DISSERTATION

A NOVEL SMOOTHER-BASED DATA ASSIMILATION METHOD FOR COMPLEX CFD

Submitted by

Christopher L. Hurst

Department of Mechanical Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2024

Doctoral Committee:

Advisor: Xinfeng Gao
Co-Advisor: Stephen Guzik

Wade Troxell
Peter Jan van Leeuwen

ABSTRACT

A NOVEL SMOOTHER-BASED DATA ASSIMILATION METHOD FOR COMPLEX CFD

Accurate computational fluid dynamics (CFD) modeling of turbulent flows is necessary for improving fluid-driven engineering designs. Traditional CFD often falls short of providing truly accurate solutions due to inherent uncertainties stemming from modeling assumptions and the chaotic nature of fluid flow. To overcome these limitations, we propose the integration of data assimilation (DA) techniques into CFD simulations. DA, which incorporates observational data into numerical models, offers a promising avenue to enhance predictability by reducing uncertainties associated with initial conditions and model parameters. This research aims to advance our understanding and application of DA for CFD modeling of highly chaotic dynamical systems. This dissertation makes several novel contributions in DA and CFD: i) A novel DA algorithm, the maximum likelihood ensemble smoother (MLES), has been developed and implemented to provide better model parameter estimation and assimilate time-integrated observations while addressing nonlinearity, ii) Multigrid-in-time techniques are applied to enhance the computational efficiency of the MLES by improving the optimization processes, and iii) The MLES+CFD framework has been validated by classical test problems such as the Lorenz 96 model and the Kuramoto-Sivashinsky equation. The effectiveness of the MLES has been demonstrated through a few test problems featuring chaos, discontinuity, or high dimensionality.

ACKNOWLEDGEMENTS

I am deeply grateful to my advisor, Dr. Gao, for her guidance and encouragement throughout this journey. Your passion for research has continuously inspired me, and I have evolved both personally and professionally beyond my expectations under your mentorship. Additionally, I extend my appreciation to Dr. Guzik for his valuable counsel and direction, which have been indispensable throughout this process.

I would like to thank my dissertation committee members, past and present: Dr. Zupanski, Dr. Windhom, Dr. van Leeuwen, and Dr. Troxell. Their feedback and suggestions over the course of my research has been much appreciated.

I also want to thank all my colleagues in the CFD and Propulsion Lab: Andy, Erik, Eli, Sean, Yijun, Jordan, Nate, Josh, Landon, Nelson, Scott, and Noah. Your continued friendship and discussions were an essential part of my growth as a researcher.

I owe a debt of gratitude to my family for their support and love throughout my time as a student, from kindergarten until now. My father, Maurice, and my mother, Colleen, have shaped and molded me to the academic I am today. You can be assured that this thesis dissertation is as much mine as it is yours. My twin sister, Allison, for the companionship and our annual trips together meant so much to me these last few years. My younger sister, Olivia, for being the first the lay claim on a doctorate in our family, which was a very good motivator for me to do the same.

I would like to thank my girlfriend, Tori. I am endlessly thankful for her presence in my life, making every moment richer and more meaningful.

My friends Franco, Marco, Julia, Marshall, Ben, Ethan, and anyone else who I may have not mentioned, please know that your friendship is deeply cherished and valued beyond measure. Thank you for being a part of my journey.

I am grateful to Dr. Gao, Dr. Guzik, and Colorado State University for the financial support for this research.

TABLE OF CONTENTS

ABSTRACT		ii
ACKNOWLEDGEMENTS		iii
LIST OF TABLES		vi
LIST OF FIGURES		vii
Chapter 1	Introduction	1
1.1	Motivations and Objectives	1
1.2	Objectives	4
1.3	Dissertation Organization	5
Chapter 2	Data Assimilation	6
2.1	Bayes' Theorem	6
2.2	Ensemble Methods	7
2.3	Variational Methods	7
2.4	Hybrid Methods	8
2.5	Maximum Likelihood Ensemble Smoother	8
2.5.1	Cost Function	10
2.5.2	Solution Process	13
2.5.3	Minimization	15
2.6	Multigrid in Time Applied to MLES	18
2.6.1	Motivation	18
2.6.2	Procedure	19
Chapter 3	Numerical Framework	23
3.1	CFD Forward Model (Chord)	23
3.1.1	Governing Equations	23
3.1.2	Grid Mapping	24
3.1.3	Spatial Discretization	26
3.1.4	Time Marching Method	27
3.2	DA+CFD Framework	28
Chapter 4	Verification and Validation	30
4.1	Lorenz-96 Model	30
4.1.1	Twin Numerical Experiment Setup	31
4.1.2	Data Assimilation Configuration	32
4.1.3	DA Performance Measures	33
4.2	Kuramoto-Sivashinsky Equation	37
4.2.1	Number of Time Points With Observations	38
4.2.2	Ensemble Size	39
4.2.3	Number of Observations	40
4.2.4	State Quantities and RMSE	40

4.2.5 Model Parameter Estimation . . . 44
4.2.6 Averaged Observations . . . 45
4.3 Verification and Validation: MLES+MGinT . . . 47
4.3.1 Performance metrics . . . 48
4.3.2 Parametric studies . . . 49

Chapter 5 Experiment Results and Discussion . . . 62
5.1 1D Shock Tube . . . 62
5.1.1 Problem Setup . . . 62
5.1.2 DA Configuration . . . 63
5.1.3 Performance Metrics . . . 64
5.1.4 Ensemble Size Sensitivity Test . . . 67
5.2 Conclusions . . . 84

Chapter 6 Conclusions and Future Work . . . 85
6.1 Conclusions . . . 85
6.2 Original Contributions . . . 86
6.3 Future Work . . . 87

Bibliography . . . 88

Appendix A Gradient and Hessian Derivations . . . 94
A.1 MLEF . . . 94
A.2 MLES . . . 100
A.3 Additional Minimization details . . . 102

LIST OF TABLES

4.1 the Lorenz 96 model experiment's data assimilation parameters. 32
4.2 The estimated model parameters for the MLES and MLEF by taking the average of the last analysis model parameters over 100 different cases. The mean of the initialization of the model parameters is set to be 50% less than that of the truth. 44
4.3 The estimated model parameters for the MLES and MLEF using the averaged observations method. 46
4.4 The forward model and DA configuration for the lag vs. shift parametric study. 52
4.5 The forward model and DA configuration for the lag vs. γ parametric study. All parameters not shown can be found in Table 4.4. 55
4.6 The forward model and DA configuration for the lag vs. Δt parametric study. All parameters not shown can be found in Table 4.4. 58

5.1 Shock tube parameters 63
5.2 DA parameters 69

LIST OF FIGURES

1.1 Visualization of the Lyapunov exponent, demonstrating how a small initial perturbation, $\delta\mathbf{U}_0$, applied to the initial model state diverges over time in a chaotic system. The Lyapunov exponent quantifies the exponential rate of separation between trajectories, providing insight into the system's sensitivity to initial conditions. 3

2.1 Graphical depiction of the MLES in state space. 10
2.2 One cycle of a smoother with a shift $L = 3$ and $S = 2$. The location of the first analysis is the red circle. The observations are shown with green crosses, with red arrows that show that they are needed in the current analysis. The truth is the black dotted line. The y-axis is an arbitrarily chosen state variable of singular spatial element. The particular observations used in the analysis are shown by showing red arrows pointing back to the current analysis time. 11
2.3 After the analysis is found, the new analysis is propagated 2 observations forward, corresponding to the value of $S = 2$. Then, the next observations, which are shown, are used in the analysis for the new, current DA window. 12
2.4 An example illustration of MGinT cycles in time. Levels 0, 1, 2, and 3 indicate the meshes from the finest to the coarsest in time, and, by applying the MLES, solutions of $\hat{\mathbf{Q}}_0$ at levels l_1, l_2, and l_3 can be found. 20
2.5 Running the coarse-grid forward model to the end of the assimilation window. State variables are obtained at times t_0, t_2, and t_5 while not acquiring information at times t_1 and t_3. 21
2.6 Running the fine-grid forward model to the end of the assimilation window. State variables are obtained at all times. 21

3.1 Graphical depiction of the DA+CFD system. 29

4.1 The $\boldsymbol{\chi}^2$ diagnostic test for the Lorenz 96 model experiment with a semi-log axis. There are two different x-axes that show the different DA cycles for the MLEF and MLES, respectively, but at the same solution times. 34
4.2 The normalized cost function values for the MLEF and MLES. There are two different x-axes that show the different DA cycles but at the same solution times. 35
4.3 The RMSE for the free run, MLEF, and MLES. Two different x-axes showing the same time point while marked by a different DA cycle number, e.g. 3 DA cycles in the MLES corresponding to 20 DA cycles in the MLEF. 36
4.4 The trajectory of Q_1 for the truth, MLES, MLEF, free run, and observations of the Lorenz 96 model over time. Both the MLEF and MLES are first updated at the 10^{th} time step and the MLEF is continually updated until the last observation available. The free run is simply a simulation by the forward model with the same steps as the MLES or MLEF but without DA. 36

4.5 The analysis RMSE of the MLES by varying S from 1 to 50. The red dotted points are the statistical average of 100 runs of the RMSE of the analysis of the MLES as a function of S. The blue circle is the MLEF at the analysis point and, as expected, matches the MLES when only one observation is taken in the window. 38
4.6 MLES performance as a function of N_e using a constant smoother window length of $S = 25$. 39
4.7 Graphical description of the error comparison with the MLES and MLEF. The MLES is updated at the vertical dashed red line and freely run to the vertical black dashed line. The MLEF is updated at each of the green hashes which represent available observations. The last update for the MLEF is located at the vertical blue dashed line and run to the same vertical black dashed line. At the vertical black dashed line, the error of the MLES and MLEF is calculated with respect to the truth and compared. This value is saved and the mean of these values is taken over 100 simulations. In the figure, the y-axis, $\mathbf{Q}$, represents just one state value in the entire spatial domain. 41
4.8 Performance comparison between the MLES and MLEF using a smaller amount of evenly spaced observations in the domain. The RMSE is taken at the end of a time window length equal to two smoother window lengths after the first initial MLES analysis update. 41
4.9 The model state and observation data at the first spatial location. 42
4.10 The model state and observation data at the 65^{th} spatial location. 42
4.11 The model state and observation data at the 221^{st} spatial location. 43
4.12 The RMSE values for the MLES, MLEF, and free run. 43
4.13 Model parameter a convergence. 45
4.14 Model parameter b convergence. 45
4.15 Model parameter c convergence. 45
4.16 To gather the averaged data, an average of the true values is taken within an interval and then the observation operator is applied and perturbed using random variance. The interval is shown as the vertical red dashed lines and the black horizontal dashed lines are the averaged true values. The black crosses represent the data obtained after applying the observation operator $\mathcal{H}(\cdot)$ and adding random variance, η. The y-axis, $\mathbf{Q}$, represents just one state value in the entire spatial domain. 46
4.17 Model parameter a convergence for the time-averaged synthesized observations. . . . 47
4.18 Model parameter b convergence for the time-averaged synthesized observations. . . . 47
4.19 Model parameter c convergence for the time-averaged synthesized observations. . . . 48
4.20 Plot of the first state variable in space, $\mathbf{Q}_1$ showcasing the effect that DA has on predicting the truth when compared to a free run. The red line is the MLES + MGinT and the blue line is the MLES. Both methods closely approximate the truth of the dotted black line. The gray dotted line is the free run with no DA applied, and quick divergence from the truth. 50
4.21 Distribution of the RMSE in the L-S parameter space. 53
4.22 Distribution of the ensemble spread in the L-S parameter space. 53
4.23 Distribution of the wall-clock times in the L-S parameter space. 54
4.24 Distribution of the minimization iterations required in the L-S parameter space. 54
4.25 Distribution of the RMSE in the L-γ parameter space. 56
4.26 Distribution of the ensemble spread in the L-γ parameter space. 56

4.27 Distribution of the wall-clock times in the L-γ parameter space. 57
4.28 Distribution of the minimization iterations required in the L-γ parameter space. 57
4.29 Distribution of the RMSE in the L-Δt parameter space. 59
4.30 Distribution of the ensemble spread in the L-Δt parameter space. 59
4.31 Distribution of the wall-clock times in the L-Δt parameter space. 60
4.32 Distribution of the minimization iterations required in the L-Δt parameter space. . . . 60

5.1 Initial state of pressure and density for the prior. For reference, the true state is given by the black line, and the prior is shown in blue. The thicker blue line shows the control and the thinner blue lines show the ensemble. In this figure, $N_e = 12$ and the initial starting time for the DA analysis is $t = 0.05$. 64
5.2 Pressure and gradient of pressure at $t = 0.05$ for the classic shock tube case. This figure shows the truth obtained by the analytical solution and the result from Chord. . . 65
5.3 Gradient of pressure at $t = 0.05$ of chord, showing the cubic spline to interpolate between cells. 66
5.4 Pressure values for the MLEF at $t = 0.05$, or the initial analysis time. The opaque red and blue lines show the ensemble members of the analysis and prior, respectively. The gray dashed line is the free run with no DA applied. 70
5.5 Pressure values for the MLES at $t = 0.05$, or the initial analysis time. The parameters for the MLES are $L = 4$ and $S = 4$. The opaque red and blue lines show the ensemble members of the analysis and prior, respectively. The gray dashed line is the free run with no DA applied. 71
5.6 Pressure values for the MLES and MLEF at $t = 0.05$, or the initial analysis time. Zoomed in on the shock location. 71
5.7 Pressure values for the MLEF of $N_e = 16$ at different times. 73
5.8 Pressure values for the MLES of $N_e = 16$ at different times. 74
5.9 Pressure values at the shock location for the MLEF at different times and different N_e values. 75
5.10 Pressure values at the shock location for the MLES at different times and different N_e values. 76
5.11 Pressure RMSE values for the MLEF at different times and different N_e values. 77
5.12 Pressure RMSE values for the MLES at different times and different N_e values. 77
5.13 Shock location for the MLEF at different times and different N_e values. 78
5.14 Shock location for the MLES at different times and different N_e values. 78
5.15 Shock location for the MLEF at different times and different N_e values. 79
5.16 Shock location for the MLES at different times and different N_e values. 79
5.17 Pressure at different times and for different configurations of the MLES. A portion of the domain is shown for identifying the quality of the shock capturing mechanics. . . . 82
5.18 Pressure RMSE at different times and for different configurations of the MLES. 83
5.19 Pressure at different times and for different configurations of the MLES. 83
5.20 Shock quality at different times and for different configurations of the MLES. 84

Chapter 1

Introduction

1.1 Motivations and Objectives

Using computational fluid dynamics (CFD) to accurately obtain a solution to complex problems rarely results in obtaining what is regarded as the 'true' solution. This is because the quality of these solutions is determined by the many assumptions associated with our numerical models, including discretization schemes, assumptions in the physics models, and so on. These assumptions eventually lead to deviations from reality. In engineering applications, addressing this discrepancy proves to be a time-consuming and resource-intensive aspect of making quality design choices. Naturally, research has been focused on reducing the dependency of these assumptions in order to obtain more accurate solutions. Great strides have been made and the continued research will ultimately prove to be worthwhile. However, progress has somewhat stagnated, reaching a juncture where using additional sources of uncertainties becomes a viable avenue to produce a more comprehensive understanding of the uncertainties involved in CFD applications. Currently, one of the biggest sources of uncertainty that is associated with complex CFD problems is chaos. Chaos is a term used to describe the phenomena commonly known as the 'butterfly-effect.' Lorenz [1], in his groundbreaking discovery, he found that if two numerical experiments were to start using the same exact configuration, these two experiment will completely diverge. The only difference was there was small round-off errors in the initial conditions. The question is then which solution provided a more accurate representation of reality? Answering this question is especially challenging without a trustworthy source to ascertain the truth. Indeed, Lorenz [1] showed the mathematical models and available computing power were for simpler than in today's typical architecture. Thus, with the widespread use of sophisticated algorithms and applications to the complex numerical experiments seen in engineering applications, this problem of chaos is unavoidable.

The focus on reducing the errors of the numerical models has drastically increased the quality of solutions in numerical CFD problems. Despite their effectiveness, these approaches are inadequate in capturing the uncertainty associated with the modeling assumptions (e.g., boundary conditions, operating conditions, geometrical configurations, and flow regimes, etc.). If, for example, all assumptions in the underlying physics are guaranteed to represent the truth of reality, the necessary modeling assumptions will not contaminate the solution at future times. This phenomena can be described by the Lyapunov exponent of the dynamical system. It is a quantity that measures the 'butterfly-effect' in which it mathematically describes the rate of separation of infinitesimally close state trajectories. When considering two trajectories in phase space with an initial perturbation, the rate of separation is given by

$$|\delta \mathbf{U}(t)| \approx e^{\lambda t} |\delta \mathbf{U}_0| \,, \tag{1.1}$$

where λ is the Lyapunov exponent, $\delta \mathbf{U}_0$ is the initial difference between the two runs, and $\delta \mathbf{U}(t)$ is the difference at some arbitrary future time. In essence, the Lyapunov exponent is a measure of chaos for a numerical model. Figure 1.1 illustrates the concept, showing how the system's sensitivity to initial conditions and the exponential rate at which trajectories separate are quantified by the Lyapunov exponent. Unsurpisingly, a turbulent regime exhibits chaotic dynamics, as demonstrated by evaluating the Lyapunov exponents of the Navier-Stokes equations in Yamada et al. [2]. Consequently, while conventional methods in numerical modeling are effective in providing robust engineering design assessments, they are bound by the mathematical constraints of chaos, ultimately rendering them inadequate in representing the truth. Recognizing these limitations, the exploration of strategies that use more information (e.g. physical data) provides a broader spectrum of sources of error. Integrating these highly accurate models into our proposed application is necessary in order to harness their precision.

Various methods exist to achieve this goal, but data assimilation (DA) [3–6] stands out. DA incorporates reliable observational data into numerical models to improve their accuracy and reliability, thus we choose DA as our approach to enhance the predictability of CFD simulations.

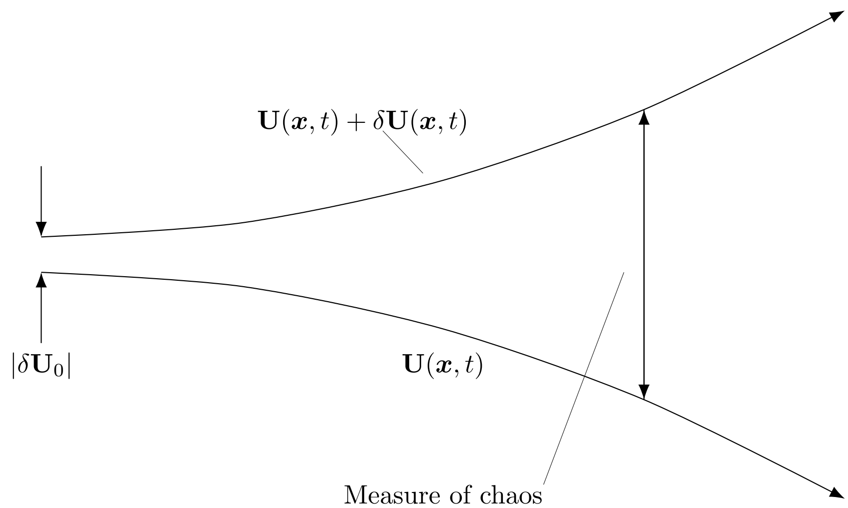

Figure 1.1: Visualization of the Lyapunov exponent, demonstrating how a small initial perturbation, $\delta\mathbf{U}_0$, applied to the initial model state diverges over time in a chaotic system. The Lyapunov exponent quantifies the exponential rate of separation between trajectories, providing insight into the system's sensitivity to initial conditions.

This is due to several reasons. Firstly, DA uses observations that are obtained from reality. This is advantageous because the availability of data from physical experiments continually grows, easing the challenge in producing experimental results. Secondly, DA must use a model for the desired problem, a benefit that can be maximized by coupling them with advanced CFD models of high precision. Lastly, DA has demonstrated success across multiple fields, serving as an excellent foundation for the research directions that we aim to pursue.

While DA has traditionally been applied to numerical weather prediction (NWP), its application has also extended to CFD [7–9]. The chaotic phenomena observed in NWP closely parallels those encountered in complex CFD and the results have shown notable success (e.g., [4, 10–14]). We use this research foundation to apply analogous methodologies from NWP to address these challenges in CFD. These studies have shown that incorporating DA into CFD simulations, they can account for uncertainties and errors in initial conditions and model parameters, reducing inaccuracies, and improving the predictability. These simulations, which often contain complex physical processes, involve highly nonlinear interactions. While the previous findings exhibit proficiency in handling these complex physics, applying DA to CFD remains a significant challenge to address these nonlinearities. The chosen algorithms must ensure access to high-quality and diverse observational data sources, developing robust DA algorithms, and efficiently integrating DA into existing CFD frameworks. In such scenarios, many DA methods face limitations, as they typically utilize only two modes of the error statistics, the mean and the variance, to represent the error characteristics of the system. This restriction leads to an underestimation of errors, hindering the accurate assimilation of observational data into the numerical models. Addressing these challenges and advancing the state-of-the-art in DA for CFD is crucial. The current dissertation focuses on developing more sophisticated assimilation algorithms for CFD modeling and simulations of engineering problems, improving the capability of assimilation of highly nonlinear physical processes, and optimizing the computational efficiency of these methods. Ultimately, the continued application of DA into CFD holds the promise of revolutionizing our ability to model and predict complex fluid dynamics with greater accuracy, benefiting a wide range of industries and scientific endeavors.

1.2 Objectives

Four research objectives have been identified to achieve the dissertation goal.

1. Build a comprehensive understanding of the limit of predictability for chaotic systems. The aim is to apply DA to CFD modeling of chaotic systems for greater accuracy, and thus understanding the coupling of DA+CFD is paramount.

2. Develop and implement a novel DA algorithm, the maximum likelihood ensemble smoother (MLES), to provide better model parameter estimation and assimilate time-integrated observations, in addition to treating nonlinearity.

3. Explore techniques to increase the computational efficiency of the MLES by leveraging multigrid-in-time for fast optimization process.

4. Apply the MLES to a problem featuring chaos or discontinuities to demonstrate its effectiveness in improving the limit of predictability.

1.3 Dissertation Organization

This dissertation is structured as follows. Chapter 2 describes the mathematical formulations of the novel DA algorithm. Chapter 3 reviews the numerical framework of the forward solvers used in CFD and the combined DA and CFD system. In Chapter 4, the system undergoes verification and validation before being applied to complex fluid dynamics problems. Chapter 5 is dedicated to discussing the configurations of the problems, the procedures, and the analysis and assessments of the results. Finally, Chapter 6 draws the conclusions, presents original contributions, and outlines potential avenues for future work.

Chapter 2

Data Assimilation

This chapter first focuses on the mathematical foundation of all DA methods. Then, it describes the development rationale and mathematical details of a novel DA method that is specifically designed to be applied to the highly nonlinear problems seen in typical CFD.

2.1 Bayes' Theorem

DA uses information from the three sources: the prior, the likelihood, and the observation data. The prior is the initial belief or probably distribution of the predicted state, typically sourced from the numerical model. The likelihood measures how well the predicted model state matches with the given observation data. The observation data is a marginal likelihood of observed data regardless of the prior information. Because error is associated with each of these conditional probabilities, we represent the available information as probability density functions (pdfs). Bayes' theorem [15] uses these pdfs and outputs the updated belief of the parameters of interest, known as the posterior pdf, $p(\hat{\mathbf{Q}}|\mathbf{O})$, as

$$p(\hat{\mathbf{Q}}|\mathbf{O}) = \frac{p(\mathbf{O}|\hat{\mathbf{Q}})}{p(\mathbf{O})} p(\hat{\mathbf{Q}}) \, , \tag{2.1}$$

where $p(\hat{\mathbf{Q}})$ is the prior pdf, $p(\mathbf{O}|\hat{\mathbf{Q}})$ is the likelihood, and $p(\mathbf{O})$ is the marginal of the observation pdf. $\mathbf{O}$ denotes the observation data and $\hat{\mathbf{Q}}$ is the deterministic vector of model states. Eq. (2.1) is the foundation of the vast majority of DA methods. Solving for the posterior pdf means to solve for the estimate of the pdf of the state based on all available information which is the answer DA seeks to find. There are many distinct methods to solve Bayes' theorem. To develop a DA algorithm that can handle the complex problems seen in CFD, it is essential to describe the three main categories of DA methods.

2.2 Ensemble Methods

The problems applied to DA are usually high dimensional and so reducing the dimension of the workspace makes the analysis more tractable. This is because obtaining a precise estimation of the uncertainty associated with the forecast produced by the numerical model demands a full-rank matrix over the model state space (N_s), requiring a sample of N_s independent model state vectors. This is manifested in the form of the forecast error covariance matrix, $\mathbf{P}_f$, whose size becomes computationally impractical, especially in high-dimensional applications where N_s is around 10^9. To address this issue, ensemble-based DA methods reduce the rank of $\mathbf{P}_f$, and therefore the problem as a whole, by using an ensemble of forecasted states to represent the uncertainty in the system. This makes the ensemble approach feasible because it reduces the covariance matrix dimension from $N_s \times N_s$ to $N_s \times N_e$, where N_e (the ensemble size) typically ranges from 10 to 100. In addition to reducing the rank of the problem significantly, the background errors contain valuable flow-dependent information, providing a comprehensive representation of the errors in the system. This is because the ensemble is propagated to the analysis time which allows $\mathbf{P}_f$ to contain information from the nonlinear forward model. A negative side effect of the limited number of ensemble members is sampling errors. These errors arise because the small ensemble fails to comprehensively capture all rapidly evolving error modes within the system, resulting in noisy correlations and underestimated variances [7, 16, 17]. To address these issues, techniques such as localization [17–20] and ensemble inflation [21, 22] are commonly employed. Mathematically, these strategies are perceived as ad-hoc and resemble heuristic interventions [23], and their effectiveness may not meet rigorous standards in addressing the sampling error.

2.3 Variational Methods

Variational DA [24,25] derives a cost function that measures the difference between the model's version of observations to the actual observations, along with any additional constraints or regularization terms. This cost function is minimized using nonlinear minimization techniques. The result is the optimal estimate of the initial state and trajectory of the model. This is well-suited for dynamical systems that contain nonlinearities because the solution procedure is inherently nonlinear. This is crucial when dealing with the nonlinear dynamics and uncertainty inherent in transient systems like turbulence in CFD modeling because previous studies have shown that the assumption of linearity breaks down in realistic situations [4, 10–12, 14, 26]. However, there are two major drawbacks to using variational methods. The first drawback is the limitation in quantifying flow-dependent background and model error statistics. This is due to the static representation of the background errors which are not consistently evaluated. The second drawback is the linearized version of the observation and model operator is required. Outside of meteorology where such versions of the model do not exist, this version of the model operator is especially difficult to derive and maintain.

2.4 Hybrid Methods

The applicability of the previous methods tends to be limited by underlying assumptions and methodologies, and they limit their usefulness to simple problems [27, 28]. To overcome these limitations, we opt to use a hybrid approach, combining both ensemble and variational methods. Hybrid approaches aim to mitigate their respective drawbacks while leveraging the strengths from each method. They incorporate the dynamical estimation of prior errors from ensemble-based methods along with the full-rank representation of all errors from variational methods. Most of these hybrid methods minimizes a variational cost function in ensemble space, eliminating the need for linearization of the forward model and observation operator. Bannister [29] provides an overview of the new development of hybrid methods that represent the forefront of solving highly nonlinear problems.

2.5 Maximum Likelihood Ensemble Smoother

Hybrid methods address the reduced rank of the prior covariance estimation and nonlinear optimization, but another major consideration to address is the filtering approach. Sequential DA filtering algorithms predict the optimal state at the current time by incorporating observations available at that specific moment. This process involves adjusting the model's predictions in real-time based on the most recent observational data. The updated state of the system aligns closely with the actual conditions observed at the given time point. However, there are limitations of sequential filtering approaches. The reason for impediment is that the forward model is employed outside of the analysis procedure. This poses a significant challenge, especially in situations involving time-integrated observations and the estimation of model parameters, where filtering methods are not adequate [30]. Unlike updating based on a singular observation point, ensemble-based smoothers use the nonlinear forward model to propagate the ensemble forward to evaluate the estimate at multiple observations over a specific time, or DA window. This effectively addresses the model's contribution to nonlinearity within the analysis procedure. Recognizing the need for a more robust solution, we turn our attention to hybrid-based smoothers, where the prediction model is integrated into the analysis procedure. Further reading for the advancements in smoother DA methods can be found in many references (e.g., [31–35]).

