?’, *Tellus A* pp. 758–773.

Larcher, T. and Egbers, C. (2005), ‘Experiments on transitions of baroclinic waves in a differentially heated rotating annulus’, *Nonlinear Processes in Geophysics* **12**(6), 1033–1041.

Lorenc, A. C., Ballard, S. P., Bell, R. S., Ingleby, N. B., Andrews, P. L. F., Barker, D. M., Bray, J. R., Clayton, A. M., Dalby, T., Li, D., Payne, T. J. and Saunders, F. W. (2000), ‘The met. office global three-dimensional variational data assimilation scheme’, *Quarterly Journal of the Royal Meteorological Society* **126**, 2991–3012.

Lorenc, A. C., Bell, R. S. and Macpherson, B. (1991), ‘The meteorological office analysis correction data assimilation scheme’, *Quarterly Journal of the Royal Meteorological Society* **117**, 59–89.

Lorenz, E. N. (1963), ‘Deterministic nonperiodic flow’, *Journal of Atmospheric Science* **20**, 130–141.

Lorenz, E. N. (1996), Predictability: A problem partly solved, *in* ‘Proc. Seminar on predictability’, Vol. 1.

Marshall, J., Adcroft, A., Hill, C., Perelman, L. and Heisey, C. (1997), ‘A finite-volume, incompressible navier stokes model for studies of the ocean on parallel computers’, *Journal of Geophysical Research: Oceans* **102**(C3), 5753–5766.

Marshall, J., Hill, C., Perelman, L. and Adcroft, A. (1997), ‘Hydrostatic, quasi-hydrostatic, and nonhydrostatic ocean modeling’, *Journal of Geophysical Research: Oceans* **102**(C3), 5733–5752.

Miller, R. N. (1994), ‘Perspectives on advanced data assimilation in strongly nonlinear systems. data assimilation: Tools for modelling the ocean in a global change perspective’, *P. P. Brasseur and J. C. J. Nihoul- Eds. NATO ASI- Springer-Verlag* **I 19**, 195–216.

Miller, T. L. and Butler, K. A. (1991), ‘Hysteresis and the transition between axisymmetric flow and wave flow in the baroclinic annulus.’, *Journal of Atmospheric Science* **48**, 811–824.

Mitchell, H. L. and Houtekamer, P. L. (2009), ‘Ensemble kalman filter configurations and their performance with the logistic map’, *Monthly Weather Review* **137**(12), 4325–4343.

MORALS (2020), ‘https://www2.physics.ox.ac.uk/research/geophysical-fluid-dynamics/morals’, *Oxford University AOPP website* .

Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., Kalnay, E., Patil, D. J. and Yorke, J. A. (2004), 'A local ensemble kalman filter for atmospheric data assimilation', *Tellus A: Dynamic Meteorology and Oceanography* **56**(5), 415–428.

Panofsky, R. A. (1949), 'Objective weather-map analysis', *Journal of Meteorology* **6**, 386–392.

Parrish, D. F. and Derber, J. C. (1992), 'The national meteorological center's sectral statistical-interpolation analysis system', *Monthly Weather Review* **120**, 1747–1763.

Pfeffer, R. L. and Fowlis, W. W. (1968), 'Wave dispersion in a rotating, differentially heated cylindrical annulus of fluid', *Journal of the Atmospheric Sciences* **25**(3), 361–371.

Ravela, S., Marshall, J., Hill, C., Wong, A. and Stransky, S. (2010), 'A realtime observatory for laboratory simulation of planetary flows', *Exp. Fluids* **48**, 915–925.

Rawlins, F., Ballard, S. P., Bovis, K. J., Clayton, A. M., Li, D., Inverarity, G. W., Lorenc, A. C. and Payne, T. J. (2007), 'The met office global four-dimensional variational data assimilation scheme', *Quarterly Journal of the Royal Meteorological Society* **133**, 347–362.

Read, P. L., Lewis, S. R. and Hide, R. (1997), 'Laboratory and numerical studies of baroclinic waves in an internally heated rotating fluid annulus: a case of wave/vortex duality?', *Journal of Fluid Mechanics* **337**, 155–191.

Read, P. L., Pérez, E. P., Moroz, I. M., Young, R. M. B., von Larcher, T. and Williams, P. D. (2015), 'General circulation of planetary atmospheres: insights from rotating annulus and related experiments', *Modeling Atmospheric and Oceanic Flows* pp. 9–44.