In this dissertation, we introduce a novel, hybrid-based smoother that can answer the challenges of realistic high-dimensional applications. We begin with a hybrid-based filter that is already known for its effectiveness in complex CFD problems, the maximum mikelihood ensemble filter (MLEF). We then extend the MLEF into a smoother. The novel approach is called the maximum likelihood ensemble smoother (MLES). By emphasizing the strengths of the MLEF and its relevance to the challenges seen complex CFD, the extension to a smoother will provide effective solutions for realistic high-dimensional applications. Specific details on the MLEF can be found in Zupanski [36]. The subsequent sections of this chapter introduce the method and offers a detailed exploration of its unique characteristics, breaking down its aspects at the conceptual, mathematical, and numerical algorithm levels.

Before introducing the mathematical procedure of the MLES, a graphical depiction of its procedure is given in Figure 2.1. In the figure, the forecasted ensemble and control state both start at the beginning of the DA window (pink dot). The solution process of the MLES applies the forward model to the ensemble (gray lines) iteratively within the DA window, indicated by the available data (green hashes and crosses). The resulting optimal analysis state (red dot) at the start of the DA window serves as the basis for forward forecast, providing the most accurate representation of the true state within the entire DA window.

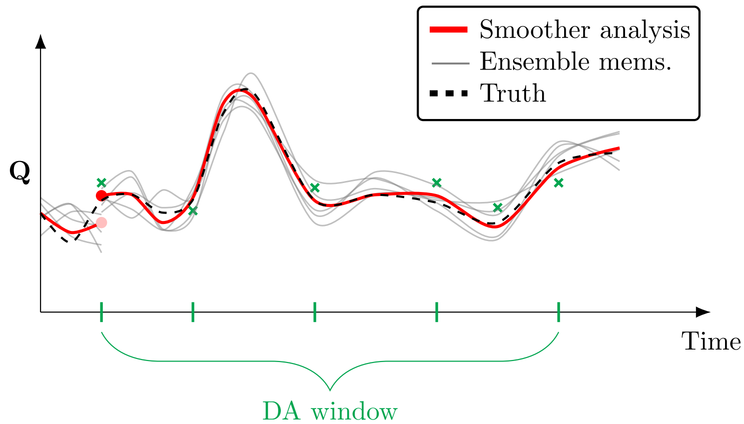

Figure 2.1: Graphical depiction of the MLES in state space.

2.5.1 Cost Function

The mathematical approach of the MLES is now described. The Bayesian framework provides a systematic approach for solving the constrained optimization problem, where the goal is to find the optimal state estimate. To achieve this, a cost function is derived by applying Gaussian assumptions on the terms in Bayes' theorem and then applying the maximum likelihood estimation:

$$\begin{aligned}\min_{\hat{\mathbf{Q}}_0} J(\hat{\mathbf{Q}}_0) &\stackrel{\text{def}}{=} \frac{1}{2}\left[\hat{\mathbf{Q}}_0 - \hat{\mathbf{Q}}^f\right]^{\mathrm{T}} \mathbf{P}_f^{-1}\left[\hat{\mathbf{Q}}_0 - \hat{\mathbf{Q}}^f\right] \\ &+ \frac{1}{2}\sum_{k=L-S+1}^{L}\left[\mathbf{O}_k - \mathcal{H}_k \circ \mathcal{M}_{0:k}(\hat{\mathbf{Q}}_0)\right]^{\mathrm{T}} \mathbf{R}_k^{-1}\left[\mathbf{O}_k - \mathcal{H}_k \circ \mathcal{M}_{0:k}(\hat{\mathbf{Q}}_0)\right] .\end{aligned} \tag{2.2}$$

where $\mathcal{H} \circ \mathcal{M}_{0:k}(\hat{\mathbf{Q}}_0)$ is a composition of the forward model and observation operators. The observation operator, $\mathcal{H}(\cdot)$, maps the solution state to the observation space, and the CFD model, $\mathcal{M}(\cdot)$, represents the forecast or forward model. This cost function, shown in Eq. (2.2), is minimized for $\hat{\mathbf{Q}}$, which is a control vector containing the solution variables $\mathbf{U}$ and model parameters $\boldsymbol{\alpha}$, such that $\hat{\mathbf{Q}} = [\mathbf{U}, \boldsymbol{\alpha}]^{\mathrm{T}}$. The forecast error covariance is represented by $\mathbf{P}_f$, and the observation error covariance is represented by $\mathbf{R}$. Both $\hat{\mathbf{Q}}^f$ and $\hat{\mathbf{Q}}$ are defined at the initial times of the DA window. The superscript T denotes the transpose, and the subscript f represents the forecast. The shift length, S, is the number of observations within the DA window. The lag length, L, is the temporal index of the last observation. After finding the optimal estimate, the DA window shifts forward by S observations to avoid overlap between DA windows. For example, with $L = 3$ and $S = 2$, the last observation considered is three observations ahead, and the next analysis occurs two steps in the future. This example is shown in Figure 2.2 and Figure 2.3, where observations needed for analysis are indicated by red arrows pointing back to the analysis time. The configuration with $L = 0$ and $S = 1$ means $\mathcal{M}_{0:0}$ is the identity and corresponds to the MLEF. The lag and shift values influence the analysis's nonlinearity because of the nonlinear effects introduced by the forward model. Therefore, investigating these values is worthwhile. For more details on lag and shift times in hybrid methods, refer to Asch et al. [37], Bocquet et al. [33], and Grudzien et al. [38].

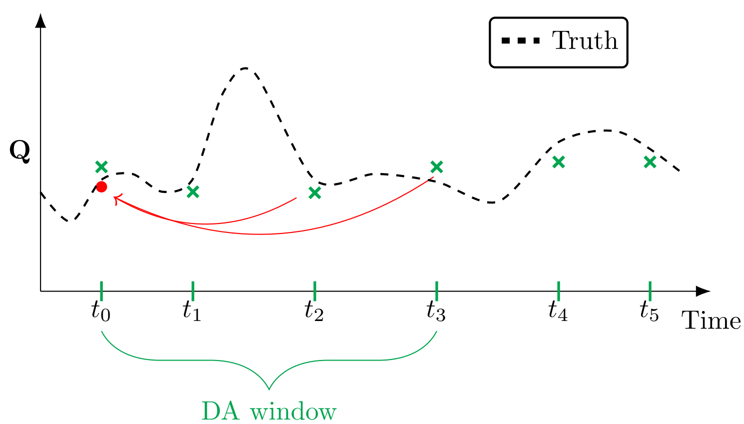

Figure 2.2: One cycle of a smoother with a shift $L = 3$ and $S = 2$. The location of the first analysis is the red circle. The observations are shown with green crosses, with red arrows that show that they are needed in the current analysis. The truth is the black dotted line. The y-axis is an arbitrarily chosen state variable of singular spatial element. The particular observations used in the analysis are shown by showing red arrows pointing back to the current analysis time.

The CFD model and observations/data errors are assumed to follow a Gaussian distribution with respect to the forecast error covariance, $\mathbf{P}_f$, and the observation error covariance, $\mathbf{R}$, respec-

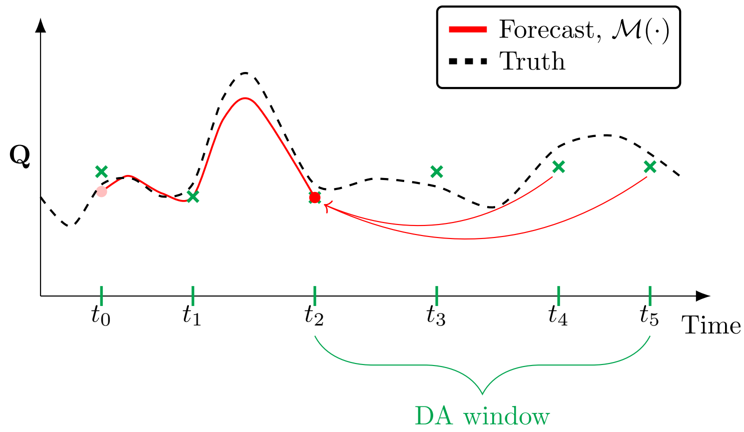

Figure 2.3: After the analysis is found, the new analysis is propagated 2 observations forward, corresponding to the value of $S = 2$. Then, the next observations, which are shown, are used in the analysis for the new, current DA window.

tively. The cost function is minimized subject to the constraints,

$$\begin{gathered}
\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{\nabla} \cdot \vec{\mathbf{F}} = \mathbf{S} \text{ in } \Omega \\
\mathbf{U} = g \text{ on } \partial\Omega \\
\hat{\mathbf{Q}} = [\mathbf{U}, \boldsymbol{\alpha}]^{\mathrm{T}} \\
\hat{\mathbf{Q}} \in \mathbb{R}^{N_s \times 1}, \quad \mathbf{U} \in \mathbb{R}^{N_{\text{var}} \times 1}, \quad \mathbf{O} \in \mathbb{R}^{N_o \times 1}, \\
\epsilon_{\mathbf{O}} \sim N(0, \mathbf{R}) \quad \text{and} \quad \epsilon_{\hat{\mathbf{Q}}} \sim N(0, \mathbf{P}_f),
\end{gathered} \tag{2.3}$$

where Ω is the physical domain of interest, N_s is the total number of the model state (variables and parameters) included in $\hat{\mathbf{Q}}$, N_{var} is the total number of solution variables in the forward model, and N_o is the total number of observations. $\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{\nabla} \cdot \vec{\mathbf{F}} = \mathbf{S}$ is the most general form of an exact conservation law with $\mathbf{F}$ being the flux of the quantity of interest and $\mathbf{S}$ being a possible source term. The errors $\epsilon_{\mathbf{O}}$ and $\epsilon_{\hat{\mathbf{Q}}}$ are random draws from normally distributed error distributions associated with unbiased observations and the model state, respectively. In simpler terms, they account for the inherent randomness or uncertainty in both observed data and model predictions. With these errors, this stochastic presentation allows DA to effectively blend observational data with model predictions while considering uncertainties from both sources.

2.5.2 Solution Process

The solution process for an analysis cycle of the MLES is now explained. Additionally, the pseudocode is given in Algorithm 1. This assumes that the forward CFD model has already propagated the ensemble (including N_e members and 1 control case), to the beginning of the DA window.

We begin by factorizing the error covariance in ensemble subspace,

$$\mathbf{P}_f = \mathbf{P}_f^{1/2}\mathbf{P}_f^{\mathrm{T}/2}\,, \tag{2.4}$$

with the columns of $\mathbf{P}_f^{1/2}$ being

$$\mathbf{P}_{f,i}^{1/2} = \frac{\hat{\mathbf{Q}}_i^f - \hat{\mathbf{Q}}_0}{\sqrt{N_e - 1}}\,. \tag{2.5}$$

$\hat{\mathbf{Q}}^f$ is the forecast deterministic state vector and $\mathbf{Q}_i^f$ is the state vector for each forecast ensemble member. What follows is a typical change of variable seen in variational DA methods where we define the new control vector

$$\hat{\mathbf{Q}}_0 = \hat{\mathbf{Q}}^f + \mathbf{P}_f^{1/2}\boldsymbol{\psi}\,, \tag{2.6}$$

where the new control variable $\boldsymbol{\psi}$ is dimensionless. This is necessary because the state vector $\hat{\mathbf{Q}}_0$ is generally multivariate and therefore can include variables with different orders of magnitude, and it is not feasible to invert the forecast error covariance in a high-dimensional problem. Define the new cost function for $\boldsymbol{\psi}$

$$\begin{aligned} J(\boldsymbol{\psi}) =& \frac{1}{2}\boldsymbol{\psi}^{\mathrm{T}}\boldsymbol{\psi} \\ &+ \frac{1}{2}\sum_{k=L-S+1}^{L}\left[\mathbf{O}_k - \mathcal{H}\circ\mathcal{M}_{0:k}(\hat{\mathbf{Q}}^f + \mathbf{P}_f^{1/2}\boldsymbol{\psi})\right]^{\mathrm{T}}\mathbf{R}_k^{-1}\left[\mathbf{O}_k - \mathcal{H}\circ\mathcal{M}_{0:k}(\hat{\mathbf{Q}}^f + \mathbf{P}_f^{1/2}\boldsymbol{\psi})\right]\,. \end{aligned} \tag{2.7}$$

At each time index k, define the matrix $\mathbf{Z}_k$, which is a collection of N_e columns, $\mathbf{Z}_{k,i}$, defined as follows:

$$\mathbf{Z}_{k,i} = \mathbf{R}^{-1/2}\mathbf{H}\mathbf{M}_{0:k}\mathbf{P}_{f,i}^{1/2} \tag{2.8}$$

$$\approx \frac{1}{\epsilon}\mathbf{R}^{-1/2}\left[\mathcal{H}\circ\mathcal{M}_{0:k}(\hat{\mathbf{Q}}_0 + \epsilon\mathbf{P}_{f,i}^{1/2}) - \mathcal{H}\circ\mathcal{M}_{0:k}(\hat{\mathbf{Q}}_0)\right] , \tag{2.9}$$

where $\mathbf{R}$ is assumed to be constant throughout the DA window and thus the time indices are dropped herein. $\mathbf{H}$ and $\mathbf{M}_{0:k}$ are the tangent linear operators of $\mathcal{H}(\cdot)$ and $\mathcal{M}(\cdot)$, respectively, and ϵ is either a small constant or equal to unity. A finite-difference approximation is used in Eq. (2.9) to approximate the tangent linear operators in Eq. (2.8). Define the gradient, $\boldsymbol{\nabla} J_{\boldsymbol{\psi}}$, and the Hessian (second derivative), $\boldsymbol{\nabla}^2 J_{\boldsymbol{\psi}}$, of Eq. (2.7) as follows:

$$\boldsymbol{\nabla} J_{\boldsymbol{\psi}} = \boldsymbol{\psi} - \sum_{k=L-S+1}^{L} \mathbf{Z}_k^{\mathrm{T}}\mathbf{R}^{-1/2}\left[\mathbf{O}_k - \mathcal{H}\circ\mathcal{M}_{0:k}(\hat{\mathbf{Q}}^f + \mathbf{P}_f^{1/2}\boldsymbol{\psi})\right] , \tag{2.10}$$

$$\boldsymbol{\nabla}^2 J_{\boldsymbol{\psi}} = \mathbf{I} + \mathbf{C} , \tag{2.11}$$

where

$$\mathbf{C} = \sum_{k=L-S+1}^{L} \mathbf{Z}_k^{\mathrm{T}}\mathbf{Z}_k . \tag{2.12}$$

A rigorous derivation of Eq. (2.11) and Eq. (2.12), as well as a further explanation about the finite-difference approximation can be found in Appendix A.

In general, when the condition number of the Hessian, $K(\boldsymbol{\nabla}^2 J_{\boldsymbol{\psi}})$, becomes larger, more iterations are needed to minimize the cost function. $K(\boldsymbol{\nabla}^2 J_{\boldsymbol{\psi}})$, can be calculated by

$$K(\boldsymbol{\nabla}^2 J_{\boldsymbol{\psi}}) = \frac{|\lambda_{\text{max}}|}{|\lambda_{\text{min}}|} , \tag{2.13}$$

where λ_{min} and λ_{max} are the smallest and largest eigenvalues of the Hessian matrix, respectively. To further optimize the problem, the optimal Hessian preconditioning introduces another change of variable

$$\boldsymbol{\psi} = (\mathbf{I} + \mathbf{C}_f)^{-\mathrm{T}/2}\, \boldsymbol{\xi}\,, \tag{2.14}$$

where

$$\mathbf{C}_f = \sum_{k=L-S+1}^{L} \mathbf{Z}_{f,k}^{\mathrm{T}} \mathbf{Z}_{f,k} \tag{2.15}$$

and the columns of $\mathbf{Z}_{f,k}$ are

$$\mathbf{Z}_{f,k,i} \approx \frac{1}{\epsilon}\mathbf{R}^{-1/2}\left[\mathcal{H} \circ \mathcal{M}_{0:k}(\hat{\mathbf{Q}}^f + \epsilon \mathbf{P}_{f,i}^{1/2}) - \mathcal{H} \circ \mathcal{M}_{0:k}(\hat{\mathbf{Q}}^f)\right] . \tag{2.16}$$

There is a deliberate difference between $\mathbf{Z}_f$ and $\mathbf{Z}_k$ because $\mathbf{C}_f$ must be calculated using the initial forecast control (i.e. before minimization), $\hat{\mathbf{Q}}^f$. The gradient is therefore

$$\boldsymbol{\nabla}_{\boldsymbol{\xi}} J = (\mathbf{I} + \mathbf{C}_f)^{-\mathrm{T}/2}\, \boldsymbol{\nabla}_{\boldsymbol{\psi}} J\,, \tag{2.17}$$

and the Hessian is

$$\boldsymbol{\nabla}^2_{\boldsymbol{\xi}} J = (\mathbf{I} + \mathbf{C}_f)^{-\mathrm{T}/2}\, \boldsymbol{\nabla}^2_{\boldsymbol{\psi}} J\, (\mathbf{I} + \mathbf{C}_f)^{-1/2} . \tag{2.18}$$

Indeed, with the optimal Hessian preconditioner, and assuming linear observation and forward model operators, the Hessian is the identity matrix.

2.5.3 Minimization

To find the optimal/analysis control vector, $\hat{\mathbf{Q}}^a$, the minimization process uses a nonlinear iterative method, such as quasi-Newton or nonlinear conjugate-gradient, to find the optimal value of $\boldsymbol{\xi}_{\text{opt}}$. In the current study, the minimization method used is the Newton method. Of course, more complex minimization techniques can be used and are more efficient for larger nonlinear unconstrained optimization problems. However, using the optimum control variable has shown that the Newton method proves to be an efficient method of minimizing the optimized cost function. The problem of minimizing a function of n variables,

$$\min f(\mathbf{x}), \quad \mathbf{x} \in \mathrm{I\!R}^{\mathrm{n}\times 1}\,, \tag{2.19}$$

where $f(\mathbf{x})$ is a smooth function such that the gradient $\boldsymbol{\nabla} f$ exists and can be solved iteratively using

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k\,, \tag{2.20}$$

where k indicates the number of iterations (not time indices in this particular case), α_k is defined as the step-length value that can vary in each iteration and is determined by the minimization method. If using the Newton's method, $\alpha_k = 1$ and the descent direction $\mathbf{d}$ can be solved using

$$\mathbf{d}_k = -\left(\boldsymbol{\nabla}^2 f(\mathbf{x})\right)^{-1} \boldsymbol{\nabla} f(\mathbf{x})\,. \tag{2.21}$$

By using these steps and following Eq. (2.20) and Eq. (2.21), our minimization update equations are

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \mathbf{d}_k\,, \tag{2.22}$$

and the descent direction is

$$\mathbf{d}_k = -\left(\boldsymbol{\nabla}^2_{\boldsymbol{\xi}} J(\hat{\mathbf{Q}}_k)\right)^{-1} \boldsymbol{\nabla}_{\boldsymbol{\xi}} J(\hat{\mathbf{Q}}_k)\,. \tag{2.23}$$

The process starts with an initial guess of $\boldsymbol{\xi}_{k=0} = 0$, which implies

$$\hat{\mathbf{Q}}_{k=0} = \hat{\mathbf{Q}}^f\,. \tag{2.24}$$

Because of the optimal Hessian preconditioning, typically only 2-5 minimization iterations are required. However, a more rigorous convergence criteria can also be used. For example, minimization can be stopped when the L_2-norm difference between the previous iteration and the current iteration is of low enough value given by

$$||\Delta \mathbf{Q}_{a,p}|| < e\,, \tag{2.25}$$

where $\Delta \mathbf{Q}_{a,p}$ is the state update for the next iteration p, and e is a small constant. Beginning with $e = 1e-4$ is an adequate initial value and can be adjusted to better match the current scale of the state vector.

For each minimization iteration, an updated value of $\boldsymbol{\xi}$ is found from Eq. (2.22), which is used to update the state vector, given by

$$\hat{\mathbf{Q}}_0 = \hat{\mathbf{Q}}^f + \mathbf{P}_f^{1/2}\left[\mathbf{I} + \sum_{k=L-S+1}^{L} \mathbf{Z}_{f,k}^{\mathrm{T}} \mathbf{Z}_{f,k}\right]^{-1/2} \boldsymbol{\xi}\,. \tag{2.26}$$

This new state estimate is then used for the next minimization iteration. After the convergence criteria is met and the optimal $\boldsymbol{\xi}_{\text{opt}}$ is found, $\hat{\mathbf{Q}}^a$ can be calculated using

$$\hat{\mathbf{Q}}^a = \hat{\mathbf{Q}}^f + \mathbf{P}_f^{1/2}\left[\mathbf{I} + \sum_{k=L-S+1}^{L} \mathbf{Z}_{f,k}^{\mathrm{T}} \mathbf{Z}_{f,k}\right]^{-1/2} \boldsymbol{\xi}_{\text{opt}}\,. \tag{2.27}$$

This results in the analysis error covariance, $\mathbf{P}_a^{1/2}$, given by

$$\mathbf{P}_a^{1/2} = \mathbf{P}_f^{1/2}\left[\mathbf{I} + \sum_{k=L-S+1}^{L} \mathbf{Z}_k^{\mathrm{T}} \mathbf{Z}_k\right]^{-1/2}\,, \tag{2.28}$$

where $\mathbf{Z}_k$ is calculated using Eq. (2.9) using $\hat{\mathbf{Q}}^a$. $\mathbf{P}_a^{1/2}$ is used to calculate the initial conditions for ensemble forecasting to the next DA cycle,

$$\hat{\mathbf{Q}}^i = \hat{\mathbf{Q}}^a + \mathbf{p}_i^a\,, \quad i = 1, \ldots, N_e\,, \tag{2.29}$$

where $\mathbf{p}_i^a$ are the columns of $\mathbf{P}_a^{1/2}$. Now apply the forward model, $\mathcal{M}$, to propagate the new initial values to the next DA window.

Algorithm 1 MLES analysis cycle

Calculate $\mathbf{P}_f$ in ensemble subspace
while $error > tolerance$ **do**
  Run $\mathcal{M}_{0:L}(\hat{\mathbf{Q}}_0)$ and the ensemble, $\mathcal{M}_{0:L}(\mathbf{Q}_0^{1:N_e})$, for the entire DA window
  Calculate $\mathbf{Z}_{f,k}$ from Eq. (2.16), and $\mathbf{C}_f$ from Eq. (2.15)
  Calculate $\mathbf{Z}_k$ from Eq. (2.9), and $\mathbf{C}$ from Eq. (2.12)
  Calculate the gradient and the Hessian of the cost function
  Use Newton method to update $\boldsymbol{\xi}$ Eq. (2.22) and $\hat{\mathbf{Q}}_0$ from Eq. (2.26)
  Calculate the $error$ from Eq. (2.25) (if not using a specified number if iterations)
end while
Use $\hat{\mathbf{Q}}^a$ from Eq. (2.28), from $\hat{\mathbf{Q}}^a$
Update the ensemble from Eq. (2.29)
return $\hat{\mathbf{Q}}^a$, $\hat{\mathbf{Q}}^{1:N_e}$

2.6 Multigrid in Time Applied to MLES

2.6.1 Motivation

One drawback associated with the MLES lies in the considerable demand for computational resources and time during the iterative analysis procedure. This is because the forward model must undergo multiple runs throughout the DA window for successful minimization. The number of model runs in the DA window is determined by the ensemble size (typically 6 to 30) and the required minimization iterations (typically 2 to 5). Although crucial for optimization, the need for multiple forward model simulations can present challenges in practical applications, particularly when dealing with computationally expensive numerical algorithms. To decrease this computational expense, our dissertation introduces the application of multigrid-in-time (MGinT) as a strategic solution to decrease the number of minimization iterations using the fine time grid.

The overall goal of multigrid-in-time (MGinT) is to increase the computational efficiency while straightforward to implement. The strategy uses time step size coarsening which is used to rapidly obtain a coarse-grid estimate of the optimal analysis state. Assuming the coarse-grid estimate proves sufficient, it serves as the starting point for subsequent fine-grid minimization. The compatibility between MLES and MGinT positions the latter as an ideal choice, requiring minimal adjustments to both the existing MLES framework and the forward model. With the fine-time-grid minimization procedure commencing much closer to the optimal analysis, a significant reduction in iterations on the more computationally expensive fine-time-grid model is achieved, leading to a notable reduction in computational cost. Specifically, the forward model employed in the cost function can use different time resolutions as needed without inhibiting the foundation of the solution procedure. These advancements result in substantial computational savings, meaning the MLES will be more practical and accessible for large-scale DA problems. This approach is a novel application of MGinT to a DA method such as the MLES, marking the first time such a technique has been employed in this context.

2.6.2 Procedure

Figure 2.4 illustrates the core process of the integration of the MLES $+$ MGinT algorithm. The figure shows a 4-grid multigrid-in-time where the optimization constrained by the nonlinear forward model can be solved on a sequence of low-resolution time-grids, indicated by the levels l_1, l_2, and l_3. The result is coarse-time-grid estimates of the analysis at the respective levels. These estimates are then used as the initial starting points for the optimization for the fine-time-grid levels, and eventually the optimization takes place on the finest-time-grid level, indicated by l_0. This process can be generalized for any number of levels of multigrid, and any configuration of multigrid cycles. The procedure shifts to the finer time-grid upon meeting the convergence criteria, which is given in Eq. (2.25).

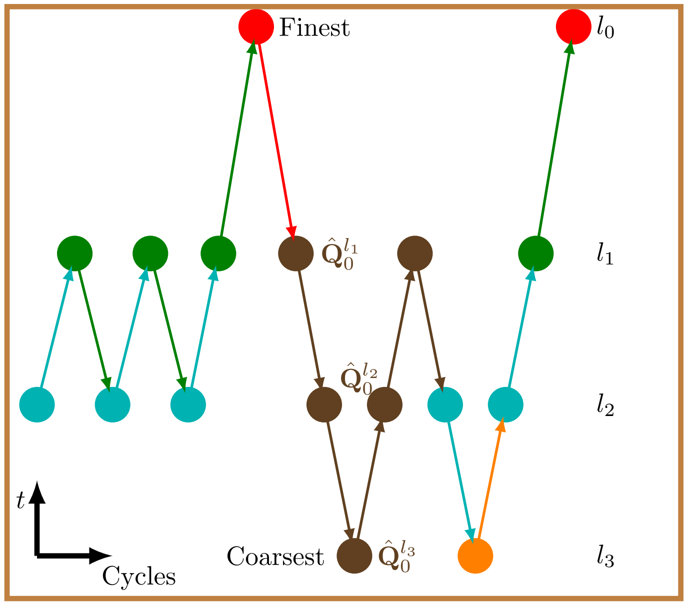

Figure 2.4: An example illustration of MGinT cycles in time. Levels 0, 1, 2, and 3 indicate the meshes from the finest to the coarsest in time, and, by applying the MLES, solutions of $\hat{\mathbf{Q}}_0$ at levels l_1, l_2, and l_3 can be found.

The process for MGinT for a 2-grid example is now described. The algorithm pseudocode is given in Algorithm 2.

1. First, run the forward model using the coarse time grid:

$$\hat{\mathbf{Q}}_N = \mathcal{M}_0^{N,\text{coarse}}(\hat{\mathbf{Q}}_0) \tag{2.30}$$

$$\mathbf{Q}_N^i = \mathcal{M}_0^{N,\text{coarse}}(\mathbf{Q}_0^i)\,, \quad i = 1, \ldots, N_e\,. \tag{2.31}$$

$\hat{\mathbf{Q}}_0$ and $\mathbf{Q}_0^i$ are the control and members in the ensemble at initial time, $t = t_0$, and $\hat{\mathbf{Q}}_N$ and $\mathbf{Q}_N^i$ are the control and members in the ensemble at final time $t = t_N$ of the assimilation window. Note the use of the coarse model propagator, $\mathcal{M}_0^{N,\text{coarse}}(\cdot)$ from time $t = t_0$ to $t = t_N$. Graphically, we can depict an example of this process which can be seen in Figure 2.5. Here, $N = 5$ and $N_e = 3$. The result of applying the coarse forward model means the state values are only available at times t_0, t_2, and t_5 while leaving gaps at times t_1 and t_3. This means the summation term in the cost function will lack the terms located at these time points. Therefore, the minimization produces a coarse-grid optimal estimate of $\hat{\mathbf{Q}}_0$ and $\mathbf{Q}_0^i$.

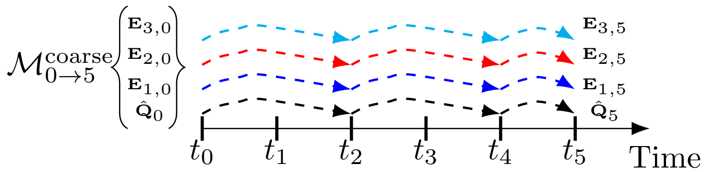

Figure 2.5: Running the coarse-grid forward model to the end of the assimilation window. State variables are obtained at times t_0, t_2, and t_5 while not acquiring information at times t_1 and t_3.

2. Use this coarse-grid estimate and apply it to the minimization procedure on the fine grid. Starting with the coarse-grid estimate of $\hat{\mathbf{Q}}_0$ and $\mathbf{Q}_0^i$, run the forward model using the fine time grid:

$$\hat{\mathbf{Q}}_N = \mathcal{M}_0^{N,\text{fine}}(\hat{\mathbf{Q}}_0) \tag{2.32}$$

$$\mathbf{Q}_N^i = \mathcal{M}_0^{N,\text{fine}}(\mathbf{Q}_0^i)\,, \quad i = 1, \ldots, N_e\,. \tag{2.33}$$

This is graphically shown in Figure 2.6.

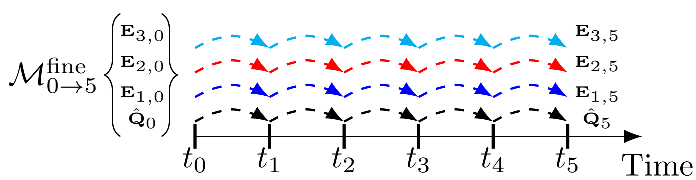

Figure 2.6: Running the fine-grid forward model to the end of the assimilation window. State variables are obtained at all times.

3. Now that all of the information is available, the minimization of the cost function produces the optimal estimate of $\hat{\mathbf{Q}}$ and $\mathbf{Q}^i$ at the beginning of the time window.

Algorithm 2 MLES+MGinT analysis cycle

Calculate $\mathbf{P}_f$ in ensemble subspace
for Multigrid levels **do**
  Adjust appropriate Δt for the current MG level
  while $error > tolerance$ **do**
    Run $\mathcal{M}_{0:L}(\hat{\mathbf{Q}}_0)$ and the ensemble, $\mathcal{M}_{0:L}(\mathbf{Q}_0^{1:N_e})$, for the entire DA window
    Calculate $\mathbf{Z}_{f,k}$ from Eq. (2.16), and $\mathbf{C}_f$ from Eq. (2.15)
    Calculate $\mathbf{Z}_k$ from Eq. (2.9), and $\mathbf{C}$ from Eq. (2.12)
    Calculate the gradient and the Hessian of the cost function
    Use Newton method to update $\boldsymbol{\xi}$ Eq. (2.22) and $\hat{\mathbf{Q}}_0$ from Eq. (2.26)
    Calculate the $error$ from Eq. (2.25)
  end while
  Calculate $\hat{\mathbf{Q}}^a$ to be used as $\hat{\mathbf{Q}}_0$ for the next multigrid time level
end for
Use $\hat{\mathbf{Q}}^a$ from Eq. (2.28), from $\hat{\mathbf{Q}}^a$
Update the ensemble from Eq. (2.29)
return $\hat{\mathbf{Q}}^a$, $\hat{\mathbf{Q}}^{1:N_e}$

Chapter 3

Numerical Framework

The current chapter describes the numerical framework developed for this dissertation. Typically, the software needed to incorporate the forward model is as important as the DA algorithms themselves to ensure a robust architecture. This is especially true when using high-dimensional fluid simulations with large amounts of data. First, it will provide an overview of Chord [39–44], our in-house numerical model used for complex fluid dynamics simulations. It describes the mathematical components that distinguish Chord as a unique, high-order finite-volume-method (FVM) CFD solver. These include the high-order spatial discretizations, the usage of mapped multiblock grids (MMB) for accommodating complex geometries, and the time marching method for efficient time-to-solution. Secondly, the framework to couple DA with the CFD is described. This details how the DA methods are integrated into Chord.

3.1 CFD Forward Model (Chord)

In this section, we discuss our in-house high-order FVM CFD code, Chord, which is used for the numerical results found later in the thesis. Chord stands out for its capability to achieve high-order accuracy in both time and space, enabling accurate modeling of fluid flows encompassing turbulence and combustion phenomena. Notably, it incorporates adaptive mesh refinement (AMR) techniques in both spatial and temporal domains, allowing for efficient handling of complex geometries while maintaining free stream conditions through the MMB technique. Moreover, Chord is optimized for performance on high-performance computing (HPC) architectures. The subsequent sections provide the details of the features found in Chord.

3.1.1 Governing Equations

On a rectangular physical domain, identical to the computational domain prior to applying any coordinate transformation, the system of partial differential equations (PDEs) governing a compressible gas comprises continuity, momentum, energy, and a set of species transport equations if chemical reactions are considered, given by,

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{u}) = 0\,, \tag{3.1}$$

$$\frac{\partial}{\partial t}(\rho \vec{u}) + \vec{\nabla} \cdot \left(\rho \vec{u}\vec{u} + p\vec{\vec{I}}\right) = \vec{\nabla} \cdot \vec{\vec{\tau}} + \rho \vec{f}\,, \tag{3.2}$$

$$\frac{\partial}{\partial t}(\rho e) + \vec{\nabla}\cdot \left[\rho \vec{u}\left(e + \frac{p}{\rho}\right)\right] = \vec{\nabla}\cdot\left(\vec{\vec{\tau}} \cdot \vec{u}\right) - \vec{\nabla} \cdot \vec{q} + \rho \vec{f} \cdot \vec{u}\,, \tag{3.3}$$

$$\frac{\partial}{\partial t}(\rho c_n) + \vec{\nabla} \cdot (\rho c_n \vec{u}) = -\,\vec{\nabla} \cdot \vec{\mathcal{J}}_n + \rho \dot{w}_n\,, \quad n = 1, \ldots, N_g\,, \tag{3.4}$$

where ρ is the density, $\vec{u}$ is the velocity vector, p is pressure, $\vec{\vec{I}}$ is the identity tensor, and $e = |\vec{u}|^2/2 + \sum c_n h_n - p/\rho$ is the total specific energy with c_n and h_n being the mass fraction and the specific enthalpy of species n, respectively. The pressure is solved by the ideal gas law $p = \rho R_u T$, where R_u is the universal gas constant and T is the temperature. $\vec{f}$ is a body force per unit volume acting on the gas, but is negligible in the present work. $\vec{q}$ is the molecular heat flux vector. The molecular fluid stress tensor, $\vec{\vec{\tau}}$, is given as

$$\vec{\vec{\tau}} = 2\mu\left(\vec{\vec{S}} - \frac{1}{3}\vec{\vec{I}}\vec{\nabla} \cdot \vec{u}\right)\,, \tag{3.5}$$

where $\vec{\vec{S}}$ is the strain rate tensor and μ is the fluid molecular viscosity. A total of N_g species comprise the gaseous mixture, with N_g transport equations. $\vec{\mathcal{J}}_n$ and $\dot{w}_n$ are the species diffusive flux and the production rate of species n, respectively. In the current study, the thermally-perfect neglects gas mixture disregards Dufour, Soret, and radiative heat transfer effects.

3.1.2 Grid Mapping

To accommodate complex geometries, the MMB technique is employed to work with structured AMR grids [45]. The transformation grid metrics, N^{T}, and metric Jacobian, J, are introduced into the system of governing equations to transform from physical space, $\vec{x}$, into computational space, $\vec{\xi}$, where $\vec{\xi} = \vec{\xi}(\vec{x})$. They are defined as

$$\mathrm{N} = J\left(\vec{\nabla}_x \vec{\xi}\right)^{\mathrm{T}}, \quad \mathrm{N}^{\mathrm{T}} = J\vec{\nabla}_x \vec{\xi}, \quad J \equiv \det\left(\vec{\nabla}_\xi \vec{x}\right). \tag{3.6}$$

After applying the coordinate transformation to Eqs. (3.1) - (3.4), the governing equations are now

$$\frac{\partial (J\mathbf{U})}{\partial t} + \vec{\nabla}_\xi \cdot \left(\mathrm{N}^{\mathrm{T}}\vec{\mathbf{F}} - \mathrm{N}^{\mathrm{T}}\vec{\boldsymbol{\mathcal{G}}}\right) = J\mathbf{S}\,, \tag{3.7}$$

where $\mathbf{U}$ is the solution vector, $\vec{\mathbf{F}}$ is the inviscid flux vector, $\vec{\boldsymbol{\mathcal{G}}}$ is the viscous flux vector, and $\mathbf{S}$ is the reaction source vector. They are defined as

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho\vec{u} \\ \rho e \\ \rho c_n \end{pmatrix}, \vec{\mathbf{F}} = \begin{pmatrix} \rho\vec{u} \\ \rho\vec{u}\vec{u} + p\vec{\vec{I}} \\ \rho\vec{u}\left(e + \frac{p}{\rho}\right) \\ \rho c_n \vec{u} \end{pmatrix}, \vec{\boldsymbol{\mathcal{G}}} = \begin{pmatrix} 0 \\ \vec{\vec{\mathcal{T}}} \\ \vec{\vec{\mathcal{T}}} \cdot \vec{u} - \vec{\mathcal{Q}} \\ -\vec{\mathcal{J}}_n \end{pmatrix}, \mathbf{S} = \begin{pmatrix} 0 \\ \rho\vec{f} \\ \rho\vec{f}\cdot\vec{u} \\ \rho\dot{w}_n \end{pmatrix}. \tag{3.8}$$

The mapped stress tensor, $\vec{\vec{\mathcal{T}}}$, is given by

$$\vec{\vec{\mathcal{T}}} = 2\mu\left(\vec{\vec{\mathcal{S}}} - \frac{1}{3}J^{-1}\vec{\vec{I}}\vec{\nabla}_\xi \cdot \left(\mathrm{N}^{\mathrm{T}}\vec{u}\right)\right), \tag{3.9}$$

where the mapped strain rate tensor, $\vec{\vec{\mathcal{S}}}$, is updated by

$$\vec{\vec{\mathcal{S}}} = \frac{1}{2}\left((\vec{\nabla}_\xi \vec{u})\left(\frac{\mathrm{N}^{\mathrm{T}}}{J}\right) + \left(\frac{\mathrm{N}^{\mathrm{T}}}{J}\right)^{\mathrm{T}}(\vec{\nabla}_\xi \vec{u})^{\mathrm{T}}\right). \tag{3.10}$$

The mapped molecular heat flux, $\vec{\mathcal{Q}}$, is modeled by

$$\vec{\mathcal{Q}} = -\left(\kappa\frac{\mathbf{N}}{J}\vec{\nabla}_\xi T - \sum_{n=1}^{N_g}\left(h_n\vec{\mathcal{J}}_n\right)\right), \tag{3.11}$$

where κ is the thermal conductivity coefficient and $\vec{\mathcal{J}}_n$ is the mapped mass diffusion of species n modeled by the Fourier's law, given by

$$\vec{\mathcal{J}}_n = -\rho D_n \frac{\mathbf{N}}{J} \vec{\nabla}_\xi \vec{u}\,. \tag{3.12}$$

In the equation, D_n is defined as the molecular diffusivity for species n.

3.1.3 Spatial Discretization

FVMs have typically been confined to second-order accuracy [46], while higher-order methods introduce complexity in terms of implementation and stability. However, increasing the order of accuracy from second to fourth enhances the convergence rate of solution errors as the grid refines in smooth flows [47]. In this study, our forward model is a FVM solver with a fourth-order accurate approximation in both space and time for smooth flows. Continuing with the governing equations of Eq. (3.7), the divergence form is initially integrated over control volumes as

$$\frac{\partial}{\partial t} \int_{V_i} J\mathbf{U}\,\mathrm{d}V + \int_{V_i} \left(\vec{\nabla}_\xi \cdot \left(\mathbf{N}^{\mathrm{T}} \vec{\mathbf{F}} - \mathbf{N}^{\mathrm{T}} \vec{\boldsymbol{\mathcal{G}}} \right) - J\mathbf{S} \right) \mathrm{d}V = 0\,. \tag{3.13}$$

On a rectangular computational domain, the cell centers are marked by the points $(i_0, ..., i_{\mathrm{D}-1}) = \boldsymbol{i} \in \mathbb{Z}^{\mathrm{D}}$. The control volumes are thus $V_{\boldsymbol{i}}$ and take the form $V_{\boldsymbol{i}} = [\boldsymbol{x}_0 + (\boldsymbol{i} - \frac{1}{2}\boldsymbol{u})\Delta x, \boldsymbol{x}_0 + (\boldsymbol{i} + \frac{1}{2}\boldsymbol{u})\Delta x]$, where $\boldsymbol{x}_0 \in \mathbb{R}^{\mathrm{D}}$ is a fixed origin of coordinates, Δx is the mesh spacing, and $\boldsymbol{u}$ is the vector whose components are all equal to one. Representing the integrals as averages and applying the divergence theorem, we get the semi-discrete form

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \langle J\mathbf{U} \rangle_{\boldsymbol{i}} = -\frac{1}{\Delta x^d} \sum_{d=1}^{D} \Bigg(&\left(\langle J\vec{\nabla}_d \vec{\xi}\, \vec{\mathbf{F}} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle J\vec{\nabla}_d \vec{\xi}\, \vec{\mathbf{F}} \rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d} \right) \\ &- \left(\langle J\vec{\nabla}_d \vec{\xi}\, \vec{\boldsymbol{\mathcal{G}}} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle J\vec{\nabla}_d \vec{\xi}\, \vec{\boldsymbol{\mathcal{G}}} \rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d} \right) \Bigg) + \langle J\mathbf{S} \rangle_{\boldsymbol{i}}\,, \end{aligned} \tag{3.14}$$

where the $\langle \cdot \rangle$ indicates cell-averaged or face-averaged quantities, and the superscript d is the direction of the flux vector. A fourth-order center-differencing scheme is employed for computing face-averaged quantities and gradients as inputs for the flux evaluation. Additional techniques are exercised during the transformation from conservative to primitive variables to maintain fourth-order accuracy during smooth-flow regimes. The stencil for computing the face-averaged primitive variables, $\langle \mathbf{W} \rangle$, uses four neighboring cells in each coordinate direction, by

$$\langle \mathbf{W} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{7}{12}(\langle \mathbf{W} \rangle_{\boldsymbol{i}} + \langle \mathbf{W} \rangle_{\boldsymbol{i}+\boldsymbol{e}^d}) - \frac{1}{12}(\langle \mathbf{W} \rangle_{\boldsymbol{i}-\boldsymbol{e}^d} + \langle \mathbf{W} \rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}) . \tag{3.15}$$

Then, the inviscid face-averaged fluxes $\langle \mathbf{F} \rangle^d_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ can be computed by $\langle \mathbf{F} \rangle^d_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \mathbf{F}^d(\mathbf{W}^d_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}) + \frac{\Delta x^2}{24}\Delta^{(d,2)}\mathbf{F}^d(\langle \mathbf{W} \rangle^d_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d})$. In addition, the viscous flux vector requires the fourth-order evaluation of the gradients of primitive variables, $\langle \vec{\nabla} \mathbf{W} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$, in addition to $\langle \mathbf{W} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$. The gradient evaluation is not duplicated here and can be found in Gao et al. [39]. Additional details regarding the stencils for hyperbolic spatial discretization near physical boundaries, including discussions on limiters and dissipation mechanisms, are further discussed in Guzik et al. [40], Gao et. al. [42], and Owen et al. [44]. For the DA integration, the discretization error is neglected, which is considered as a “perfect” model scenario.

3.1.4 Time Marching Method

To advance the solution, we employ the fourth-order standard four-stage Runge-Kutta (RK4) method to evolve Eq. (3.14) for cold flow. However, for reacting flow, small time-step constraints can be placed for the simulation if the combustion is stiff. The small chemical time step due to fast chemistry is addressed by using an implicit-explicit (IMEX) time integration. Specifically, the fourth-order additive Runge-Kutta (ARK_4) method is used and more details can be found in Kennedy et al. [48], Christopher et al. [49]. The ARK_4 method has demonstrated that it can gain computational efficiency by splitting the physics into stiff and non-stiff terms. This is done to address each using implicit and explicit methods, respectively. The convection and diffusion physics are solved explicitly and the chemical reactions are solved implicitly. Christopher et al. [49] showed that ARK_4 has the potential to provide a $70\times$ speedup over the standard RK4 in the 2D test case, and three-orders-of-magnitude-larger time step sizes are achieved for the three-dimensional test case. Although ARK_4 serves as the time marching method in this dissertation, it is not the focus of discussion. Interested readers are directed to the referenced studies for further details.

3.2 DA+CFD Framework

The integration of the CFD forward model and DA algorithms is crucial for seamless data exchange within the software framework. This integration ensures the efficiency of the assimilation process. There are three main steps in the DA+CFD (or "DA+Chord") system, as illustrated in Figure 3.1, including forward model, innovation (aka. observation increment), and analysis. This flow chart uses the symbol of the deterministic state of MLES for illustration. Initially, every member of the ensemble and the control undergoes propagation by the forward CFD model until the observation data becomes available, at which point $\mathbf{P}_f$ is calculated. The CFD model predicts the system's temporal and spatial evolution, with scales influenced by the numerical algorithm employed. Observations, which can be measurements in space and time, rely on the technique used in physical experiments for their spatial and temporal scales. During assimilation, the innovation is computed as the difference between the observation and the initial guess of the forecast state. In the analysis phase, CFD simulations and experimental data are combined to estimate the optimal state based on statistical error analysis criteria. Finally, the analysis state reinitializes the CFD model for each ensemble member to propagate the forward CFD model in subsequent DA cycles.

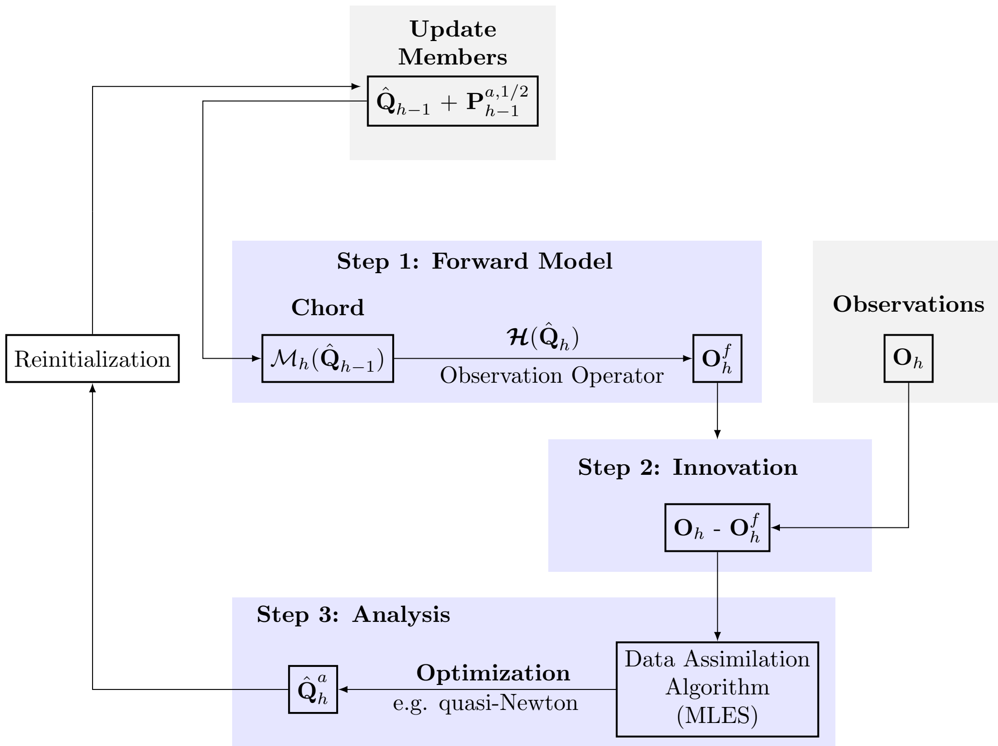

Figure 3.1: Graphical depiction of the DA+CFD system.

To establish a robust DA workflow, the CFD forward model and DA algorithms must communicate with a simple robust framework. The DA procedure needs this efficient data exchange and will enhance the predictive capabilities. Ultimately, the approach will prove to provide a deeper understanding and give a reliable forecast of dynamic systems.

Chapter 4

Verification and Validation

Before applying the MLES to enhance predictions of practical turbulent flows, it must be verified and validated within the DA+CFD system established in this study. In the current chapter, the Lorenz-96 model and the Kuramoto-Sivashinsky equation problem serve to verify and validate the efficacy of the MLES. These are selected as they are a well-established canonical DA and fluid dynamics problems with readily available literature data for benchmarking the framework (e.g. Zupanski [36], Kalnay et al. [7], Jardak et al. [50], and Asch et al. [37], to name a few). By comparing CFD predictions with and without DA, we can effectively evaluate the performance of assimilation through metrics such as χ^2 diagnostics and RMS errors. This assessment shows the reliability and effectiveness of the MLES within the DA+CFD system and can then be used for turbulent flows. In particular, the MLES is benchmarked against the already-proven MLEF. These steps are crucial for ensuring the robustness and accuracy of predictive models in complex fluid dynamics scenarios.

4.1 Lorenz-96 Model

The framework is developed and validated by the well known Lorenz 96 model [51]. This model is chosen because it is used in many validation cases for DA and has literature data for the performance assessment of the framework. Additionally, by choosing a correct forcing term, the model is commonly known to cause chaotic behavior and can therefore provide insight to future turbulent flow simulations. The result of the MLES algorithm is compared to those of the MLEF algorithm. The performance of the MLES algorithm is measured by χ^2 diagnostics, the normalized cost function, and a root-mean-square error (RMSE) analysis. The following provide details of the experimental setup, descriptions of the tests, and the results of the experiments.

The Lorenz 96 model is given as

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = (x_{i+1} - x_{i-2})\, x_{i-1} - x_i + F\,, \quad i = 1, \ldots, M\,. \tag{4.1}$$

A value of $F = 8$ is known to cause chaotic behavior and chosen as such for the validation. In the current study, the number of variables in the test is chosen to be $M = 40$.

4.1.1 Twin Numerical Experiment Setup

In DA problems where methods are verified, the "truth twin experiment" is a common setup that serves as an approach for validation and testing of DA algorithms. This experimental framework involves creating a controlled environment where the true state of the system, referred to as the “truth,” is known. Synthetic observations are then simulated from this true state, incorporating a source of error such as synthetic measurement noise, spatial and temporal resolution limitations, or missing data. For example, once the truth run is initialized, the forward model propagates it to the desired solution time. Observations are synthesized from the truth, which is defined as

$$\mathbf{O}_n = \mathcal{H}\left(\hat{\mathbf{Q}}_{\text{true}}\right) + \boldsymbol{\eta}\,, \tag{4.2}$$

where n is the time index, $\hat{\mathbf{Q}}_{\text{true}}$ is the true state, and $\boldsymbol{\eta}$ is a random perturbation taken from a normal distribution of a chosen variance and zero mean. Subsequently, the DA algorithm is applied to assimilate these simulated observations into a numerical model representing the system dynamics. Following the assimilation process, the performance metrics are used to evaluate the assimilated state estimates against the true state to assess the performance of the assimilation algorithm. Metrics such as RMSE, χ^2 diagnostics, normalized cost function, averaged statistical measured, and other measures are commonly used for this evaluation. Additionally, sensitivity analyses may be conducted by varying parameters such as the observations taken in the domain, ensemble size, or other DA parameters to assess its robustness and sensitivity under different conditions.

4.1.2 Data Assimilation Configuration

There are a number of parameters specific to DA and are explained herein. The choice of N_e greatly determines the ability of the ensemble-based DA algorithms to accurately capture the error in the current system. A relatively low ensemble size ($N_e = 6$ to 8, depending on the system) generally does not accurately capture error and artificially inflating the error is necessary. The time between observations is arbitrarily chosen to be the same as the forward model time step size. This also dictates the length of a DA cycle. The MLEF is performed at every observation point in time and the MLES DA window length is determined by the observation interval as well as L and S parameters. N_O is the number of observations. σ_{obs} is the variance of the normal distribution used for synthesizing observations taken from the truth as shown in Eq. (4.9). The diagonal elements of the observation covariance, $\mathbf{R}$, are unchanging with values σ_{obs}^2. The number of steps to be used in the minimization procedure is based on Zupanski [36]. The DA configuration parameters for the validation are given in Table 4.1. These values are used to gather statistics for the tests

Table 4.1: the Lorenz 96 model experiment's data assimilation parameters.

Variable	Value
N_e	15 members
Observation time interval	0.05 time units
N_O	40 (every computation cell)
S	10
L	10
σ_{obs}	0.05
Diagonal elements of $\mathbf{R}$	0.05^2
Minimization steps	3

in the following sections. Note that the observation error covariance matrix $\mathbf{R}$ only has diagonal elements specified in the table above and stays constant through the entire simulation, i.e., the random realization for the gathered observation data remains the same for each DA cycle. Also note that for the MLEF, one DA cycle consists of running the model to each observation available.

For the MLES, however, one DA cycle for this experiment is 10 times the length of the DA cycle in the MLEF because the number of observations taken in the time window is S.

4.1.3 DA Performance Measures

χ^2 diagnostics

The performance of the MLES relies on the estimation of the input error covariances, $\hat{\mathbf{Q}}$, $\mathbf{R}$, and $\mathbf{P}_f$. These directly influence the analysis covariance, $\mathbf{P}_a$. The χ^2 diagnostics test [26, 36, 52] uses the innovation (i.e. $\mathbf{O}-\mathcal{H}(\hat{\mathbf{Q}})$) covariance matrix to evaluate the correctness of the input error covariances. It is given, in observation space, as

$$\chi^2 = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{N_O}, \quad \mathbf{X} = \left(\mathbf{R} + \mathbf{H}(\hat{\mathbf{Q}})\mathbf{P}_f\mathbf{H}^{\mathrm{T}}(\hat{\mathbf{Q}})\right)^{-1/2}\left(\mathbf{O} - \mathcal{H}(\hat{\mathbf{Q}})\right), \tag{4.3}$$

where $\mathbf{H}(\hat{\mathbf{Q}})$ is the tangent linear model of the observation operator, $\mathcal{H}(\cdot)$. $\hat{\mathbf{Q}}$ is the analysis state at each DA cycle. For a Gaussian distribution of innovations and a linear observation operator, the expected values of χ^2 should be equal to 1.0. However, due to the calculation of $\mathbf{P}_f$ wherein the nonlinear model is used, and also due to the finite number of observations per cycle, the actual values for the assimilated runs will not be exactly 1.0. The result of the test is shown in Figure 4.1. Every DA cycle for the smoother represents 10 complete DA cycles in the MLEF and thus different x-axes are used. Nevertheless, both experiments show the same solution time at the same corresponding coordinate locations. Both MLEF and MLES show $\boldsymbol{\chi}^2$ values close to 1.0 for each DA cycle. The mean $\boldsymbol{\chi}^2$ values for the entire run are 0.9509 for the MLEF and 1.088 for the MLES, respectively. Either by running further experiments or by increasing the number of Gaussian distributed observations, one would expect each subsequent run to have a mean close to 1.0. The free run $\boldsymbol{\chi}^2$ values are expectedly much larger and emphasizes the chaotic behavior of the experiment's model.

Normalized Cost Function

We examine the normalized cost function where the number of observations taken in the spatial domain is the normalization factor and is calculated by

$$\left(\frac{J}{N_O}\right) = \frac{1}{N_O}\left(\frac{1}{2}\left(\mathbf{O} - \mathcal{H}(\hat{\mathbf{Q}})\right)^{\mathrm{T}} \mathbf{R}_j^{-1}\left(\mathbf{O} - \mathcal{H}(\hat{\mathbf{Q}})\right)\right) . \tag{4.4}$$

The cost function J and the control state $\hat{\mathbf{Q}}$ are calculated after the minimization process has occurred and therefore the analysis state is used for Eq. (4.4). The normalized cost function is used to assess the minimization in a multi-variate problem and is related to the χ^2 test. The uncertainty error is reduced and thus the expected values are expected to be approximately 0.5. The results can be seen in Figure 4.2 and every value is close to 0.5.