Reichle, R. H., Walker, J. P., Koster, R. D. and Houser, P. R. (2002), 'Extended versus ensemble kalman filtering for land data assimilation', *Journal of hydrometeorology* **3**(6), 728–740.

Sakov, P., Evensen, G. and Bertino, L. (2010), 'Asynchronous data assimilation with the enkf', *Tellus, Series A: Dynamic Meteorology and Oceanography* **62**(1), 24–29.

Sakov, P. and Oke, P. R. (2008), 'A deterministic formulation of the ensemble kalman filter: an alternative to ensemble square root filters', *Tellus A: Dynamic Meteorology and Oceanography* **60**(2), 361–371.

Sasaki, Y. (1970), 'Some basic formalisms in numerical variational analysis', *Mon. Weather Rev.* **98**, 875–883.

Sitte, B. and Egbers, C. (2000), 'Higher order dynamics of baroclinic waves. – in: Physics of rotating fluids', *Springer* pp. 355–375.

Smith, G. L., Schmidt, S. F. and McGee, L. A. (1962), 'Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle', *National Aeronautics and Space Administration* .

Stemler, T. and Judd, K. (2009), 'A guide to using shadowing filters for forecasting and state estimation', *Phys. D* **238**, 1260–1273.

Taylor, G. I. (1921), 'Experiments with rotating fluids', *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* **100**(703), 114–121.

Taylor, G. I. (1923), 'Viii. stability of a viscous liquid contained between two rotating cylinders', *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* **223**(605-615), 289–343.

van Leeuwen, P. J. (2010), 'Nonlinear data assimilation in geosciences: an extremely efficient particle filter', *Quarterly Journal of the Royal Meteorological Society* **136**, 1991–1999.

Vettin, F. (1857), *Ann. Phys. Chem. Leipzig* **102**, 55–94.

Vincze, M., Borchert, S., Achatz, U., von Larcher, T., Baumann, M., Hertel, C., Remmler, S., Beck, T., Alexandrov, K. D., Egbers, C., Froehlich, J., Heuveline, V., Hickel, S. and Harlander, U. (2014), 'Benchmarking in a rotating annulus: a comparative experimental and numerical study of baroclinic wave dynamics', *Meteorologische Zeitschrift* **23**, 611–635.

Vincze, M., Harlander, U., von Larcher, T. and Egbers, C. (2014), 'An experimental study of regime transitions in a differentially heated baroclinic annulus with flat and sloping bottom topographies', *Nonlinear Processes in Geophysics* **21**, 237–250.

von Larcher, T. and Egbers, C. (2005), 'Experiments on transitions of baroclinic waves in a differentially heated rotating annulus.', *Nonlinear Processes in Geophysics* **12**, 1033–1041.

Wehner, W., Xu, C., Schuster, E., Moreau, D., Mazon, D., Walker, M. L., Humphreys, D. A. and In, Y. (2011), Data-driven modeling and feedback tracking control of the toroidal rotation profile for advanced tokamak scenarios in diii-d, *in* '2011 IEEE International Conference on Control Applications (CCA)', IEEE, pp. 858–863.

Weng, H., Barcilon, A. and Magnan, J. (1986), 'Transitions between baroclinic flow regimes', *Journal of the atmospheric sciences* **43**(16), 1760–1777.

Weng, H.-Y. and Barcilon, A. (1987), 'Wave structure and evolution in baroclinic flow regimes', *Quarterly Journal of the Royal Meteorological Society* **113**(478), 1271–1294.

Whitaker, J., Hamill, T., Wei, X., Song, Y. and Toth, Z. (2008), ‘Ensemble data assimilation with the ncep global forecast system’, *Monthly Weather Review* **136**, 463–482.

Young, R. M. B. and Read, P. L. (2013), ‘Data assimilation in the laboratory using a rotating annulus experiment’, *Quarterly Journal of the Royal Meteorological Society* **139**, 1488–1504.

Young, R. M. B., Read, P. L., Früh, W.-G., Smith, D. and Risch, S. (2015), ‘The thermally-driven rotating annulus: horizontal velocities in regular and weakly chaotic flow regimes [data-set]’, *Oxford: Oxford University Research Archive* .

Zhang, F., Zhang, M. and Hansen, J. A. (2009), ‘Coupling ensemble kalman filter with four-dimensional variational data assimilation’, *Advances in Atmospheric Sciences* **26**(1), 1–8.