RMSE

The RMSE is also used to measure the difference between the analysis and the truth. The truth is simply the deterministic simulation by the forward model using perfect conditions and

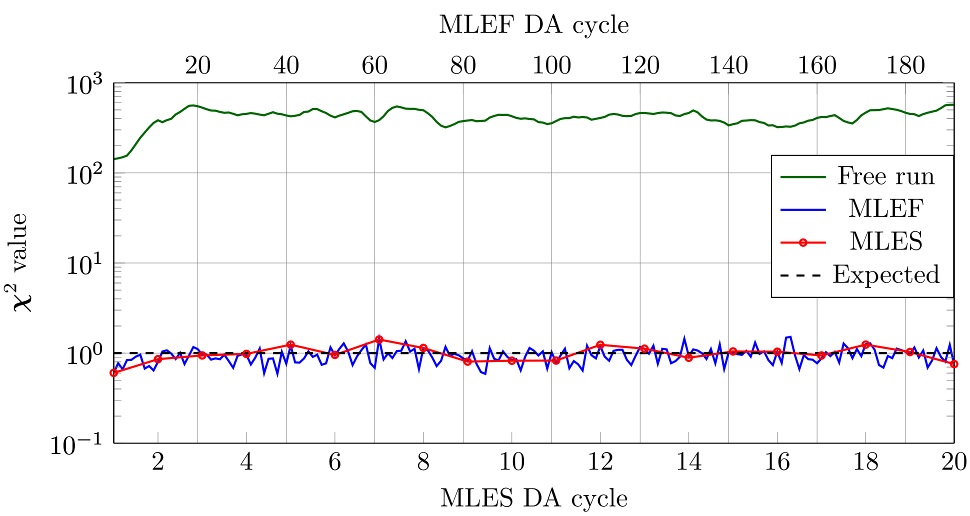

Figure 4.1: The χ^2 diagnostic test for the Lorenz 96 model experiment with a semi-log axis. There are two different x-axes that show the different DA cycles for the MLEF and MLES, respectively, but at the same solution times.

parameters. The RMSE is calculated by

$$\text{RMSE} = \sqrt{\frac{\sum_{m=1}^{N_S} \left(\mathbf{Q}_{\text{true}} - \hat{\mathbf{Q}}^a \right)_m^2}{N_S}} \,. \tag{4.5}$$

The results of the RMSE are seen in Figure 4.21 for each DA cycle and compared between the MLEF and MLES. In addition, the RMSE of the free run is shown and clearly larger than that of the MLEF and MLES, which is expected. Both the MLEF and MLES have similar values and, overall, the RMSE shows a stable convergence with the truth as well as having stable performance throughout the experiment.

Additionally, the first state value, Q_1, is plotted as a function of the time step for the MLES and MLEF and the synthesized data for this particular state value is also shown in Figure 4.4. This particular case setup is the same as described in Section 4.1.2. The first DA update is located at the 11^{th} time step and is the corresponding analysis update for the MLES of the time window. After the MLES analysis update, it is freely forecasted to an end time of 120 time steps. The first analysis point for the MLEF is also at the 11^{th} time step but is updated at every available observation until the last observation which is located at the 20^{th} time step. At this point, it is freely forecasted to

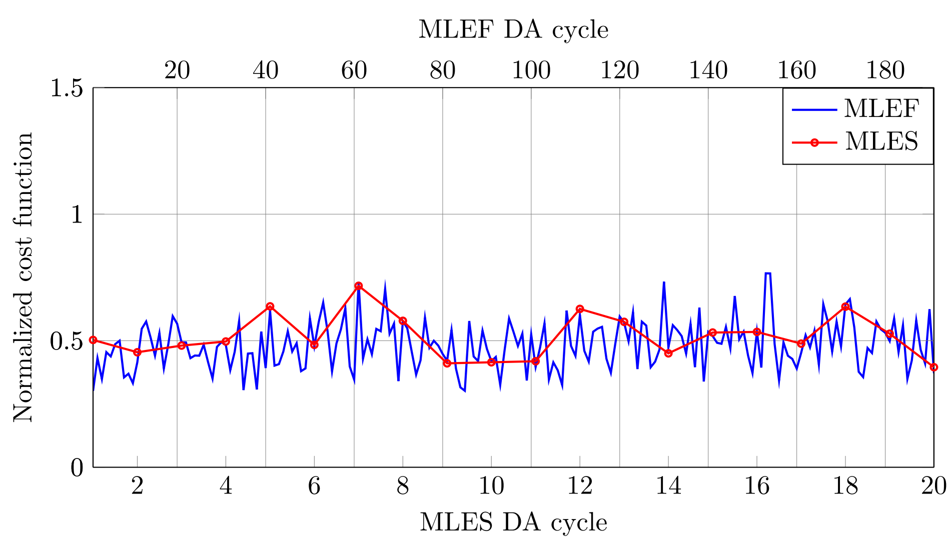

Figure 4.2: The normalized cost function values for the MLEF and MLES. There are two different x-axes that show the different DA cycles but at the same solution times.

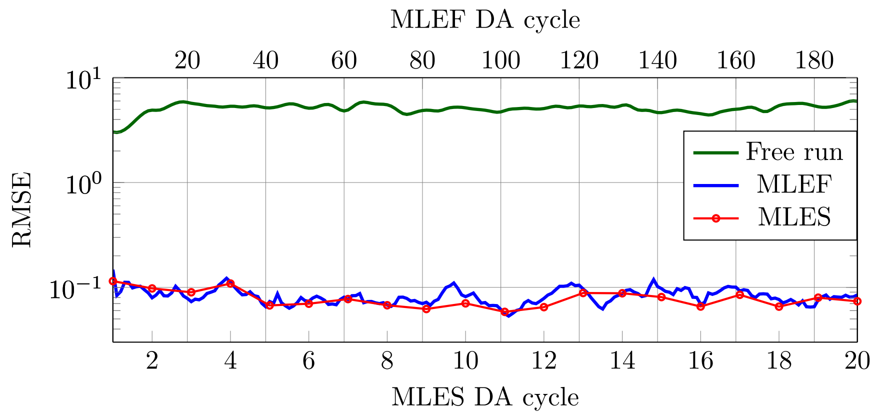

Figure 4.3: The RMSE for the free run, MLEF, and MLES. Two different x-axes showing the same time point while marked by a different DA cycle number, e.g. 3 DA cycles in the MLES corresponding to 20 DA cycles in the MLEF.

the same end time as the MLES. A free run that does not have any DA updates is also included. Fundamentally, Figure 4.4 displays the increase in predictability after DA is applied. Both the

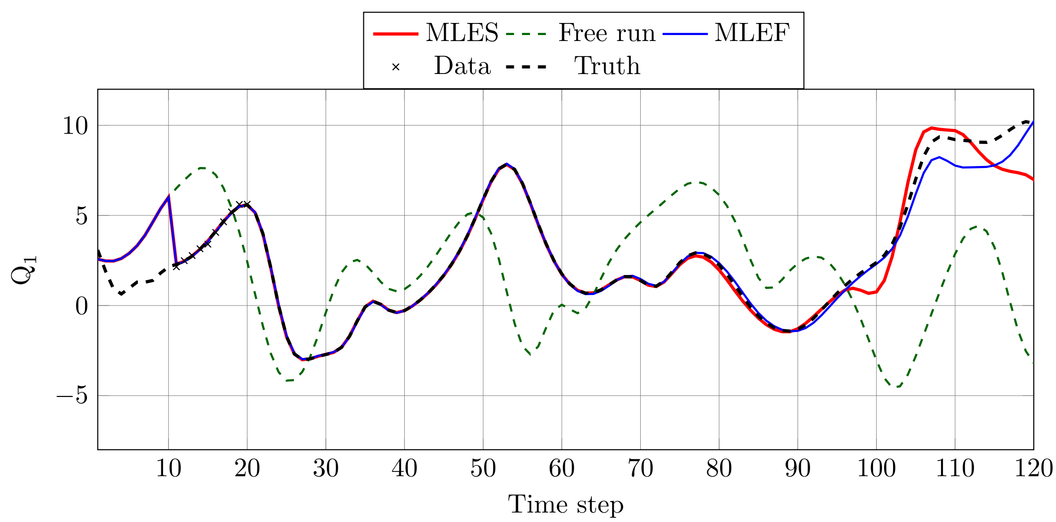

Figure 4.4: The trajectory of Q_1 for the truth, MLES, MLEF, free run, and observations of the Lorenz 96 model over time. Both the MLEF and MLES are first updated at the 10^{th} time step and the MLEF is continually updated until the last observation available. The free run is simply a simulation by the forward model with the same steps as the MLES or MLEF but without DA.

MLEF and MLES diverge from the truth at approximately the 100^{th} time step while the free run completely diverges from the truth very early. The DA simulations have good approximations of the truth state but their trajectories start to deviate from the truth as a consequence of the nature of the chaotic system. It is worth mentioning that the MLEF is competent and efficient in general; it is encouraging and promising to have shown the MLES to have comparable performance as the MLEF for these cases.

4.2 Kuramoto-Sivashinsky Equation

The validated MLES algorithm is now applied to solve the 1D Kuramoto-Sivashinsky equation. Originally developed for modeling instabilities in flame fronts [53, 54], this simulation provides insight to the MLES application to turbulent reacting CFD problems. The algorithm is tested and compared against the original MLEF algorithm.

The model equation is

$$\frac{\partial y}{\partial t} = -a\frac{\partial^2 y}{\partial x^2} - by\frac{\partial y}{\partial x} - c\frac{\partial^4 y}{\partial x^4}\,, \tag{4.6}$$

where, in the truth run, the model parameter coefficients are $a = 1$, $b = 1$, and $c = 1$. The boundary conditions are chosen to be periodic and the computational mesh has 256 cells uniformly distributed over $x \in [-100, 100]$. The initialization of the ensemble is based on the lagged forecast method [7]. That is, given a time interval length of $t \in [0, t_\tau]$, a "burn-in" simulation of length t_τ is run using an arbitrarily chosen initial condition. Given an ensemble of size N_e, one picks N_e normally distributed temporal points over the $[0, t_\tau]$ interval. At each of the chosen temporal points, there is a corresponding state vector, $\mathbf{Q}$, from the "burn-in" simulation which is then used as the initial conditions for the members in the ensemble. The true state is set to be the simulation in the middle of the simulation length and the initial state of the control run is set to be the state average of the members in the ensemble. t_τ is set to be 5 time units. Where applicable, initial uncertainty of the model parameters is based on a normally distributed probability density function. The variance for the synthesized observations is chosen to be $\sigma_{\text{obs}} = 0.001$. The time step of the forward model is $\Delta t = 0.005$ time units.

4.2.1 Number of Time Points With Observations

We first perform an experiment to study the effect of the MLES as a function of S (the number of observations used within a smoothing window). That is, using a fixed time interval between observations, S is varied and the RMSE of the analysis update of the initial DA cycle (i.e., the beginning of the smoothing window) is then calculated. L is equal to S in this numerical experiment to match the last observation in time. Note that by fixing the time interval between observations and adjusting S, the total length of the entire smoothing window changes. Figure 4.5 shows the statistical average of the RMSE calculated over 100 individual runs. S varies from 1 to 50. The fixed time interval between observations is $\Delta t = 0.05$ time units and the domain is fully observed. Also, $N_e = 20$ for each run in this experiment. As a check, the MLEF is included and shows the

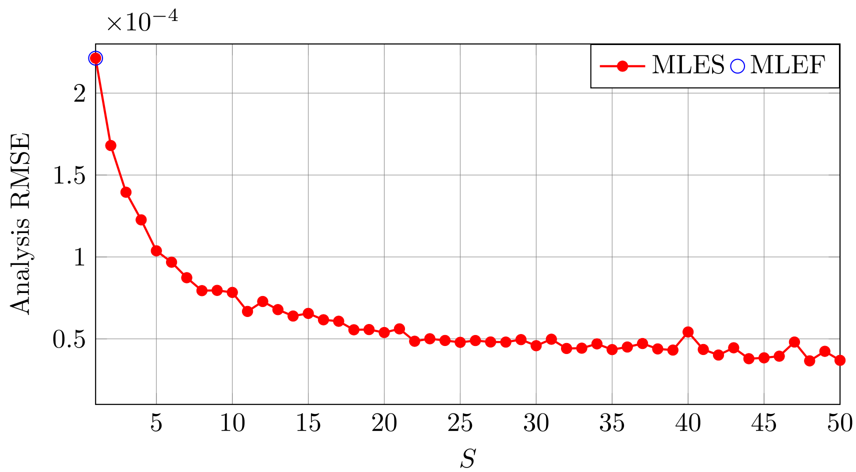

Figure 4.5: The analysis RMSE of the MLES by varying S from 1 to 50. The red dotted points are the statistical average of 100 runs of the RMSE of the analysis of the MLES as a function of S. The blue circle is the MLEF at the analysis point and, as expected, matches the MLES when only one observation is taken in the window.

analysis RMSE matches the MLES when $S = 1$ which is to be expected. Then, as S is increased, the RMSE values of the analysis drop until there is a saturation of performance. We expect the overall performance to eventually degrade as S is increased beyond a certain threshold as the distance between the last observation and the location of the analysis becomes larger. In other words, the introduction of nonlinearity (mainly from the forward model) of a larger smoother window will eventually cause an increase in the error. Nevertheless, Figure 4.5 shows that given more information within a smoothing window, the error is reduced. It is worth noting that while there are small oscillations in the RMSE values (see points $S = 11$, 40, and 46, for example), it is expected that this curve will smooth out as more and more cases are run to help with the averaging. The important result is that the general trend of error reduction in the MLES goes down as more information is provided within a smoothing window. Furthermore, there is a balance between the optimal state and the computational cost for assimilating observations. For example, $S = 18$ provides the desired optimal state.

4.2.2 Ensemble Size

The ensemble size is varied from $N_e = 5$ to $N_e = 50$ and a similar test is run as described in Section 4.2.1. The smoother window is kept constant using $S = 25$. The RMSE of the analysis is reported in Figure 4.6. The performance of the MLES is severely diminished when a small ensemble size is used. The performance is found to be relatively stable after approximately N_e is chosen to be 15 or greater. However, more members than 20 are not necessary for this case according to the sensitivity study.

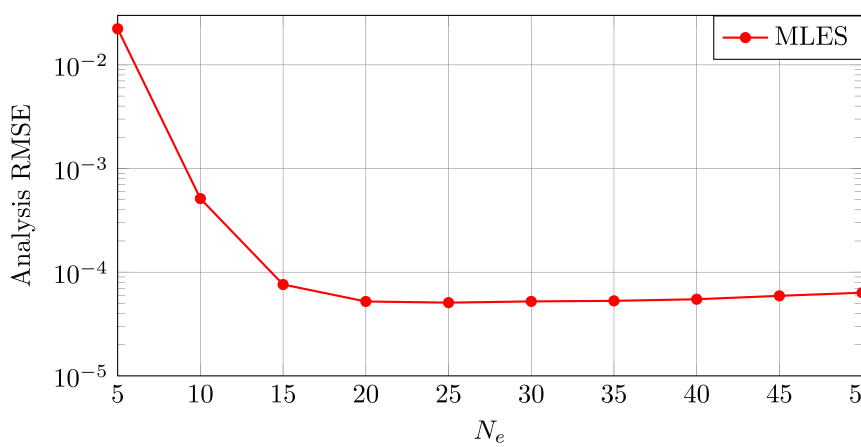

Figure 4.6: MLES performance as a function of N_e using a constant smoother window length of $S = 25$.

4.2.3 Number of Observations

The MLES is now used to directly compare results with the MLEF. The current test cases are performed by varying the number of available observations in the domain (N_O) while keeping S constant for each case. Spatially, these observations are evenly spaced. For the MLES, only one smoother window (and therefore only one analysis update) is used. Once the optimal analysis state has been found, the control state of the MLES is forecasted to a temporal location equal to twice the MLES smoother window length and the RMSE is calculated. In order to directly compare with the MLES, the control state of the MLEF is updated using S DA cycles. At the last point of the MLEF update, the MLEF control state is forecasted to the same ending point of the MLES and the RMSE is calculated. This means that the time between the updated analysis location for the MLES and the end of the simulation is equivalent to two smoother window lengths whereas the last MLEF analysis update is exactly one smoother window length. The process is graphically shown in Figure 4.7. The size of the ensemble is set to $N_e = 20$ members and the smoother window is set to be $S = 25$. The other simulation parameters are the same as presented in Section 4.2.1. The result of the averaged RMSE over 100 independent runs for each N_O is shown in Figure 4.8. The RMSE for both algorithms decreases with more available observations which is to be expected. The error reduction for the MLES is slightly better for all cases. This is promising and interesting as both DA algorithms have been provided with the same information, while the main difference is that the MLES has a longer time between the location of the RMSE calculation and the initial analysis. This will be an area of emphasis for future studies in order to better understand the impact of the observations on the performance.

4.2.4 State Quantities and RMSE

Three state quantities are presented to compare the improvement of predictability for both the MLES and MLEF. Using $N_O = 64$, the state variable at spatial domain locations of 1, 65, and 221 for the MLES, MLEF, free run, and truth run are shown in Figures 4.9-4.11. Also shown are the observed quantities at these locations as well. The observation interval is chosen to be 200

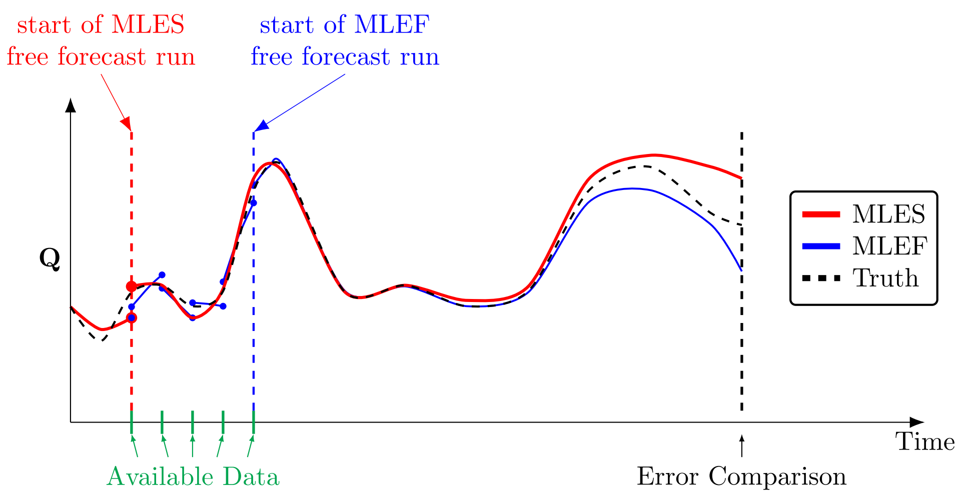

Figure 4.7: Graphical description of the error comparison with the MLES and MLEF. The MLES is updated at the vertical dashed red line and freely run to the vertical black dashed line. The MLEF is updated at each of the green hashes which represent available observations. The last update for the MLEF is located at the vertical blue dashed line and run to the same vertical black dashed line. At the vertical black dashed line, the error of the MLES and MLEF is calculated with respect to the truth and compared. This value is saved and the mean of these values is taken over 100 simulations. In the figure, the y-axis, $\mathbf{Q}$, represents just one state value in the entire spatial domain.

time steps in this case. Each DA algorithm correctly matches the truth state beginning at the first analysis cycle located at time step 0.1×10^4. Deviation for both runs can be clearly distinguished at time step 2×10^4. For each spatial location presented, the MLES does a slightly better job at

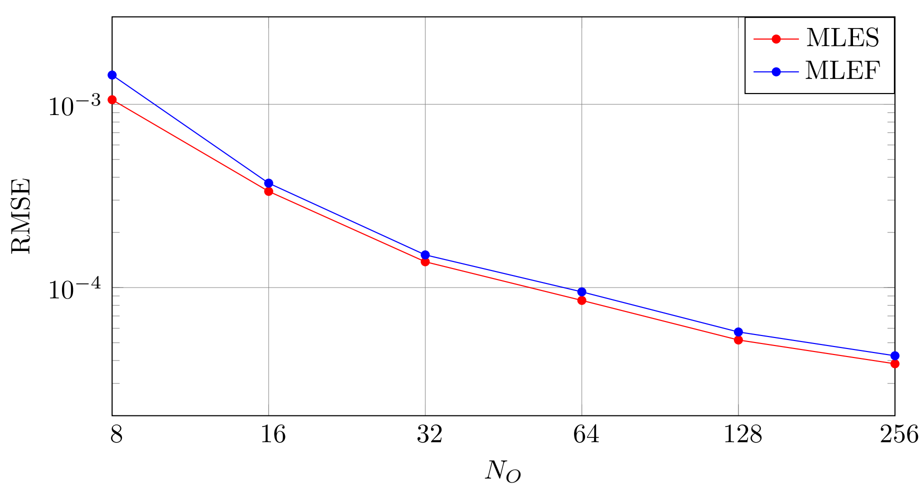

Figure 4.8: Performance comparison between the MLES and MLEF using a smaller amount of evenly spaced observations in the domain. The RMSE is taken at the end of a time window length equal to two smoother window lengths after the first initial MLES analysis update.

predicting the truth for a slightly longer time than the MLEF. It is worth noting that at time step 1.8×10^4, the MLES does slightly diverge from the truth but is still a good approximation of the truth.

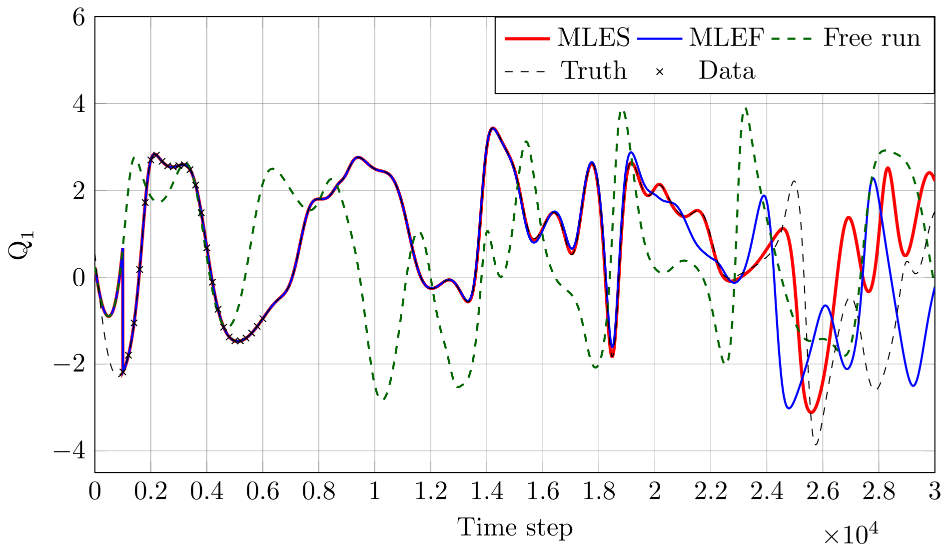

Figure 4.9: The model state and observation data at the first spatial location.

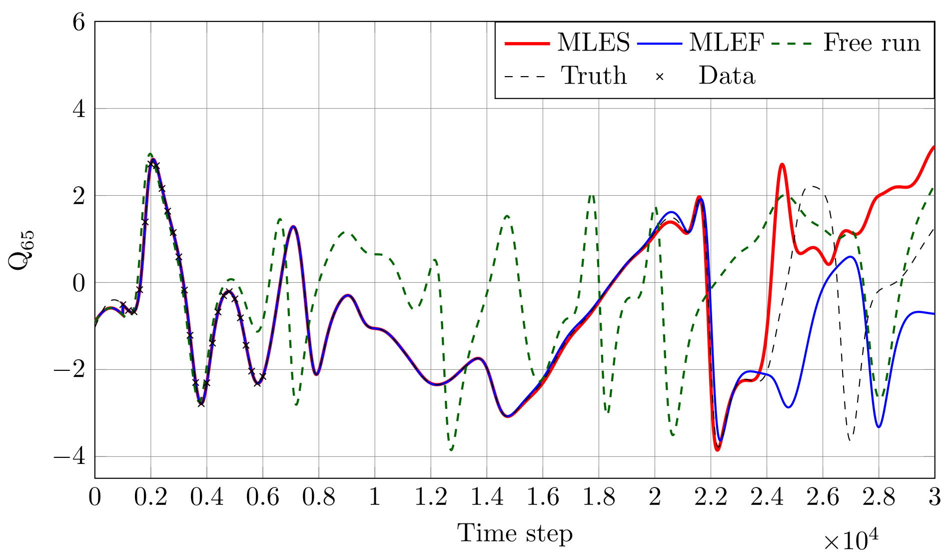

Figure 4.10: The model state and observation data at the 65^{th} spatial location.

The RMSE value is also given for the current simulation and is shown in Figure 4.12. The MLEF updates can clearly be seen from the vertical drops after the initial analysis state update. The plot shows good initial convergence to the truth and accurately depicts the chaotic nature of the KS equation as both DA algorithms eventually diverge from the truth. Note that this is a single run and previous results have been collected from an average of these runs.

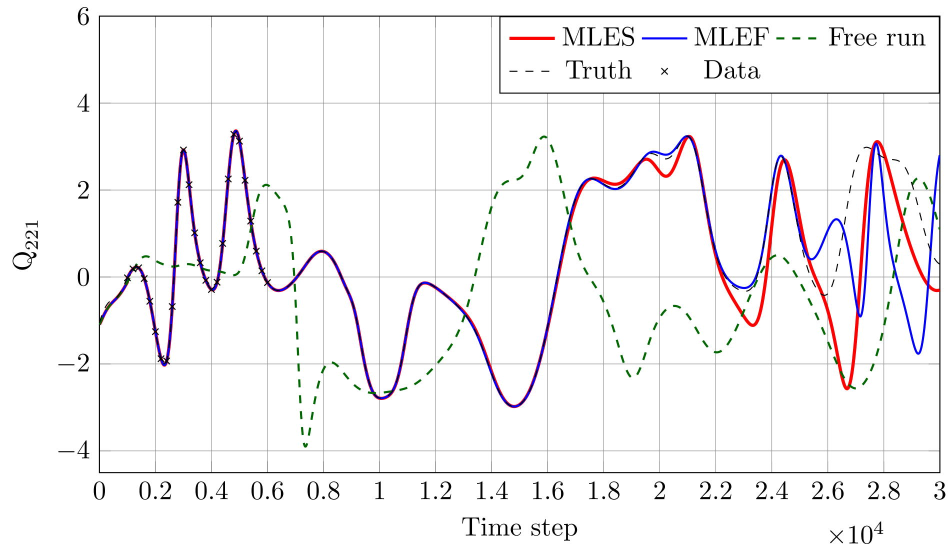

Figure 4.11: The model state and observation data at the 221st spatial location.

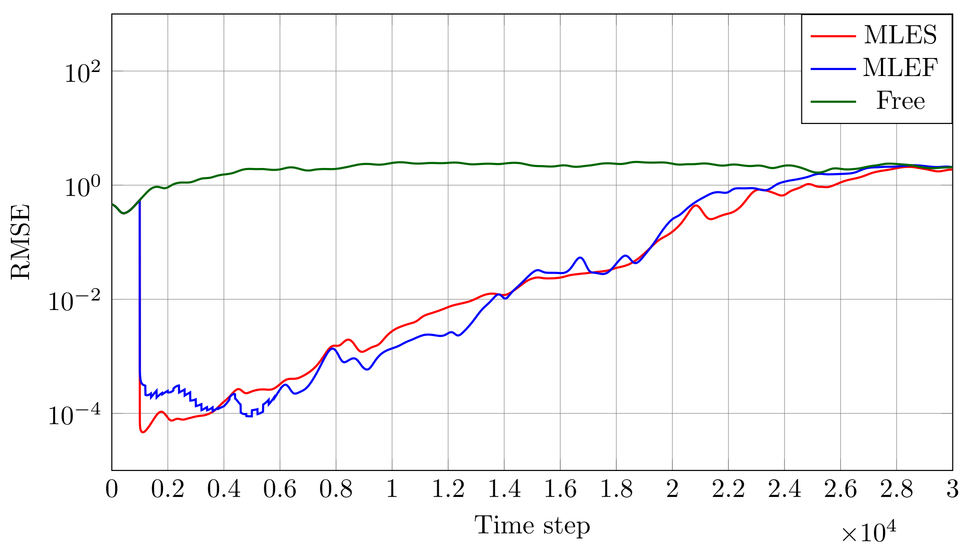

Figure 4.12: The RMSE values for the MLES, MLEF, and free run.

4.2.5 Model Parameter Estimation

The estimation of model parameters of the KS equation, a, b, and c, are now included in the DA process and therefore added to the control variable $\mathbf{Q}$. The initial uncertainty of the parameters for each member in the ensemble is chosen by a random Gaussian PDF with mean 0.50 and variance of 0.05. For reference, the truth of these parameters is known as $a = b = c = 1$. 10 DA cycles for the MLES is used and the corresponding amount of DA cycles is used for MLEF. The initial "burn-in" period is reduced to 2 time units. The rest of the case setup is similar to that in Section 4.2.1. Note that the model parameters are not observed as this is often the case. The statistical average of the mean of the final analysis update of the model parameters for the MLES and MLEF after 100 independent cases is shown in Table 4.2. All three model parameters for both the MLES and MLEF to converge to approximately the true values.

Table 4.2: The estimated model parameters for the MLES and MLEF by taking the average of the last analysis model parameters over 100 different cases. The mean of the initialization of the model parameters is set to be 50% less than that of the truth.

Model Param.	Truth	MLES	MLEF
a	1.00	1.0003	1.0010
b	1.00	1.0013	1.0009
c	1.00	1.0001	1.0012

Additionally, the plots of the model parameter update over time is also shown in Figures 4.13-4.15 for a single run. Although the simulations over the entire solution time are not shown for the sake of space, both MLES and MLEF converge to the true model parameters quickly. Both DA algorithms show good prediction of the true model parameters even with a poor initial uncertainty of the truth. The MLES does better for a and c while slightly less for b when compared to the MLEF. More complex chaotic systems will be investigated to see whether the MLES consistently does better than the MLEF in general.

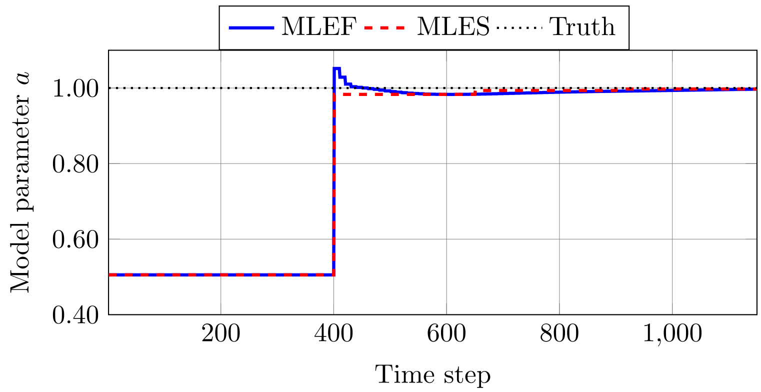

Figure 4.13: Model parameter a convergence.

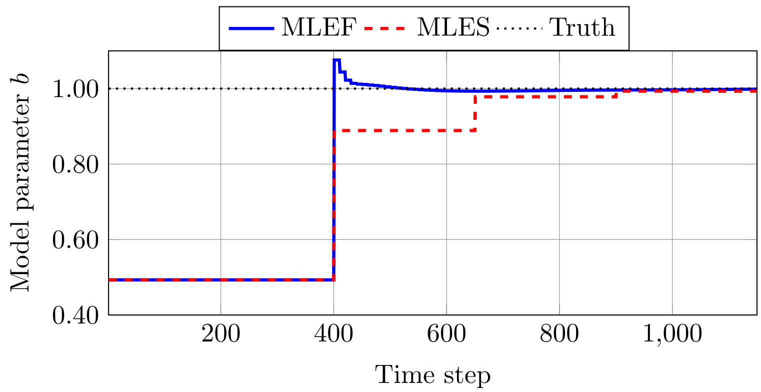

Figure 4.14: Model parameter b convergence.

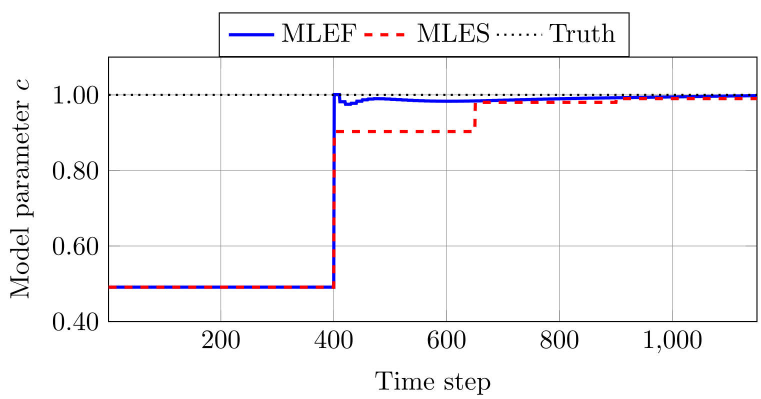

Figure 4.15: Model parameter c convergence.

4.2.6 Averaged Observations

In fluid dynamics problems that involve chaotic turbulence, rather than providing instantaneous data, the time-averaged flow data are desirable. Thus, it is of interest to use time-averaged data. To synthesize the data, a time-averaged true value is used in the observation operator, but a random perturbation η is applied to the averaged value as "measurement error." The acquisition of this data is shown in Figure 4.16. Apart from the averaged observations, the case setup is identical to that in Section 4.2.5 where the statistical average of the model parameters are found after running 100 cases. Table 4.3 presents the model parameter estimates by the MLES and MLEF. All three model parameters for both the MLES and MLEF converge to the true values and the MLES performs slightly better for a and c than the MLEF.

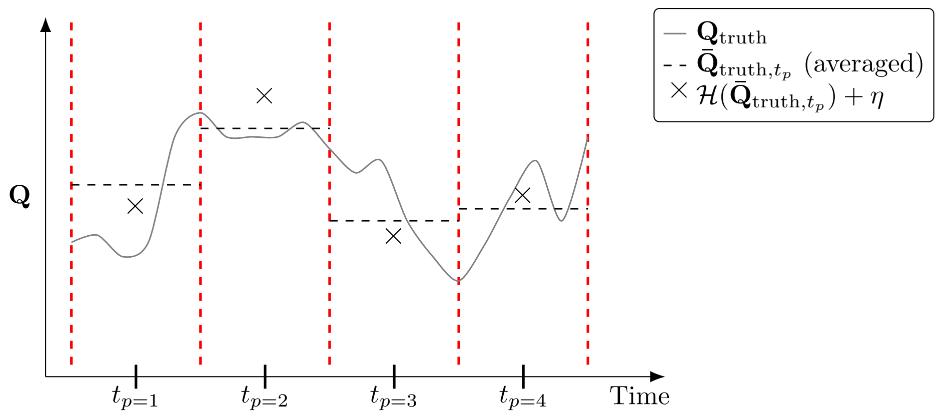

Figure 4.16: To gather the averaged data, an average of the true values is taken within an interval and then the observation operator is applied and perturbed using random variance. The interval is shown as the vertical red dashed lines and the black horizontal dashed lines are the averaged true values. The black crosses represent the data obtained after applying the observation operator $\mathcal{H}(\cdot)$ and adding random variance, η. The y-axis, $\mathbf{Q}$, represents just one state value in the entire spatial domain.

Table 4.3: The estimated model parameters for the MLES and MLEF using the averaged observations method.

Model Param.	Truth	MLES	MLEF
a	1.00	0.9997	1.0004
b	1.00	1.0011	1.0006
c	1.00	0.9996	1.0006

The plots of the model parameter update over time are also shown in Figures 4.17-4.19 for a single run. Note only 3 MLES DA cycles (and the equivalent number of MLEF DA cycles) of the simulation are shown as both MLES and MLEF converge to the true model parameters quickly.

Both DA algorithms show good prediction of the true model parameters even with a poor initial uncertainty of the truth. Notice that the MLES performs slightly better in this verification process.

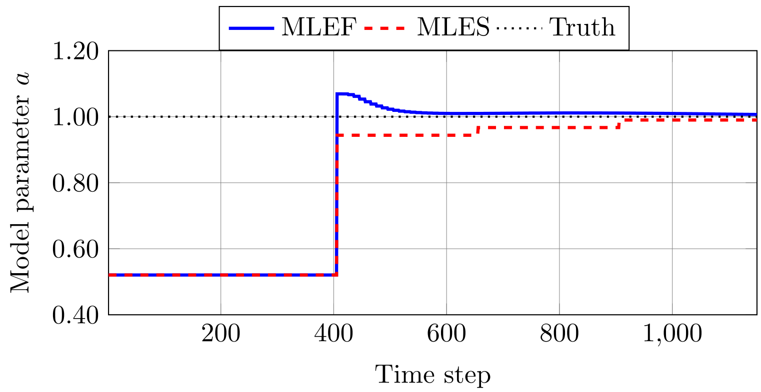

Figure 4.17: Model parameter a convergence for the time-averaged synthesized observations.

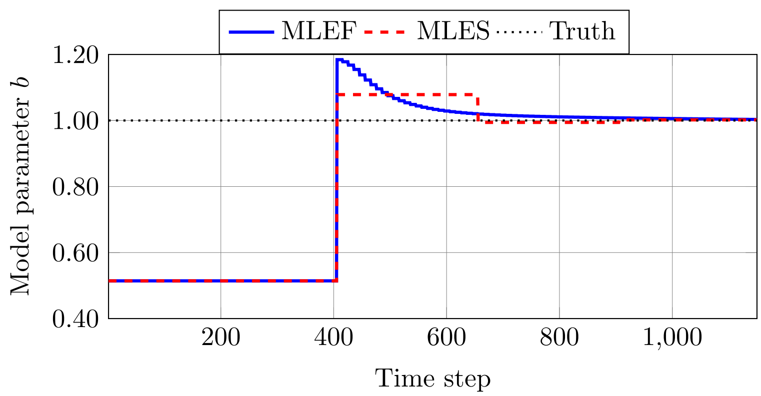

Figure 4.18: Model parameter b convergence for the time-averaged synthesized observations.

4.3 Verification and Validation: MLES+MGinT

The current section gives details the verification of MLES+MGinT using similar tests as seen in the verification of the MLES.

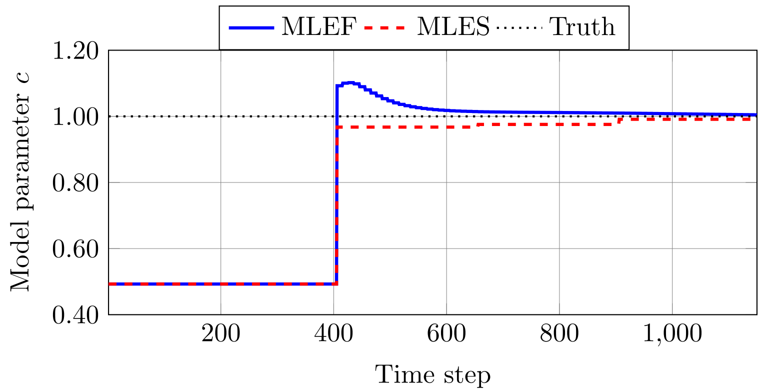

Figure 4.19: Model parameter c convergence for the time-averaged synthesized observations.

4.3.1 Performance metrics

To compare the performance of the MLES and MLES+MGinT, we compute a temporal mean of performance metrics over many analysis times, denoted by k. The first metric is the root-mean-square error (RMSE), which is given as

$$\mathrm{RMSE}_k = \sqrt{\frac{\sum_{m=1}^{M} \left(\mathbf{Q}_{k,\text{true}} - \hat{\mathbf{Q}}_k^a\right)_m^2}{M}}, \tag{4.7}$$

and indicates the deviation from the truth.

The second metric is the ensemble spread-skill relationship. It is used to assess the accuracy of the linear-Gaussian approximation in DA. It involves comparing the RMSE to the ensemble spread [55]. If the ensemble RMSE and the ensemble spread are of approximately the same order, it indicates a satisfactory spread-skill relationship. Large deviations indicate inadequacies in the linear-Gaussian approximation. This relationship gives important insights into the overall performance in highly nonlinear applications. The ensemble spread is defined as follows:

$$\mathrm{spread}_k = \sqrt{\frac{1}{N_e - 1} \frac{\sum_{m=1}^{N_e} \left(\mathbf{P}_{k,a,m}^{\mathrm{T}/2} \mathbf{P}_{k,a,m}^{1/2}\right)}{M}}, \tag{4.8}$$

where $\mathbf{P}_{k,a,m}^{\mathrm{T}/2}$ is the analysis error covariance matrix at time index k and column index m.

The third metric is the wall-clock time to minimize the cost function. To evaluate the wall-clock time, the MLES+MGinT and MLES algorithms are implemented and run on the same computational setup. The execution time, given in seconds, for each algorithm is measured and compared. This comparison measures the computational efficiency of the selected method. A shorter wall-clock time for MLES+MGinT indicates faster execution and potentially better scalability, making it advantageous for large-scale applications.

Lastly, the comparison of the number of iterations required for the minimization process is performed between the MLES+MGinT and MLES algorithms. Specifically, the focus is on evaluating the number of iterations needed on the finest time-grid level for MLES+MGinT. Evaluating the iterations on the finest time-grid level is crucial because, in general, computations on the finest grids are exponentially more resource-intensive. As a result, understanding the iteration count on this level provides valuable insights into the efficiency and computational cost of the MLES+MGinT algorithm in comparison to the standard MLES. It's important to note that unlike MLES, the MLES+MGinT algorithm requires iterations on every time level.

Minimization

In this appendix, the convergence criteria (i.e. when to stop the minimization iterations) is given by Eq. (2.25), and e, or $error$, is $1e-04$.

4.3.2 Parametric studies

For each of the following numerical experiments, we modify parameters, assess the performance metrics, and present the results as a heat map in the parameter space. The parameters that are adjusted are the lag, shift, observation nonlinearity, and the model's time step size on the finest level. These are chosen due to their influence on the level of nonlinearity in the DA procedure. The significance of adjusting the nonlinearity is that it directly relates to the understanding of complex CFD problems. As CFD problems inherently exhibit a high degree of nonlinearity, characterizing the effects of nonlinearity through parameter adjustment becomes crucial when applying DA to such problems.

Verification

First, to establish our confidence in the MLES+MGinT algorithm and its implementation, the solution is compared to the truth and that of the MLES. Figure 4.20 shows four lines: the results of using the MLES+MGinT, the MLES, the truth, and a free run. The MLES+MGinT, the MLES, and the free run are all initialized from the same initial state. This state is slightly perturbed from the truth's initial starting point. The forward model is the same for every case. The expectation, under ideal DA configurations (i.e. low observation variance, relatively large ensemble size, etc.), show both the MLES+MGinT and MLES methods show good adherence to the truth. By contrast, the free run, found by using the same initial conditions and applying the numerical model without DA, diverges from the truth almost immediately. The observed behavior in Figure 4.20 supports the effectiveness of the MLES+MGinT algorithm and its implementation. This particular case was taken from the parametric study below, using $L = 8$ and $S = 4$.

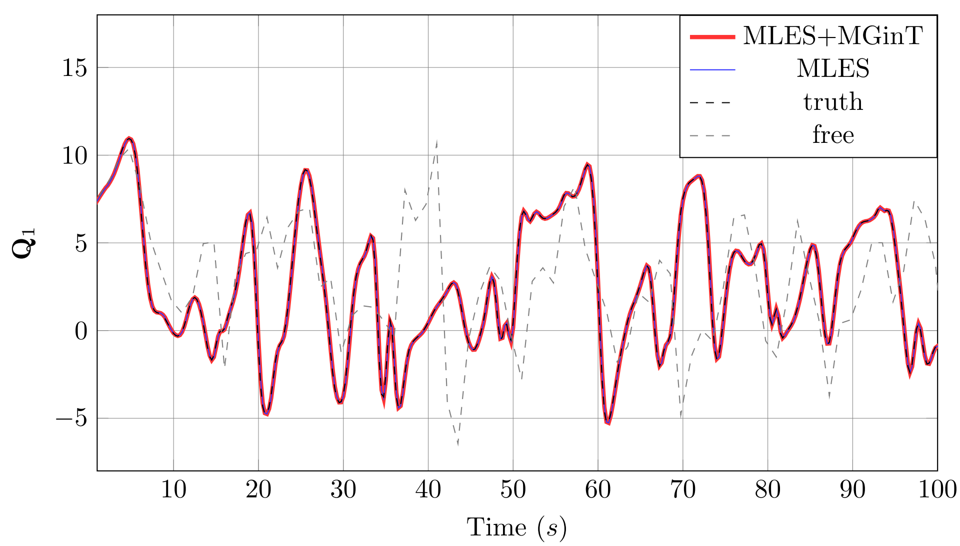

Figure 4.20: Plot of the first state variable in space, $\mathbf{Q}_1$ showcasing the effect that DA has on predicting the truth when compared to a free run. The red line is the MLES+MGinT and the blue line is the MLES. Both methods closely approximate the truth of the dotted black line. The gray dotted line is the free run with no DA applied, and quick divergence from the truth.

Lag vs. shift

The two parameters that are adjusted in this section are the lag and shift. Their numerical effect on the analysis procedure is explained in Chapter 2, but their effect on the nonlinearity of the system is now explained. Increasing the lag means the length of the DA window is increased. This implies that calculating the cost function requires a longer integration of the forward model. Because the forward model is nonlinear, this results in increased nonlinearity in the analysis procedure. For chaotic problems, this eventually deteriorates the quality of the posterior estimate and the solution becomes unstable. When this happens, the reliable estimation of the truth is lost and the DA solution is considered to be diverged. Additionally, observations that are temporally further from the start of the DA window may contribute to numerous local minima. These additional minima can significantly impact the optimization's performance.

Smoothers with a shift greater than one offer improved computational efficiency because it reduces the necessary number of cycles by a factor of S. This is because using a larger shift value reduces the number of cycles required to analyze a time series of observations with sequential DA windows. However, a greater S value means there is longer free forecasts between analysis times which introduces more nonlinearity between posterior estimates.

Choosing L and S requires a balance between the length of the DA window and the number of observations to incorporate in the analysis. The objective for this particular parametric study is to compare these aspects while also comparing the computational efficiency of the MLES and the MLES+MGinT. The configuration for this numerical experiment is given in Table 4.4.

Figures 4.21-4.24 depict the results for RMSE, spread, wall-clock analysis times, and the number of minimization iterations. Each figure describes the behavior of a specific performance metric in the parameter space of a specific performance metric, with the left subfigure representing the MLES and the right one representing the MLES+MGinT method. The x-axis represents the S values, and the y-axis represents the L values. Note that if $S > L$, the data used for each DA window overlaps and thus are not shown.

Table 4.4: The forward model and DA configuration for the lag vs. shift parametric study.

Forward model	**Configuration**
Number of variables	40 ($M = 40$)
Time step size	$\Delta t = 0.01\,s$
DA	**Configuration**
DA Method	MLES and MLES+MGinT
Size of the ensemble	$21 + 1$
Control vector	$\mathbf{Q} = [q_1, q_2, \ldots, q_{40}]^{\mathrm{T}}$
Observation operator	$\mathcal{H}(\mathbf{Q}) = \mathbf{Q}$
Observations	Synthetic data are generated with a simulated truth for $q_1, q_2, \ldots, q_{40}$
Observation error	N(0,$\epsilon_{\mathbf{O}}$), $\epsilon_{\mathbf{O}} = 1.0$
Number of analyses	100
Observation frequency	$\Delta t_{\mathbf{O}} = 0.05$
Time multigrid levels	3
Multigrid Δt	$\Delta t_{\text{coarsest}} = \Delta t_{\mathbf{O}}$, $\Delta t_{\text{finest}} = \Delta t$ $\Delta t_{\text{intermediate}} = (\Delta t_{\text{coarsest}} - \Delta t_{\text{finest}})/2$
Lag	$L = 1, 4, 8, 16, 24, 32, 40, 48$
Shift	$S = 1, 4, 8, 16, 24, 32, 40, 48$

To obtain the values in the parameter space, a total of 100 analysis points are averaged for each combination of L and S, following an initial burn-in period to reduce sensitivity to initial conditions. The corresponding values are represented by colors in the figures, as indicated in the legends. Blue indicates lower values, while red indicates higher values for their respective performance metrics. Note that the legend in Figure 4.22 has the same scaling as in Figure 4.21 for direct comparison. Also, in Figure 4.24, the number of minimization iterations for the MLES+ MGinT is given for the fine-time-grid only and not coarser-time-grids.

According to Figure 4.21, the quality of the analysis for both the MLES and MLES+MGinT deteriorates with a lag greater than 32. This is consistent with the increased nonlinearity introduced into the system and indicates that $L > 32$ is the limit of reliability in this particular numerical setup. The spread for both methods are similar and, as expected, we see a large deviation in magnitudes between the spread and RMSE values for the aforementioned diverged cases with $L > 32$.

Regarding the wall-clock times needed to achieve the optimal estimate, there is a slight improvement observed for the MLES + MGinT compared to the MLES at lower lags and shifts. Nevertheless, the most notable improvement is evident in the number of required iterations for minimization, where the MLES + MGinT approach consistently outperforms the MLES method. This improvement is particularly notable in scenarios where the performance of the MLES is not severely degraded.

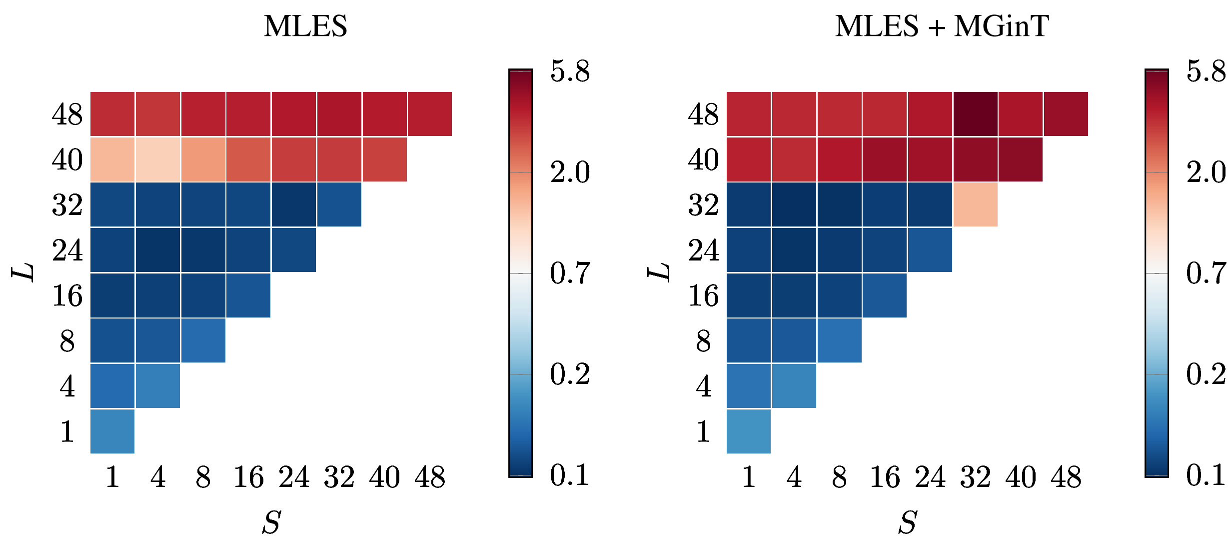

Figure 4.21: Distribution of the RMSE in the L-S parameter space.

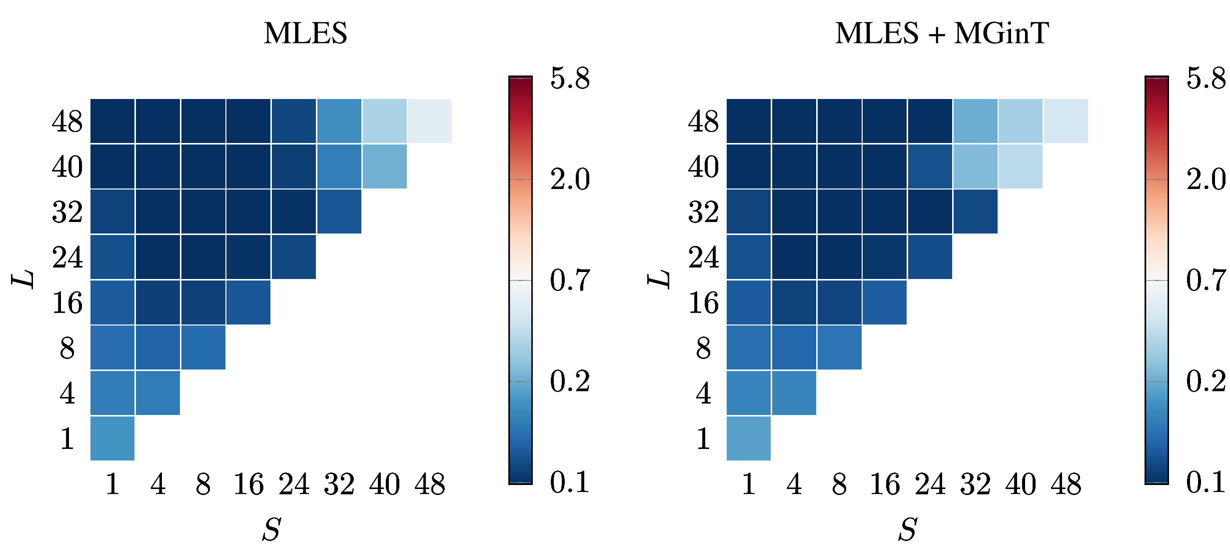

Figure 4.22: Distribution of the ensemble spread in the L-S parameter space.

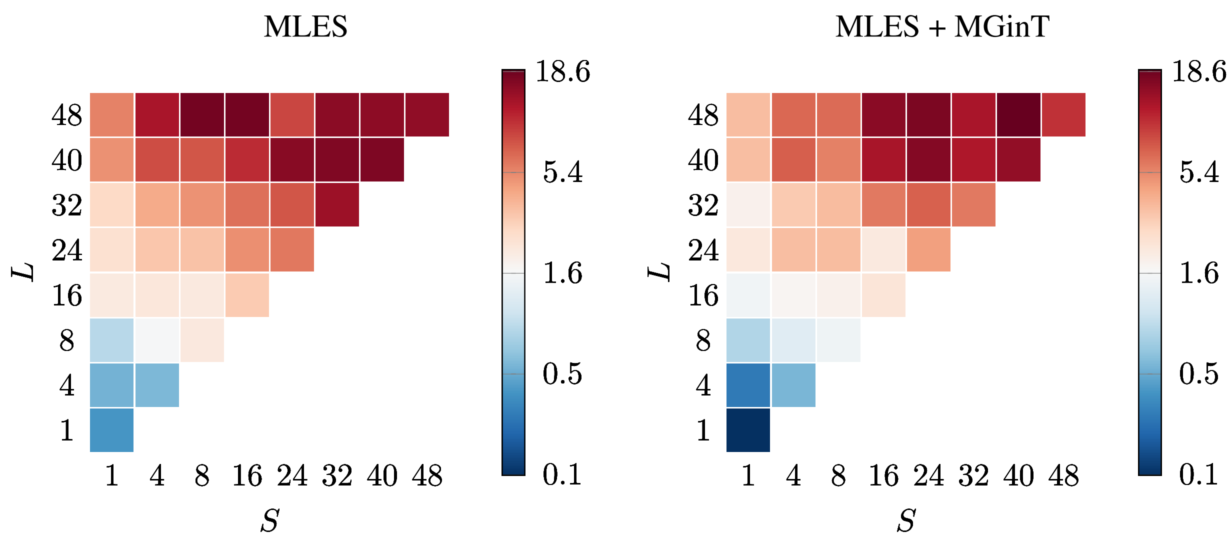

Figure 4.23: Distribution of the wall-clock times in the L-S parameter space.

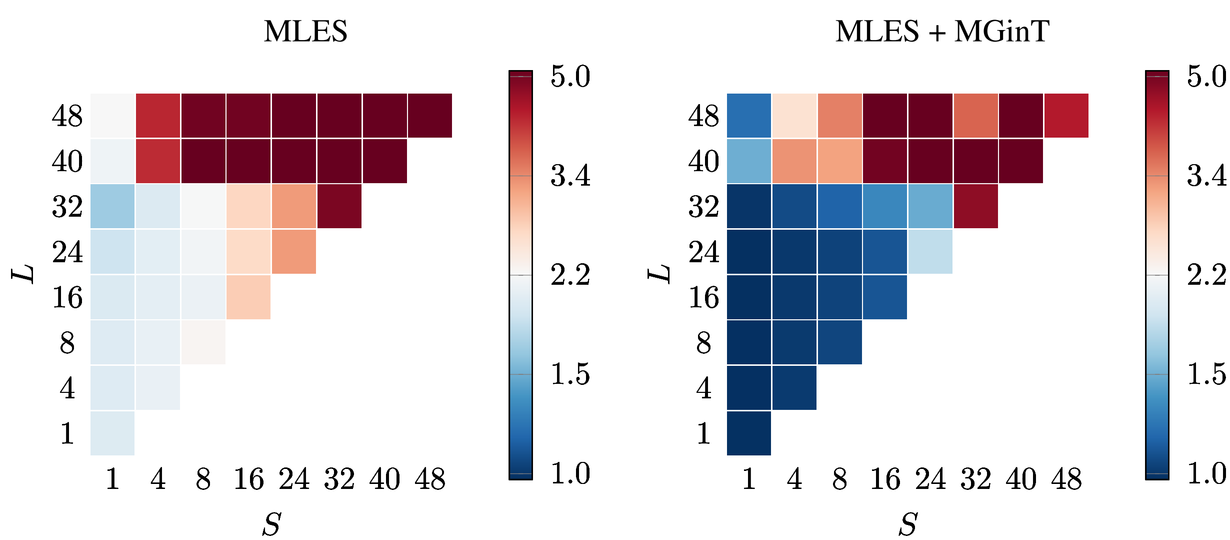

Figure 4.24: Distribution of the minimization iterations required in the L-S parameter space.

Lag vs. nonlinear observations

In this parametric study, the two parameters that are modified are the lag and the nonlinearity of the observation operator. Observations are generated by

$$\mathbf{O}_k = \mathcal{H}\left(\hat{\mathbf{Q}}_{k,\text{true}}\right) + \boldsymbol{\eta}\,, \tag{4.9}$$

where k is the time index, $\boldsymbol{\eta}$ is a random perturbation taken from a normal distribution of a chosen variance, $\epsilon_{\mathbf{O}}$, and mean of 0. $\mathcal{H}(\cdot)$ is given as

$$\mathcal{H}(\mathbf{Q}_k) = \frac{\mathbf{Q}}{2} \circ \left[1 + \left(\frac{|\mathbf{Q}|}{10} \right)^{\gamma - 1} \right], \tag{4.10}$$

where γ is an integer number and is used to control the nonlinearity of the mapping, and $\circ$ is a Shur product. Eq. (4.10) is the same as in Asch et al. [37] and is chosen for the purpose of reproducibility. A higher value of γ corresponds to a more nonlinear operator. The details of the observation operator can also be found in Table 4.5.

Table 4.5: The forward model and DA configuration for the lag vs. γ parametric study. All parameters not shown can be found in Table 4.4.

DA	**Configuration**
Observation operator	$\mathcal{H}(\mathbf{Q}) = \frac{\mathbf{Q}}{2} \left[1 + \left(\frac{\lvert\mathbf{Q}\rvert}{10} \right)^{\gamma-1} \right], \gamma = 1, \ldots, 10$
Lag	$L = 1, 4, 8, 16, 24, 32, 40, 48$
Shift	$S = 1$

The results are shown in Figures 4.25-4.28 and follow the same layout as the figures in Figure 4.3.2. Consistent with the findings from the previous section, the RMSE values show a significant increase when $L > 32$. However, it is worth noting that the introduction of the nonlinear observation operator had a more pronounced impact on the MLES+MGinT compared to the MLES. In particular, the right plot in Figure 4.25, representing the performance of MLES+MGinT, demonstrates deteriorating RMSE values starting at $L = 24$ and $\gamma = 5$. This degradation occurs earlier compared to the MLES method. The ensemble spread values remain relatively similar between the two methods.

Despite the slight increase in RMSE values, the MLES+MGinT method shows an improvement in terms of wall-clock times and the number of iterations required on the fine-time-grid level. In cases where the MLES + MGinT method performs well in terms of RMSE, only about 1 iteration is needed, whereas the MLES method typically requires 2 iterations.

In summary, while the MLES+MGinT method experiences a degradation in RMSE values compared to the MLES method, it still demonstrates advantages in terms of computational efficiency, reflected in reduced wall-clock times and iterations required for convergence on the fine-time-grid level.

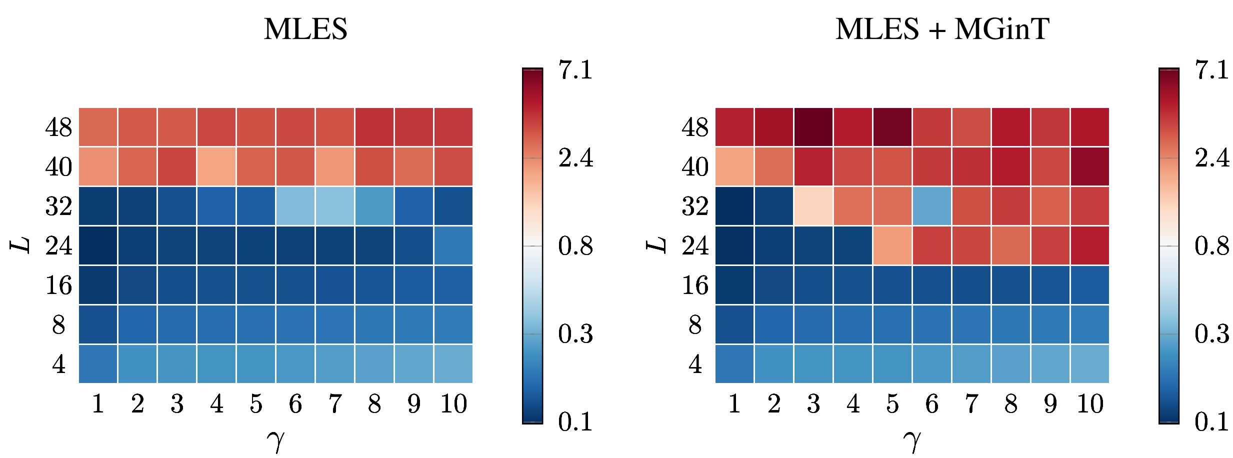

Figure 4.25: Distribution of the RMSE in the L-γ parameter space.

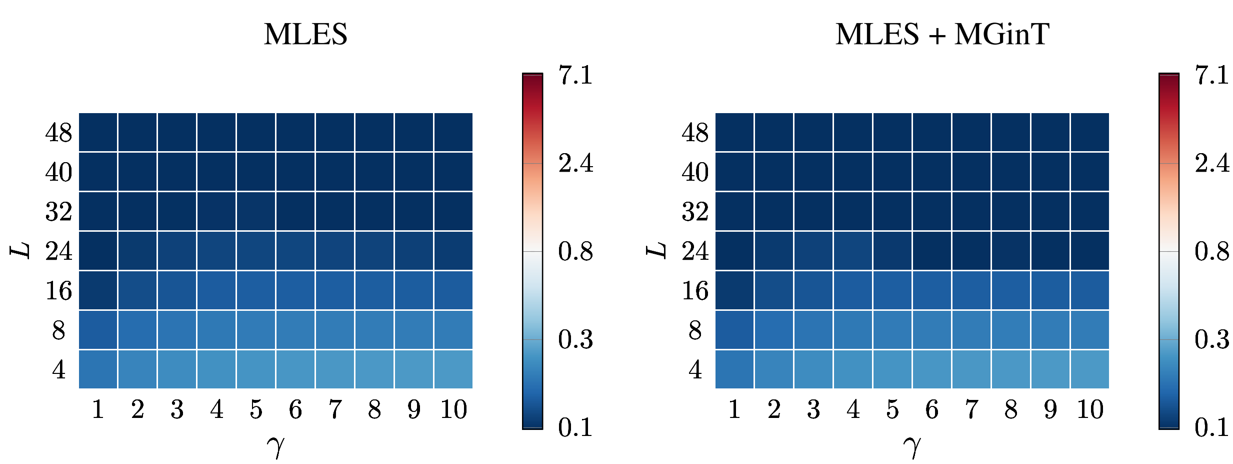

Figure 4.26: Distribution of the ensemble spread in the L-γ parameter space.

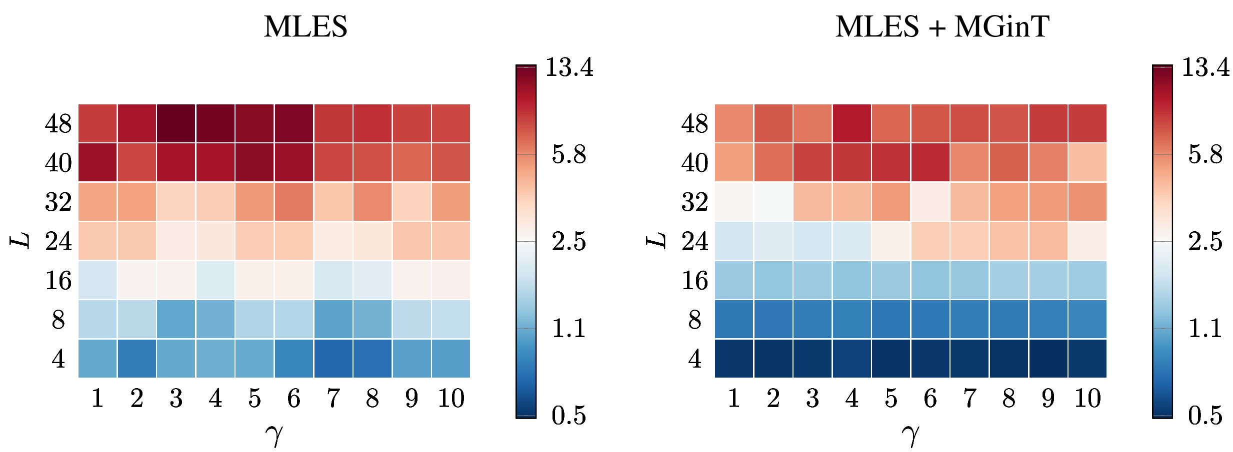

Figure 4.27: Distribution of the wall-clock times in the L-γ parameter space.

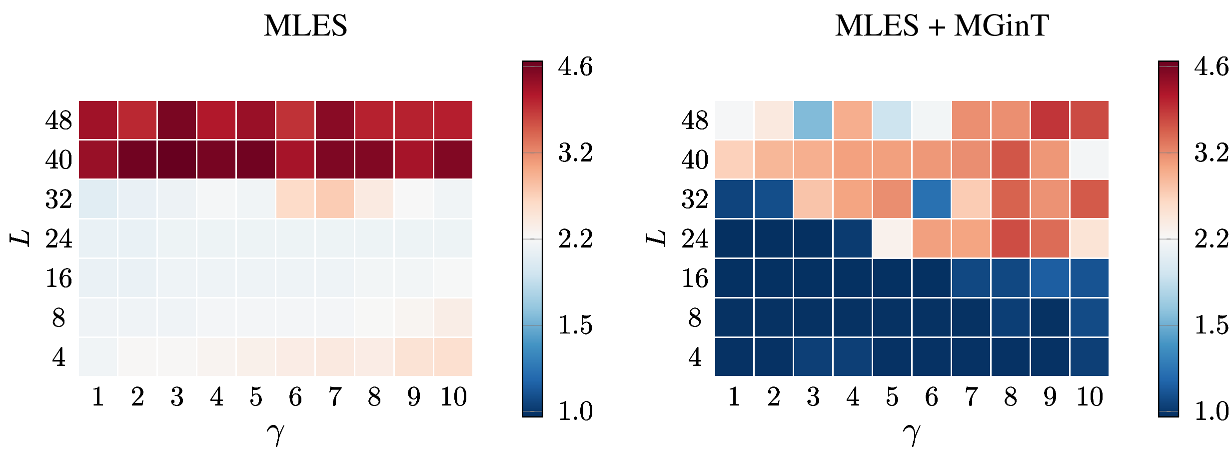

Figure 4.28: Distribution of the minimization iterations required in the L-γ parameter space.

Lag vs. Δt

In this section, we show the results when the nonlinear forecasting errors are the primary source of nonlinearity in the analysis. This is accomplished by adjusting the lag and the time step size of the forward model. A greater time step size equates to longer times between analyses which has a similar affect of increasing the lag wherein more nonlinearity is introduced. The case setup is shown in Table 4.6.

Table 4.6: The forward model and DA configuration for the lag vs. Δt parametric study. All parameters not shown can be found in Table 4.4.

Forward model	**Configuration**
Time step size	$\Delta t = 0.1 \times 10^{-3}, 0.25 \times 10^{-3}, 0.5 \times 10^{-3},$ $0.75 \times 10^{-3}, 1.0 \times 10^{-3}$ s
DA	**Configuration**
Lag	$L = 4, 8, 16, 24, 32, 40, 48$
Shift	$S = 1$

The results are shown in Figures 4.29-4.32 and share the same traits as the figures in Section 4.3.2. The RMSE and spread plots are nearly indistinguishable between the MLES and MLES + MGinT. Slightly faster wall-clock times are seen for the MLES + MGinT and there is a definitive reduction in iterations required on the fine-time-grid for the MLES + MGinT when compared to the MLES.

Each parametric study shares similar trends when comparing the RMSE and spread values between the MLES and MLES + MGinT methods. However, it is important to acknowledge that these studies are conducted using a simplified and computationally cheap model, which may lessen the effect of the differences in wall-clock times between MLES+MGinT and MLES. Nevertheless, it is expected that when applied to complex CFD problems, the impact of the wall-clock time will be more significant. This is especially relevant when taking into account the minimization iterations required, where the MLES + MGinT consistently outperforms the MLES in the majority

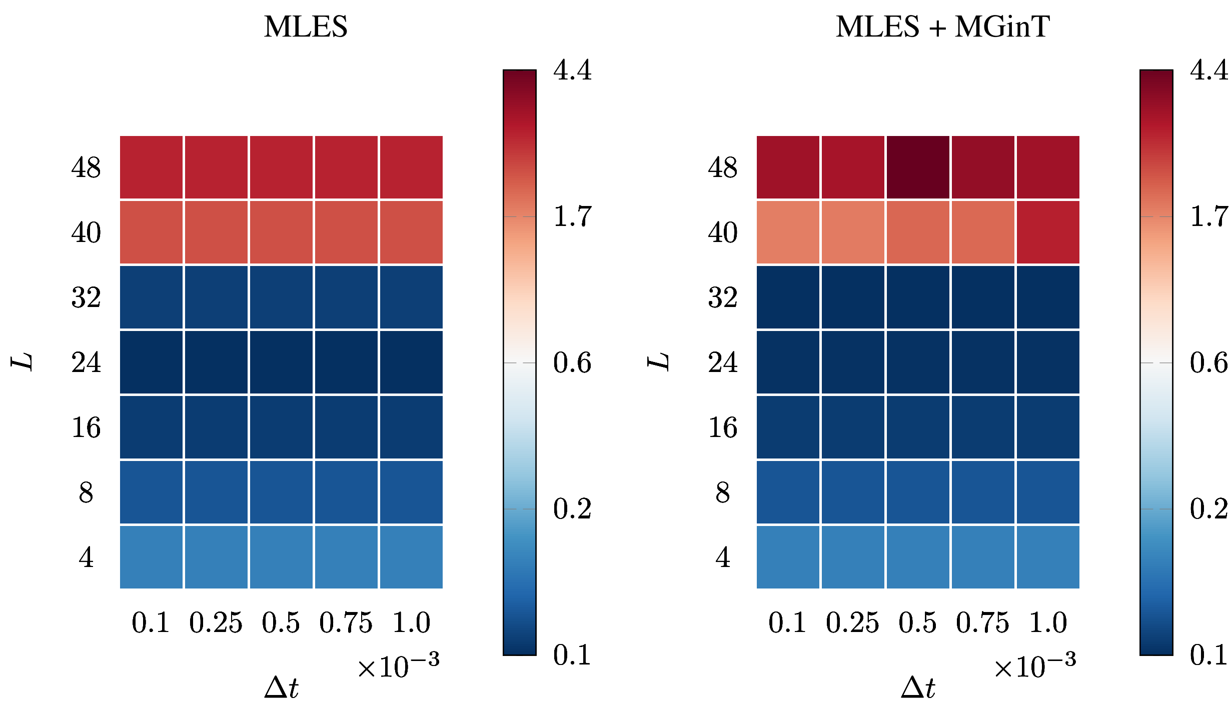

Figure 4.29: Distribution of the RMSE in the L-Δt parameter space.

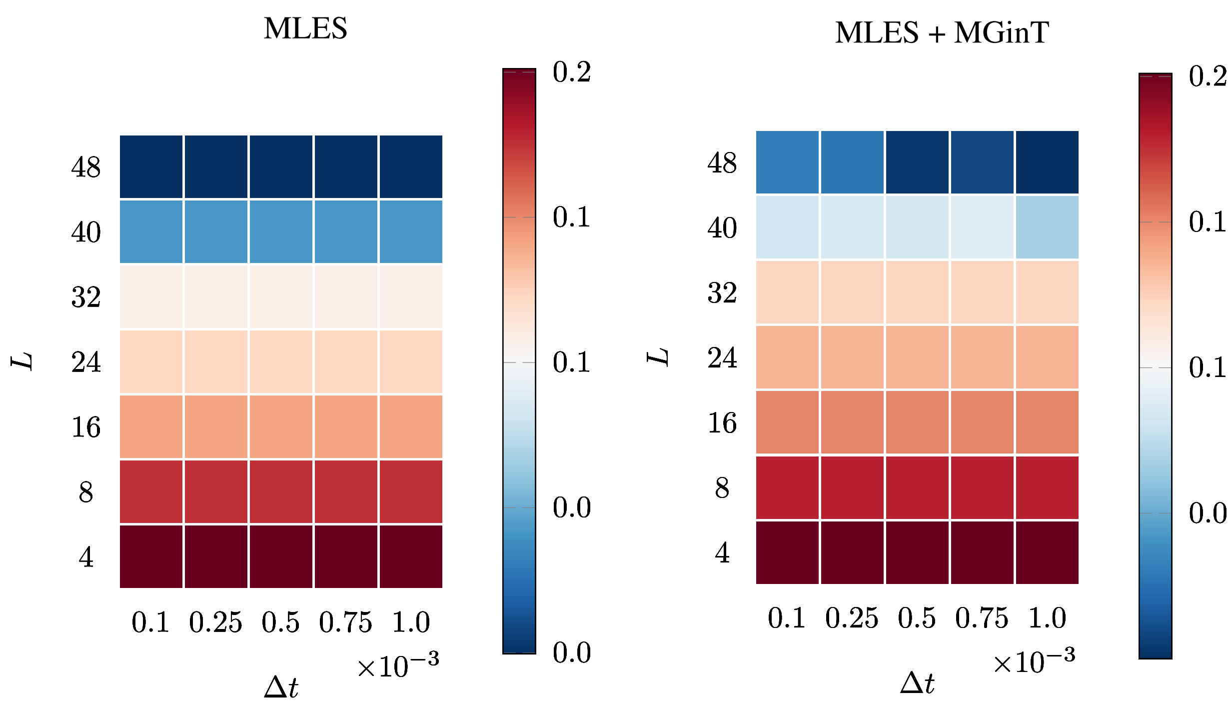

Figure 4.30: Distribution of the ensemble spread in the L-Δt parameter space.

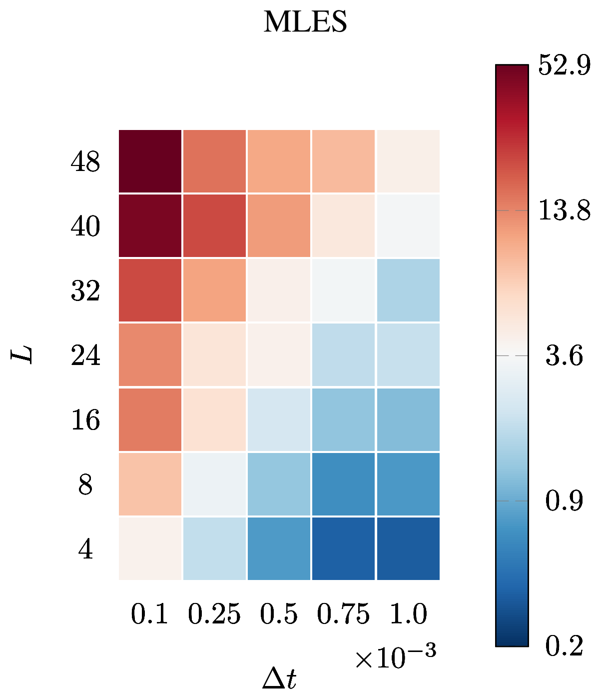

Figure 4.31: Distribution of the wall-clock times in the L-Δt parameter space.

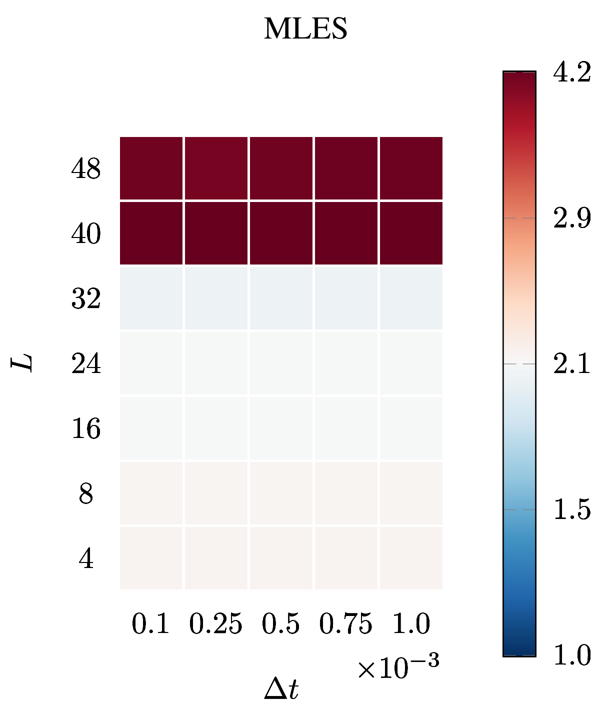

Figure 4.32: Distribution of the minimization iterations required in the L-Δt parameter space.

of the numerical experiments. This finding is particularly promising because avoiding the use of the fine-time-grid model will have a clear positive impact on the computational resources required.

In summary, while the comparisons between the MLES and MLES+MGinT in terms of RMSE, spread, and wall-clock times may not show drastic differences in the current simplified model, the MLES + MGinT demonstrates a clear advantage in terms of minimization iterations and provides potential for efficient utilization of computational resources. This positive impact is expected to become more prominent when applying more complex fluid dynamics problems.

The verification and validation process for the MLES proves its reliability and accuracy, reinforcing its suitability for CFD applications. With this, the MLES will serve as a tool for DA-enhanced CFD predictions of more complex fluid dynamics problems.

Chapter 5

Experiment Results and Discussion

5.1 1D Shock Tube

The MLES is applied to the Sod's [56] 1D shock tube problem. This problem has the complex physics of shock wave propagation and fluid interactions and is suitable for testing DA methods because of its simple configuration. Houba et al. [57] presented an application to flows with discontinuities by using conventional ensemble DA methods, however this is the first time in the literature that an ensemble-variational smoother method has been performed on the 1D shock tube. Beyond academic research, the shock tube problem's applicability extends to engineering scenarios, where shock-capturing numerical methods can be employed alongside DA to improve the estimate of the shock location. By applying the MLES to the shock tube problem, we gain insight into the strengths and limitations of the MLES, enhancing its broader applicability to more intricate fluid dynamics simulations.

5.1.1 Problem Setup

In this particular setup, we investigate and compare the performance of the MLES and the MLEF in the context of Sods' 1D shock tube problem. The shock tube problem is initialized by specific initial parameters, which represent the true, analytical values used for the comparison. The analytical state values are then propagated in time according to Pulliam et al. [58] which are then used in obtaining the performance metrics as described in Section 4.1.3. These parameters are outlined in Table 5.1. p_L and ρ_L denote the initial pressure and density on the left side of the tube, while p_R and ρ_R represent the corresponding initial values on the right side. the heat capacity ratio, γ, is set to 1.4. Additionally, the spatial domain of the shock tube is defined by the coordinates x_L and x_R, with values of 0 and 1, respectively. The forward model used in DA, denoted as $\mathcal{M}$, is

Parameter	Value
p_L	1 Pa
ρ_L	1 Kg/m^3
p_R	0.1 Pa
ρ_R	0.125 Kg/m^3
γ	1.4
x_L	0
x_R	1

Table 5.1: Shock tube parameters

Chord. The analytical solution serves as the true and known values against which the performance of the MLES and MLEF methods will be evaluated and compared.

5.1.2 DA Configuration

Ensemble Initialization

Initializing the ensemble requires a diverse set of initial conditions to provide variance in the initial DA analysis. For every ensemble member, the parameters p_L, ρ_L, p_R, and ρ_R are generated randomly and is then propagated using these randomly selected values. Then, a random time is selected from this random assortment of initial parameters by drawing from a Gaussian distribution with a mean equal to the initial starting time of the DA analysis and a variance equal to the starting time, thereby contributing temporal diversity to the ensemble. The ensemble of N_e+1 members are selected from this random draw of configurations. Among these ensemble members, the control is the set of values that are closest to the true solution within this assortment. The ensemble initialization is illustrated in Figure 5.1, where the left subplot shows the distribution of initial densities, and the right subplot exhibits the corresponding pressures. These visualizations provide a clear depiction of the ensemble's initial states, forming the basis for the prior information utilized by the DA algorithm. The variability introduced during the initialization process ensures a diverse representation of possible states.

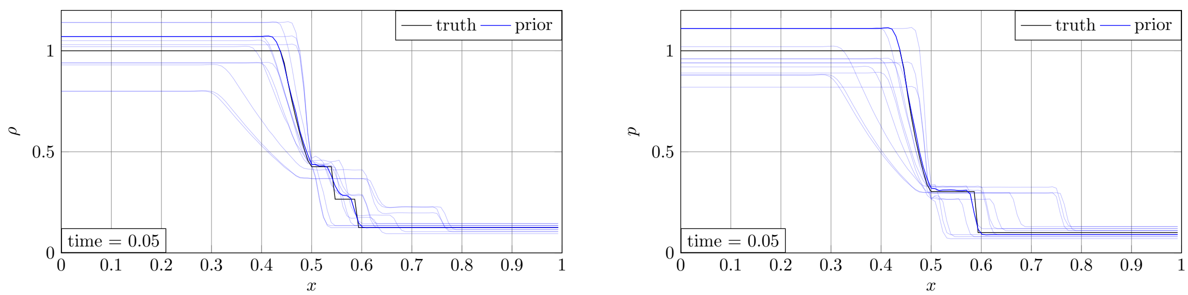

Figure 5.1: Initial state of pressure and density for the prior. For reference, the true state is given by the black line, and the prior is shown in blue. The thicker blue line shows the control and the thinner blue lines show the ensemble. In this figure, $N_e = 12$ and the initial starting time for the DA analysis is $t = 0.05$.

5.1.3 Performance Metrics

Performance metrics, including RMSE, shock location, shock quality, and quantitative analysis of the state values, play a pivotal role in evaluating the effectiveness of the MLES. RMSE serves as a fundamental measure, providing a quantitative assessment of the overall accuracy in predicting variables such as pressure and density. Shock location and shock quality offer specialized insights into the dynamic nature of shock waves by evaluating the MLES's ability to accurately position the shock and capture its "steepness." By employing these metrics, we gain a comprehensive understanding of how well the DA algorithm reproduces critical aspects of the shock tube's behavior, guiding the refinement and optimization of these methods for enhanced reliability in real-world fluid dynamics simulations.

RMSE

For root-mean-square-error calculations, we compare the state estimate of the shock tube provided by DA with the true, analytical values. The equation is

$$RMSE = \sqrt{\frac{\Sigma_i^N \left(x_{i,t} - x_{i,a}\right)^2}{N}} \tag{5.1}$$

where N is the number of cells chosen in the domain, $x_{i,t}$ is the true variable (pressure, density, etc.) at cell i, and $x_{i,a}$ is the analysis variable (pressure, density, etc.) at cell i provided by DA. Note this can be used for calculating either the *prior* values (ie before a DA analysis update) and *analysis* values. A distinction will be made in the results section.

Shock Location

The right-moving shock location is calculated and compared to the true location of the shock. The true shock location is found from the analytical solution, providing a reference. Numerically computing the shock location through Chord involves a series of computational steps. Firstly, the pressure gradient is computed through a straightforward second-order finite difference method, and a biased difference stencil is employed at the domain boundaries. An illustrative example is presented in Figure 5.2, showcasing the analytical truth and Chord's result of the pressure and pressure gradient.

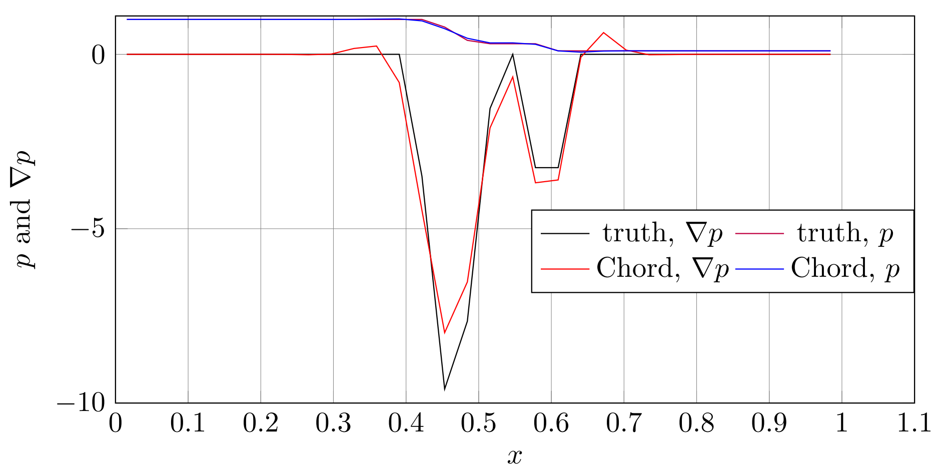

Figure 5.2: Pressure and gradient of pressure at $t = 0.05$ for the classic shock tube case. This figure shows the truth obtained by the analytical solution and the result from Chord.

To refine the estimation and to potentially capture the shock location between cells, a cubic spline is applied to the pressure gradient, where BSplineKit [59] is used. The result of the pressure gradient spline is shown in Figure 5.3.

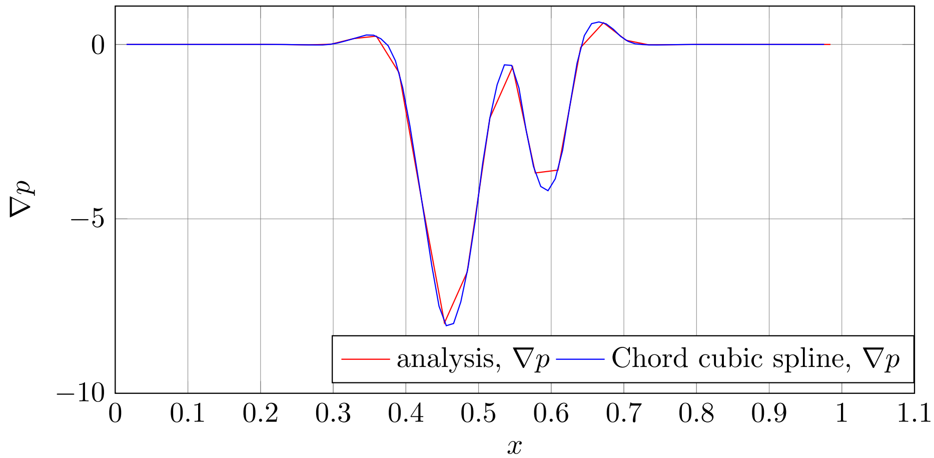

Figure 5.3: Gradient of pressure at $t = 0.05$ of chord, showing the cubic spline to interpolate between cells.

Lastly, a straightforward local minima algorithm identifies the minimum points in the pressure gradient. Focused on capturing the right-moving shock location, our search is constrained to the right-half of the domain, ensuring that the located position aligns with the correct pressure gradient minima. As depicted in Figure 5.3, the shock location is determined to be approximately 0.595, corresponding to the local minima of the pressure gradient.

Shock Quality

Shock quality serves as a critical metric to assess how accurately the DA method captures the "steepness" of the shock within the shock tube scenario. In CFD solvers, numerical techniques such as limiters are essential for maintaining solution stability and accuracy around discontinuities by imposing constraints on the gradients (e. g., McCorquodale et al. [60]). Namely, a first-order representation near sharp transitions to preserve monotonicity prevents spurious oscillations in the numerical solution [61]. The evaluation of shock quality involves measuring the minimum values of the pressure gradient, previously discussed. A percentage difference is then calculated between the true and analysis pressure gradient minima. In Figure 5.2, the minima of pressure at the shock location is approx -3.25, and for the analysis (using the cubic spline result in Figure 5.3), the minima is approx. -4.19. This percentage difference, using

$$\%diff.\text{ (or shock quality)} = \frac{|x_t - x_a|}{(x_t + x_a)/2} \tag{5.2}$$

where x_t is the truth and x_a is the analysis values. In this case, the percentage difference is 0.253. We are looking for a smaller percentage difference.

This particular metric is chosen because a higher pressure gradient indicates the shock is 'sharper'. Theoretically, the analytical pressure gradient is undefined due to the true discontinuity. However, with the constraint of discrete points, we can directly compare the quality of the discontinuity at the shock location. The shock quality metric provides a nuanced understanding of how well DA captures the sharpness of the shock, crucial for evaluating the fidelity of the assimilation process in representing the dynamics of the shock tube problem.

State Values

The assessment of state values at different times stands as a valuable and quantitative metric, offering a comprehensive evaluation of the DA method's performance. This comparison involves scrutinizing various state variables, such as pressure and density, at distinct time points throughout the simulation. By examining the evolution of these variables, we gain valuable insights into how well DA captures the temporal dynamics of the system. This qualitative approach provides an understanding of the DA's ability to reproduce complex phenomena, such as shock wave interactions and fluid behavior, enhancing the overall reliability of the simulation. The state values metric serves as a practical and intuitive measure, allowing for a holistic evaluation of the DA method's capability to accurately represent the intricate temporal variations within the simulated fluid dynamics.

5.1.4 Ensemble Size Sensitivity Test

We present results that compares the MLEF and MLES by adjusting the ensemble size, where N_e is the number of ensemble members. The objective of adjusting the ensemble size is to understand the impact of a reduced ensemble on the model's error statistics. As explained in Section 2.2, the ensemble is designed to capture the reduced subspace of the error statistics within the model. However, it is often impractical to span the entire error subspace due to computational limitations. By systematically varying the ensemble size from 8 to 40, the study aims to investigate the trade-off between computational efficiency and the ability to represent the model error statistics comprehensively.

Specifically for the MLES, the number of observations taken in the DA window is fixed at 4. The MLEF is the classic configuration where the analysis time is at the current time and the observation is also taken at the analysis time. The main goal is to provide insight as to the optimal ensemble size. This optimal ensemble size will be a compromise between performance and computational resources required. A performance saturation is expected at some point and an optimal ensemble size will be chosen to be used for the rest of the experiments.

Other Initial DA Configuration Parameters

Some other pertinent DA configuration parameters are given in Table 5.2. A meticulous approach was adopted in selecting these parameters, resembling a 'perfect' numerical experiment. This choice aims to yield optimal results for the shock tube problem which serves as an important verification step before using more complex DA configurations. The rationale behind this strategic selection is twofold: first, it ensures a comprehensive understanding of DA's functionality under an ideal, best-case-scenario setting, laying a robust groundwork for subsequent analyses. Second, by establishing a successful baseline, the focus can seamlessly shift towards applying more intricate DA configurations involving increased nonlinearity and reduced degrees of freedom. This enables a systematic exploration of the alterable DA configurations which is important to understand because it provides insight into progressively challenging scenarios.

The initial steps, characterized by the 'perfect' DA numerical experiment, thus serve as a critical preparatory phase, setting the stage for the in-depth investigation and application of more sophisticated DA configurations in subsequent sections of this thesis. The end time is chosen to be $t = 0.23$, which is before the shock hits the boundary. The number of analyses is adjusted so that

Parameter	Value
Analysis inflation	1.1
Observation uncertainty	$1.0e-05$
τ_{DA} (time between observations)	0.01
spin (t for DA start)	$t = 0.05$
end time	$t = 0.23$
observations	pressure (acquired from truth)
observations spatial locations	every cell
lag (MLES only)	4
shift (MLES only)	4

Table 5.2: DA parameters

both the MLEF and MLES finish their respective analyses before the end time. In this case, the MLEF has analyses points at

$$t_{\text{analysis times, MLEF}} = 0.05, 0.06, \ldots, 0.22, 0.23,$$

and the MLES has analyses at

$$t_{\text{analysis times, MLES}} = 0.05, 0.09, 0.13, 0.17.$$

Because the analysis for the MLES takes observations at future locations, it must stop once observations are not available. We still look at the resulting DA metrics at each time, even if no analyses occur at some points.

MLEF

Various state values are shown for each case. To show the initial analysis time for both the MLES and MLEF, pressure plots are shown of the truth, observations, prior, and analysis pressures. These plots serve to provide a visual representation of the analysis update's impact during the initial time for DA. The opaque red and blue lines represent the ensemble members of the analysis and prior, respectively, while the gray dashed line corresponds to the free run with no DA applied. For this free run, the initial parameters are the same as in Table 5.1.

The configuration for the MLEF and $N_e = 16$ is shown in Figure 5.4. The shock at approximately $x = 0.58$ in the domain is improved dramatically.

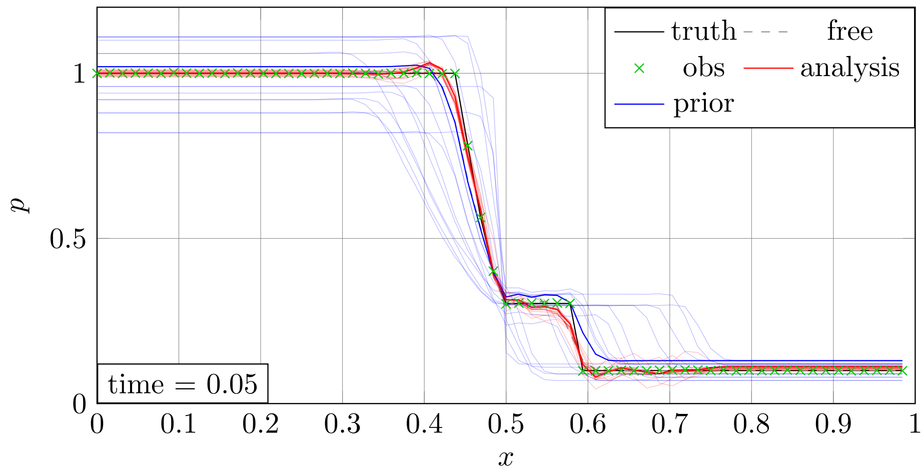

Figure 5.4: Pressure values for the MLEF at $t = 0.05$, or the initial analysis time. The opaque red and blue lines show the ensemble members of the analysis and prior, respectively. The gray dashed line is the free run with no DA applied.

MLES

Now the configuration for the MLES and $N_e = 16$ is shown in Figure 5.6. The improvement for both the MLEF and MLES serves as a noteworthy validation milestone wherein both the MLEF and MLES effectively captures and refines the system's dynamics during the initial analysis time.

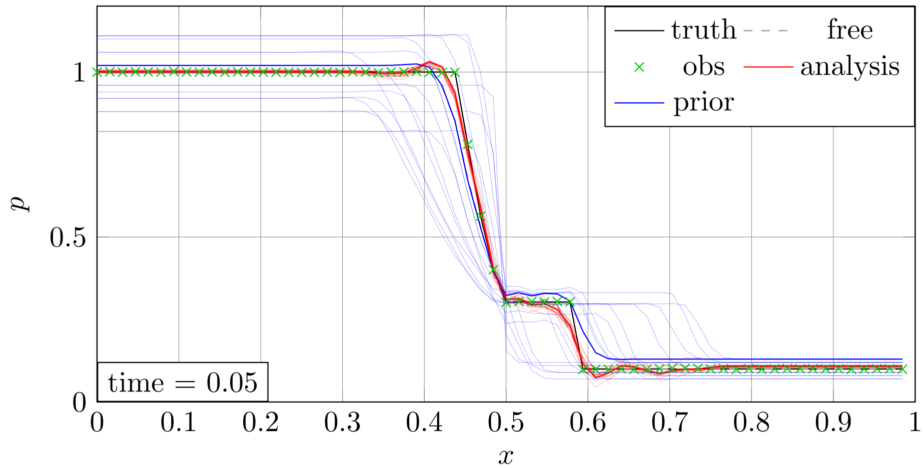

Figure 5.5: Pressure values for the MLES at $t = 0.05$, or the initial analysis time. The parameters for the MLES are $L = 4$ and $S = 4$. The opaque red and blue lines show the ensemble members of the analysis and prior, respectively. The gray dashed line is the free run with no DA applied.

Direct Comparison at Right-Moving Shock

Here, the analysis is shown for both the MLES and MLEF in the same plots as above with an emphasis at the right-moving shock. The prior states are omitted in this case for clarity, as they are the same for both the MLEF and MLES.

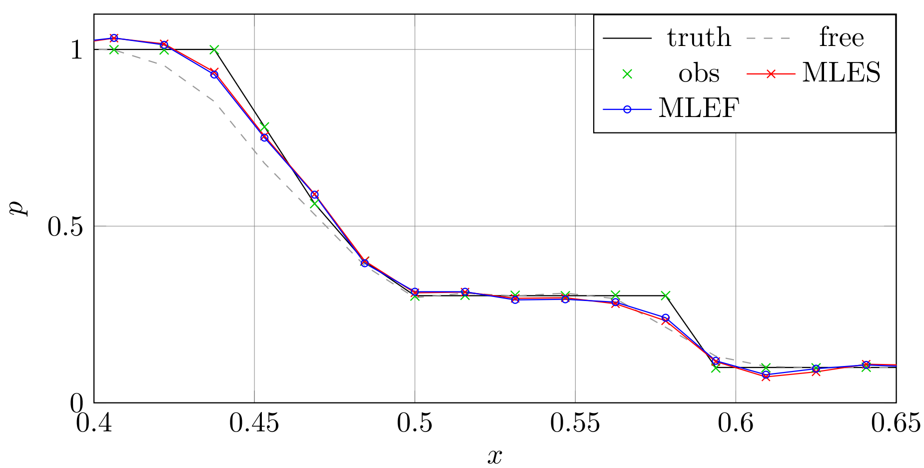

Figure 5.6: Pressure values for the MLES and MLEF at $t = 0.05$, or the initial analysis time. Zoomed in on the shock location.

The MLES exhibits a slightly better performance in capturing the dynamics near the right-moving shock. This is indicated by the visual inspection that the MLES analysis is updated slightly closer to the true values. This qualitative performance shows the capability of the MLES methodology to provide a marginally more accurate representation of the system's state variables during the initial analysis time for this particular numerical experiment at this particular solution time. The visual emphasis on the shock location within the comparison plot illuminates the subtle differences in the predictive capabilities of these two data assimilation methods.

MLEF and MLES with $N_e = 16$: Time Comparisons

State comparisons for the MLEF and MLES values are given for pressure at different times. The primary objective is to ensure the foundational elements of the DA process are operating optimally. The focus remains on observing and verifying the behavior of the ensemble analyses at different time points, providing a comprehensive understanding of the assimilation process's dynamics. First, plots of the MLEF with $N_e = 16$ are shown in Figure 5.7. Equivalent plots are shown for the MLES in Figure 5.8.

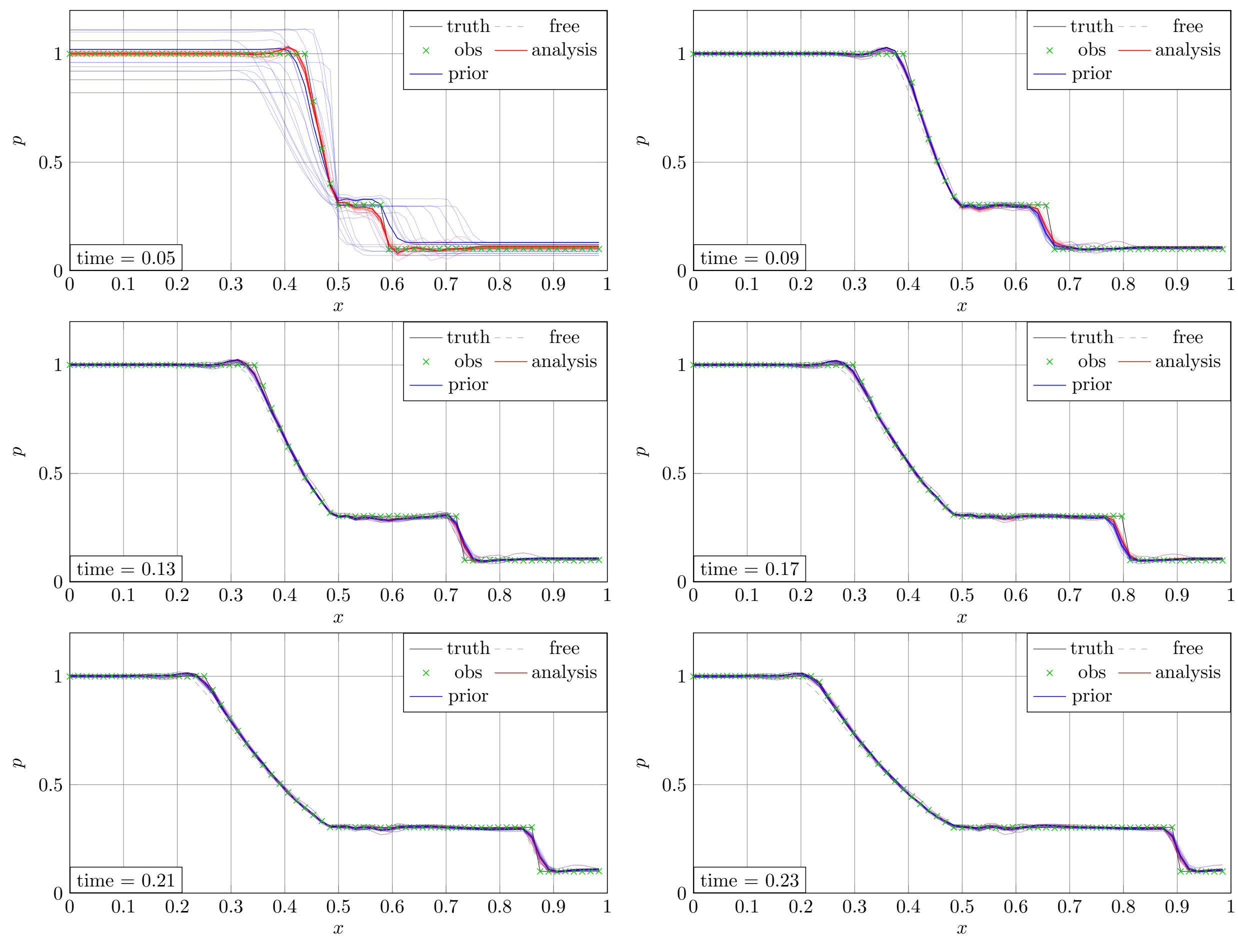

Figure 5.7: Pressure values for the MLEF of $N_e = 16$ at different times.

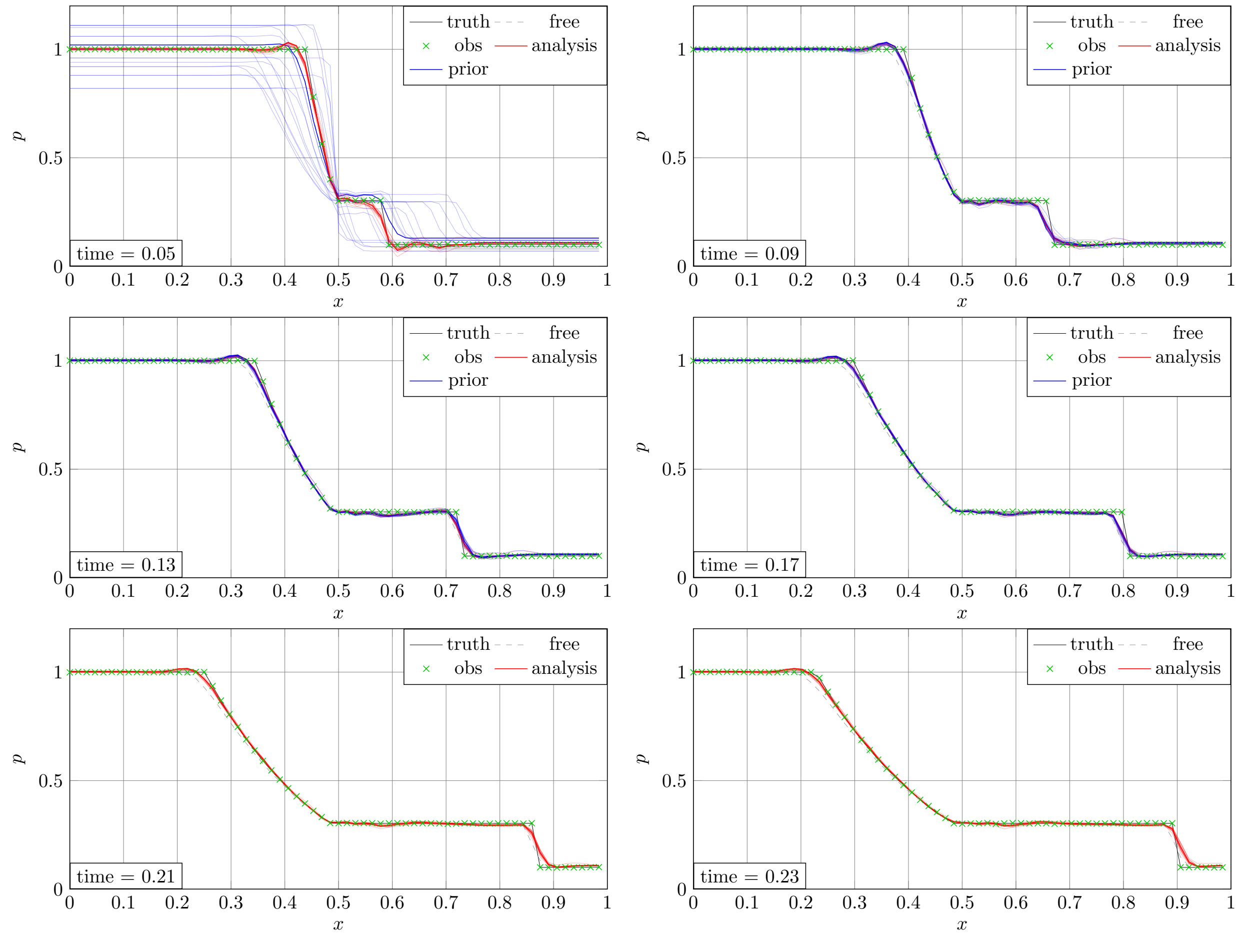

Figure 5.8: Pressure values for the MLES of $N_e = 16$ at different times.

One observation emerges concerning the analysis update of the ensemble. There is a discernible trend toward a low variance in the ensemble after the initial update, denoting a relatively small spread in the analysis ensemble. This characteristic is primarily attributed to the small observation errors. This exerts a large influence on the analysis update spread. This phenomenon, while influencing the ensemble's variance, does not detract from the overall objective of this phase, which is to validate the foundational aspects of the DA process. The subsequent sections will delve into more complex configurations, leveraging the insights gained from this preliminary analysis.

Performance Metrics of the MLEF and MLES with $N_e = 8 - 40$

Pressure plots for the MLEF and MLES are given for $N_e = 8$, 16, 24, 32, and 40. The free run is simulated again with the initial conditions in Table 5.1. Figure 5.9 shows the results for the MLEF and Figure 5.10 shows the results for the MLES. Each of these figures focuses on the shock location and zooms in on the area of interest. For $N_e = 16$, 24, 32, and 40, the results of pressure show minimal difference whereas $N_e = 8$ is visually a worse. The same trend is seen for the MLES, where $N_e = 16$, 24, 32, and 40, show minimal differences in pressure.

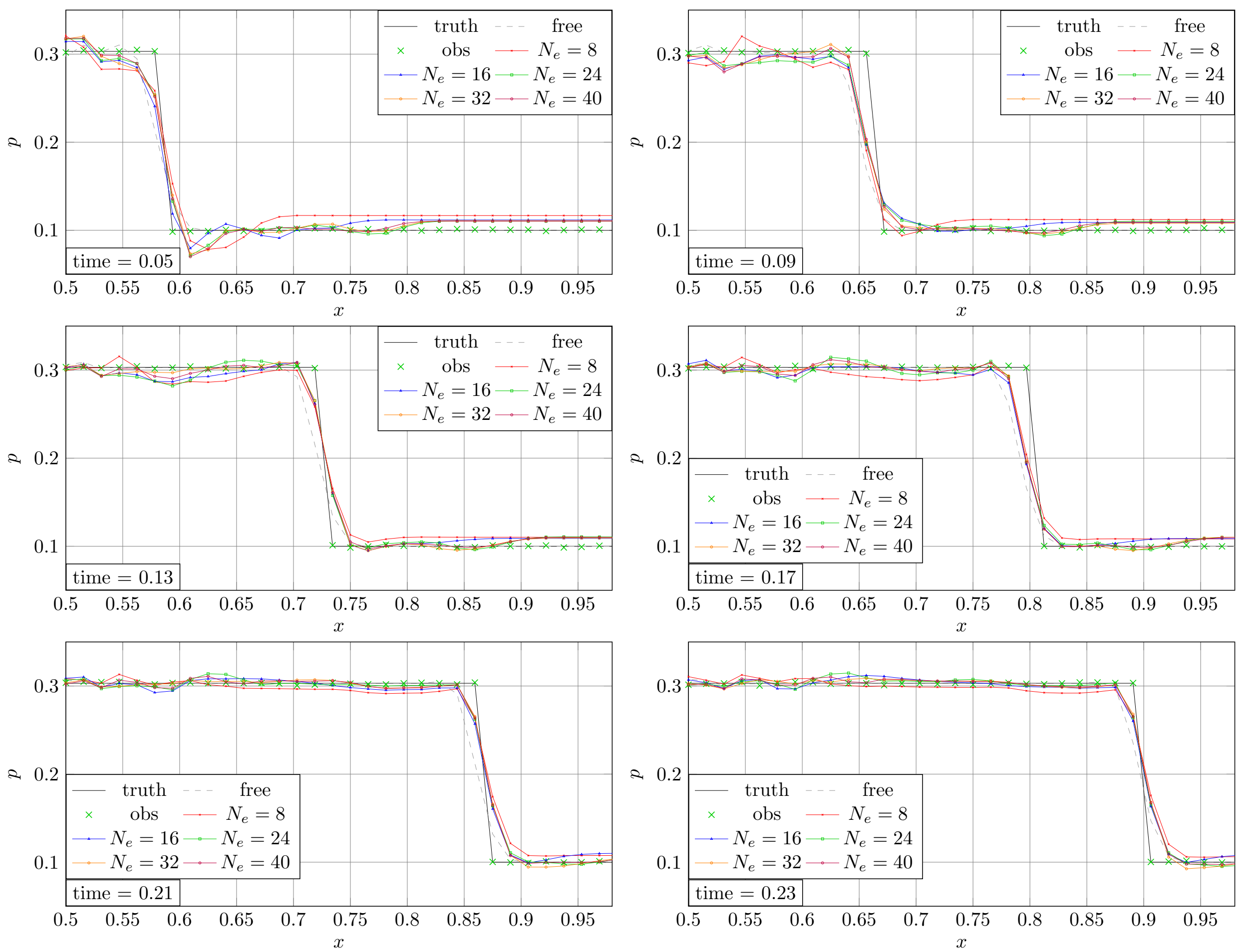

Figure 5.9: Pressure values at the shock location for the MLEF at different times and different N_e values.

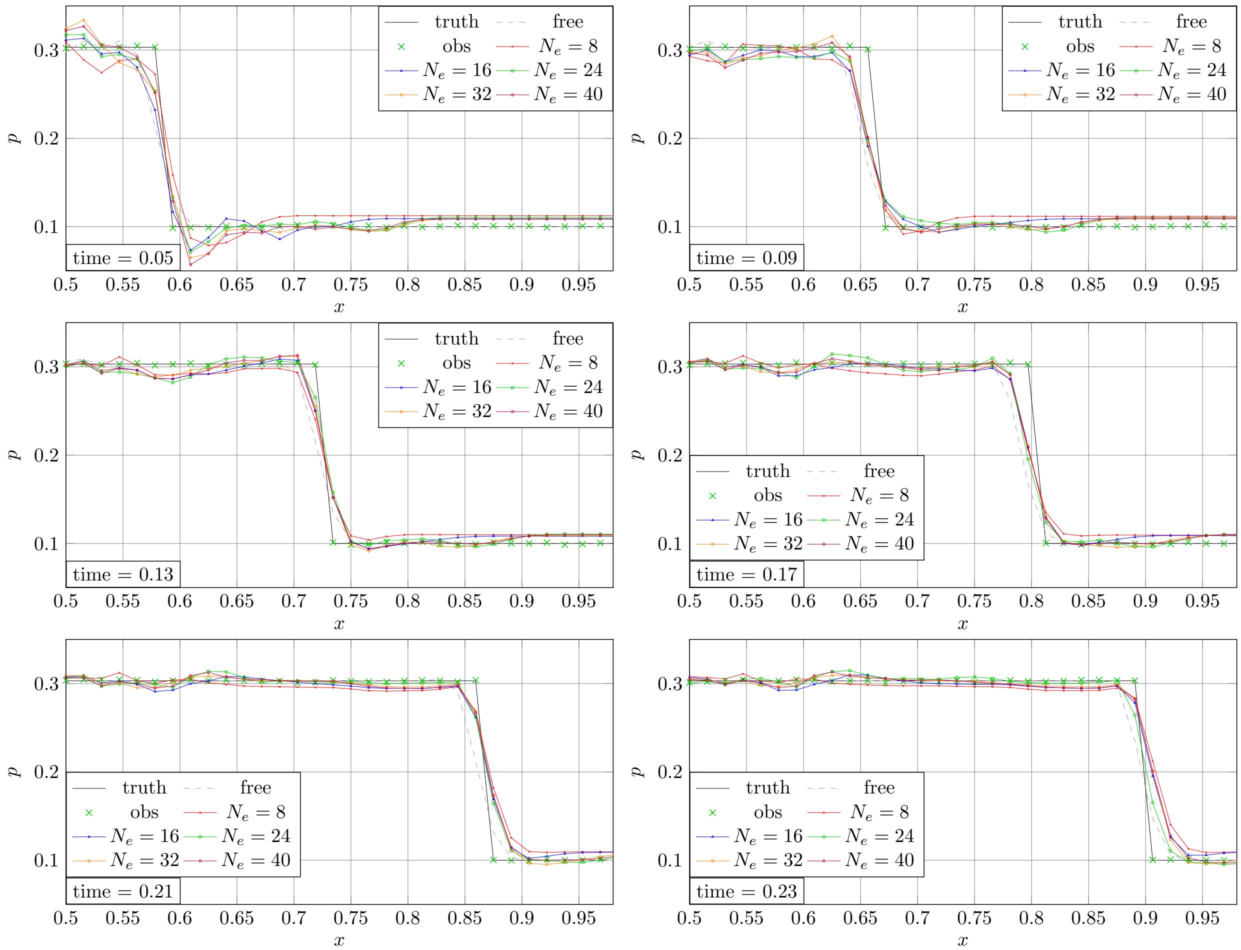

Figure 5.10: Pressure values at the shock location for the MLES at different times and different N_e values.

The RMSE, shock location, and shock quality results are now given. Figure 5.11 shows the pressure RMSE for the MLEF and Figure 5.12 shows the equivalent metric for the MLES. Figure 5.13 shows the physical shock locations for the MLEF and Figure 5.14 shows the shock locations of the MLES. Figure 5.15 shows the shock qualities for the MLEF and Figure 5.16 shows the shock qualities of the MLES. Each figure shows the metric values at the temporal points where there are available observations. For the MLEF, this also corresponds to every analysis.

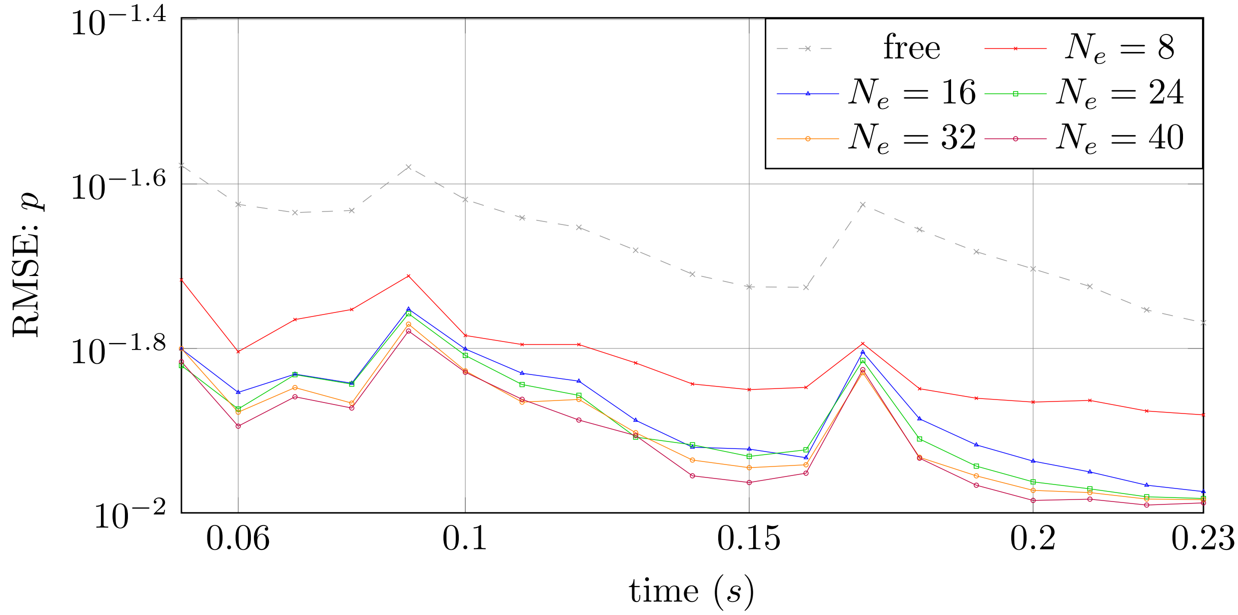

Figure 5.11: Pressure RMSE values for the MLEF at different times and different N_e values.

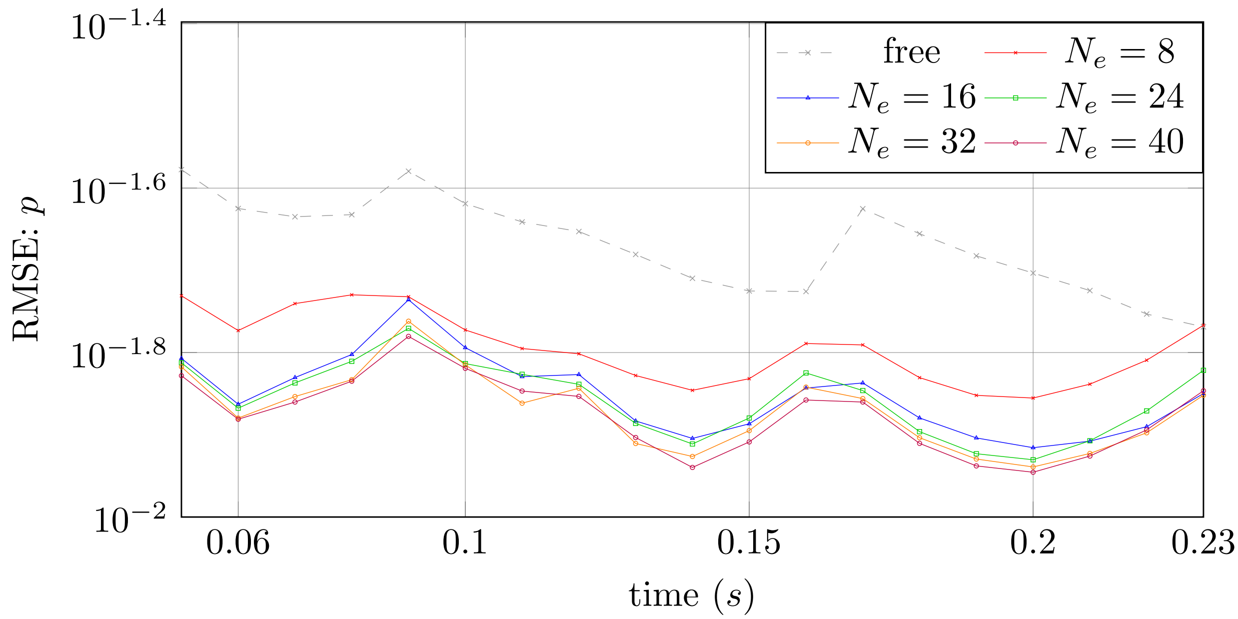

Figure 5.12: Pressure RMSE values for the MLES at different times and different N_e values.

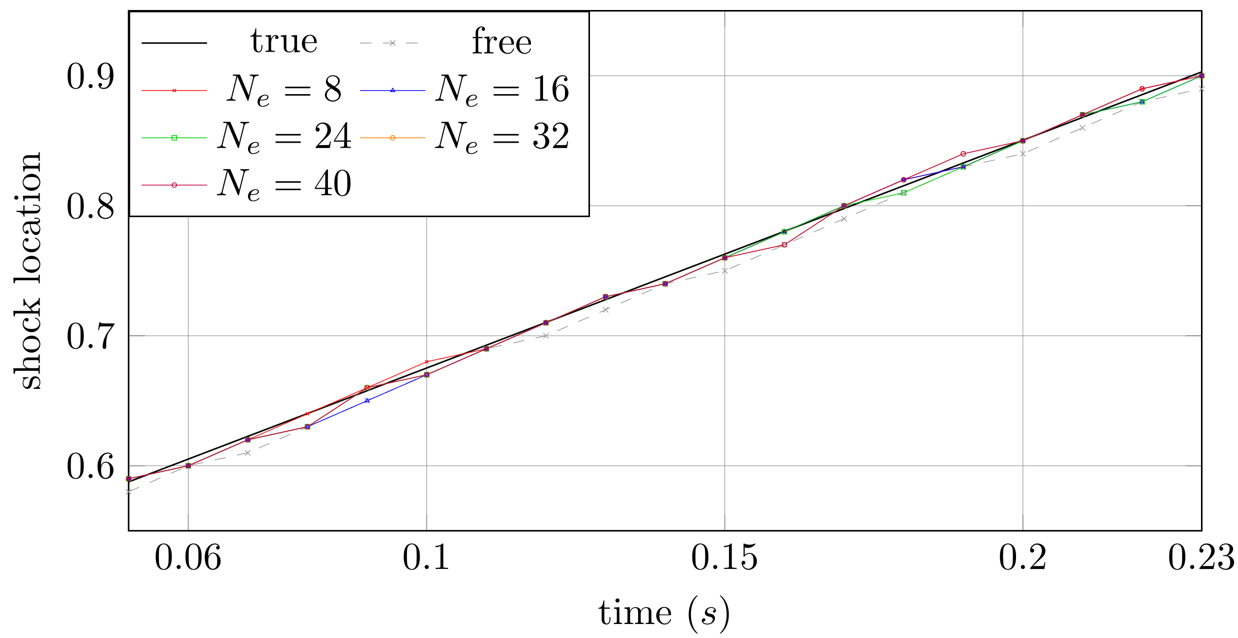

Figure 5.13: Shock location for the MLEF at different times and different N_e values.

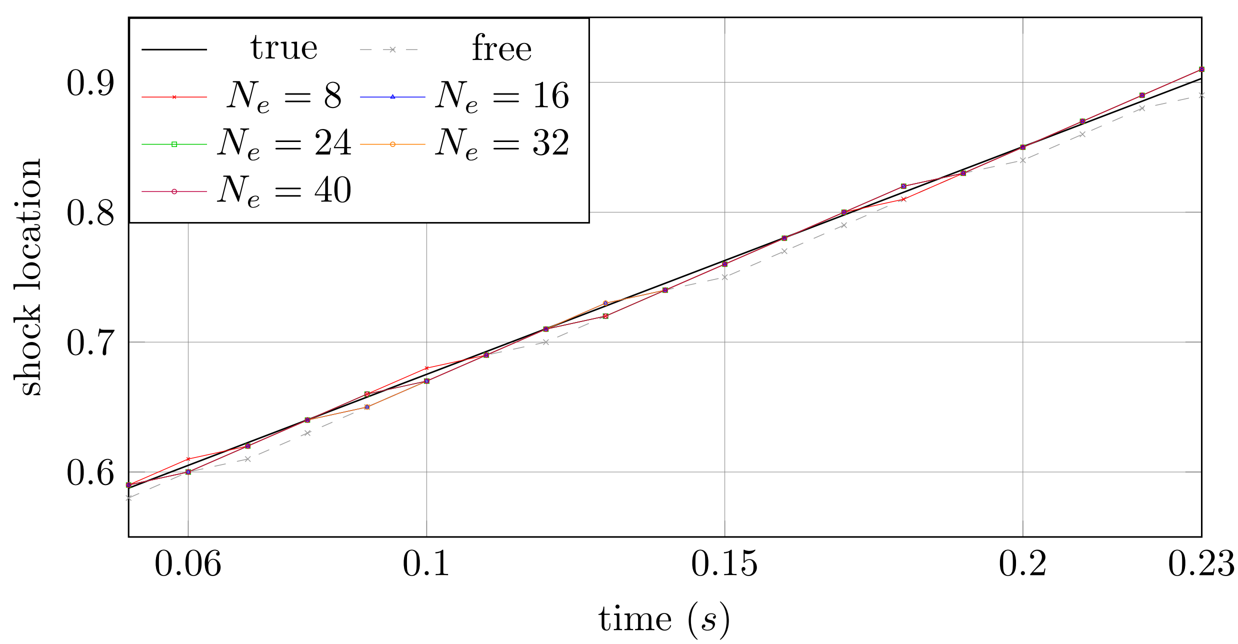

Figure 5.14: Shock location for the MLES at different times and different N_e values.

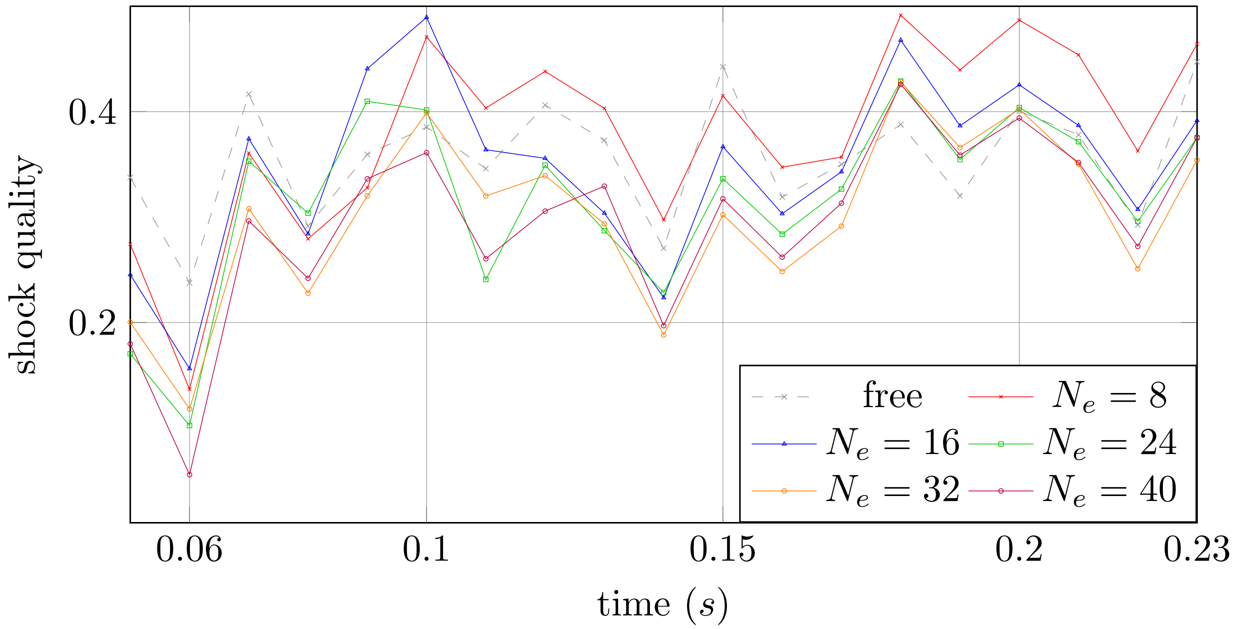

Figure 5.15: Shock location for the MLEF at different times and different N_e values.

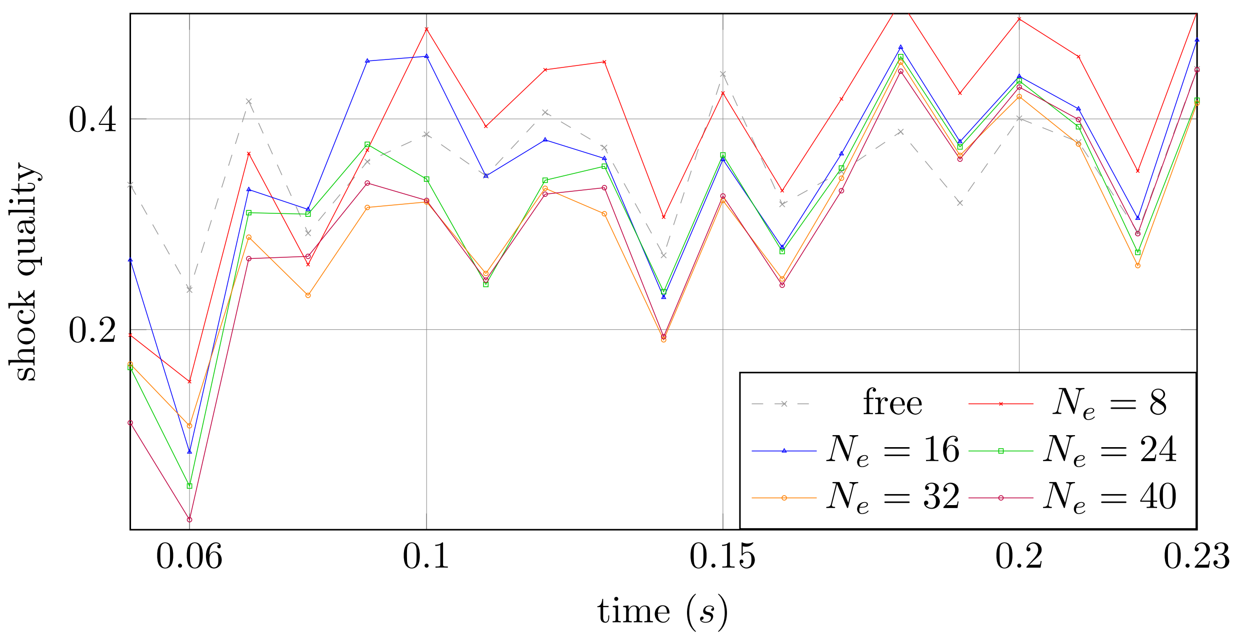

Figure 5.16: Shock location for the MLES at different times and different N_e values.

This ensemble size sensitivity analysis shows the importance of selecting an appropriate N_e that balances computational efficiency and optimal performance. The negligible variations observed in pressure values for higher N_e values suggest the potential for achieving stable and accurate results with a larger ensemble. In analyzing the sensitivity study on different ensemble sizes, several trends emerge across the performance metrics. In the RMSE plots, a larger ensemble size generally leads to improved results, aligning with the expected behavior that a larger ensemble can better capture the error statistics of the model. However, as seen in the pressure plots, the ensemble analysis spreads are quite small and thus the error statistics may be lost regardless. In Figure 5.12, deteriorating RMSE values after $t = 0.2$ is an indication that there are no analysis after $t = 0.17$. This is because the later analysis times would require observations outside the domain (i. e. in this numerical setup, an analysis time at $t = 0.21$ would require the last observation at $t = 0.25$, which does not exist). The shock location metric proves that each configuration is fully capable of capturing the location of the shock, but no discernible difference can be seen between configurations. All configurations perform slightly better than the free run which is to be expected. This observation is especially promising as the free run's initial parameters are the same as in Table 5.1, indicating that even with known initial parameters, the MLES with unknown initial parameters outperforms the CFD simulation. The shock quality metric shows that all methods initially perform acceptably but exhibit a quick deterioration. This decline is attributed to the small variance in ensemble updates. Averaging the shock quality values could potentially provide a more stable representation of the overall performance. The inherent variability in the updates, influenced by the specific locations where the ensemble priors are updated, introduces an element of uncertainty, contributing to the fluctuations observed in the results.

These findings provide a basis for selecting an ensemble size in subsequent analyses. To give a balanced trade-off between computational resources and the accuracy of the analysis, an ensemble size of $N_e = 16$ is deemed optimal as no significant loss in performance was noticed. This choice represents a practical compromise, considering the observed trends and the associated computational costs, offering a trade-off that meets the objectives effectively.

Adjusting DA Window Length

In the next numerical experiment, we explore the complexities introduced by increased nonlinearity by adjusting the length of the MLES's DA smoother window. Lengthening the DA window with more observations provides the assimilation process with more observational information, as well as more flow-dependent errors. However, this also presents a trade-off in terms of introducing nonlinearity stemming from the prediction model. The challenge lies in determining the optimal number of observations within the updating window, considering the nonlinear nature of the model. The shock tube problem, propagated by Chord, serves this purpose due to its high nonlinearity. Our focus is to understand the limitations associated with increasing the DA window size by incorporating more observations, and extending their temporal range from the current analysis time. The following numerical experiments aim to provide this quantitative information.

With the results from the N_e study, a fixed ensemble size of $N_e = 16$ is used. L is adjusted from $2 - 8$ and S is adjusted from $2 - 8$. For clarity, all possible combinations of L and S are omitted for clarity and only the optimal analysis is shown for the given MLES configurations. In this numerical experiment, observations are now taken at computational cells close to the shock and their random realizations are now 0.01. This simulates a pressure sensor that follows the shock as it evolves. Additionally, the free run now shares the same initial conditions as the prior of the DA runs, simulating the same unknown initial conditions as the MLES. Due to this, the free run, without any DA applied and having initial conditions far from the actual true shock tube initial parameters, is distinctly worse than all states with DA applied. Figure 5.17 shows the pressure plots at various times for the MLES with several combinations of L and S. Figure 5.18 shows the pressure RMSE, Figure 5.19 shows the physical shock locations, and Figure 5.20 shows the shock qualities. Each figure shows the metric values at the temporal points where there are available observations.

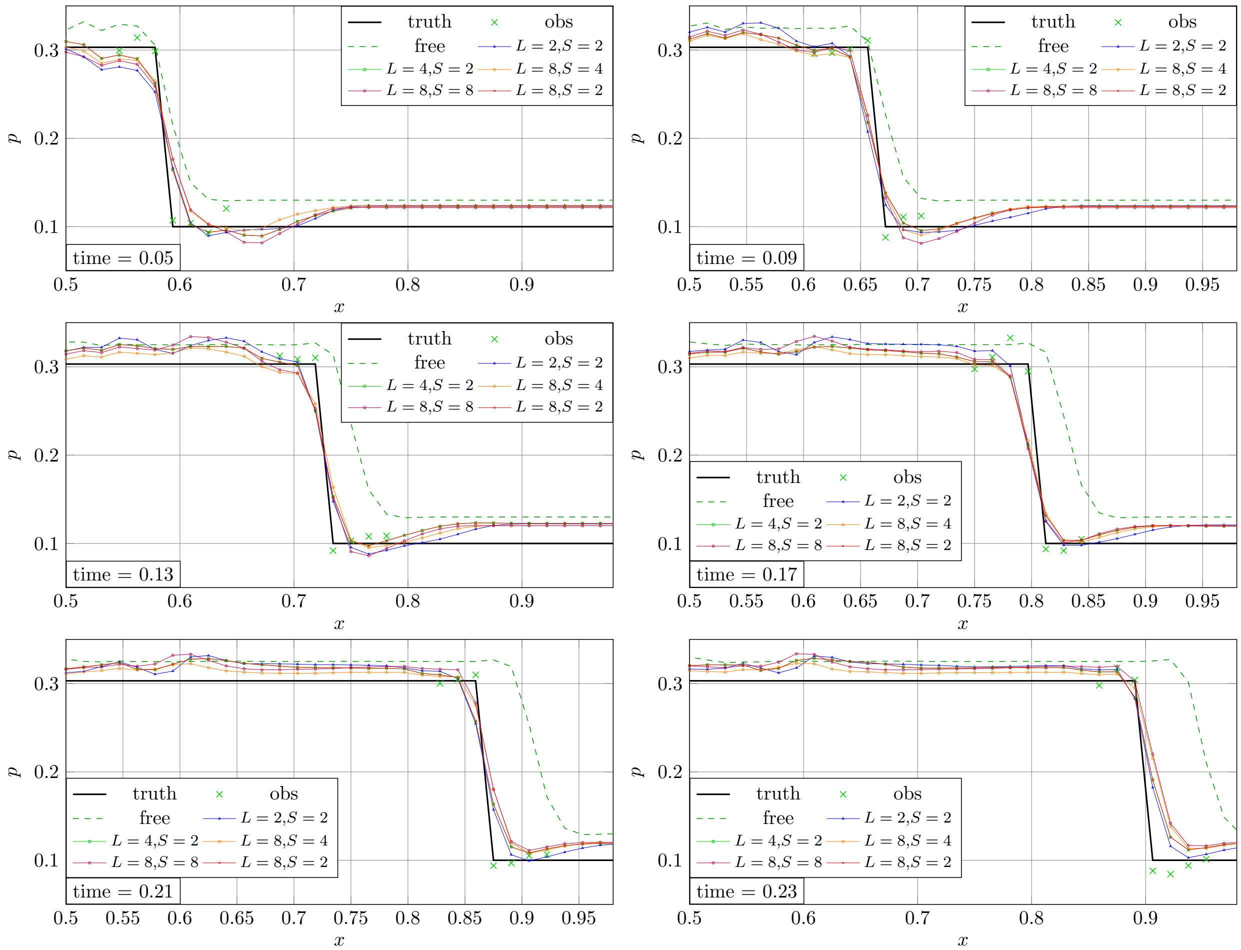

Figure 5.17: Pressure at different times and for different configurations of the MLES. A portion of the domain is shown for identifying the quality of the shock capturing mechanics.

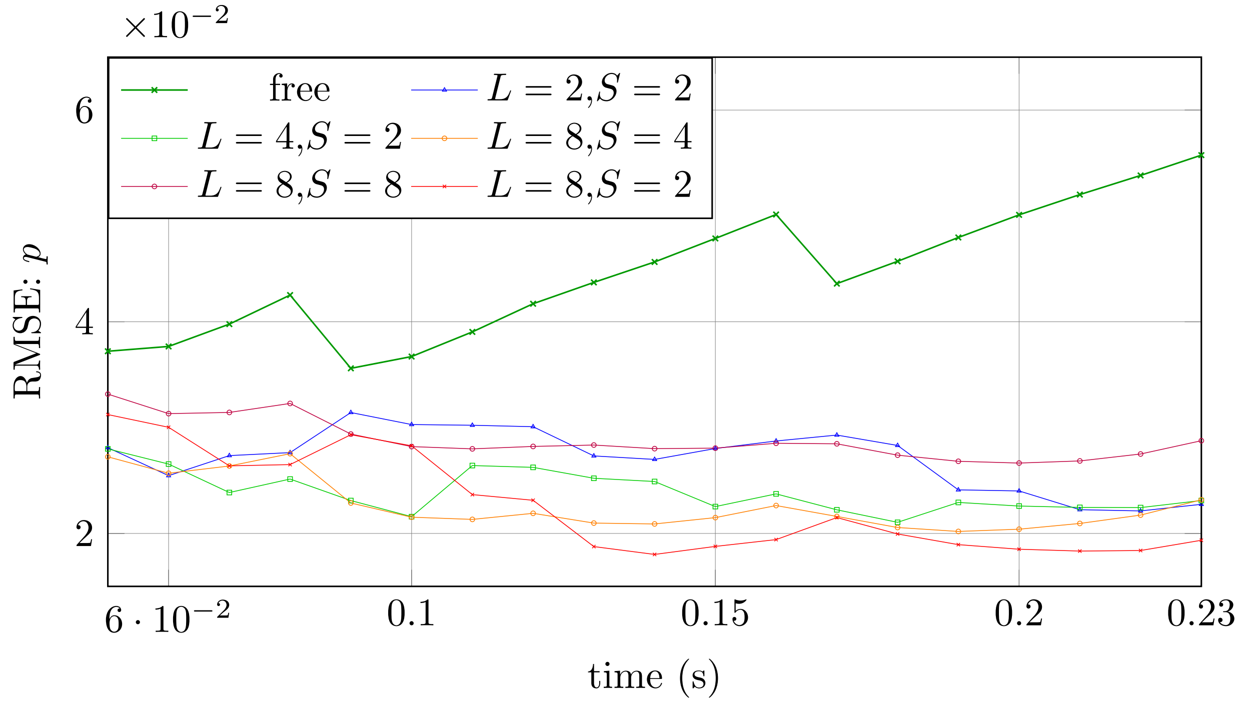

Figure 5.18: Pressure RMSE at different times and for different configurations of the MLES.

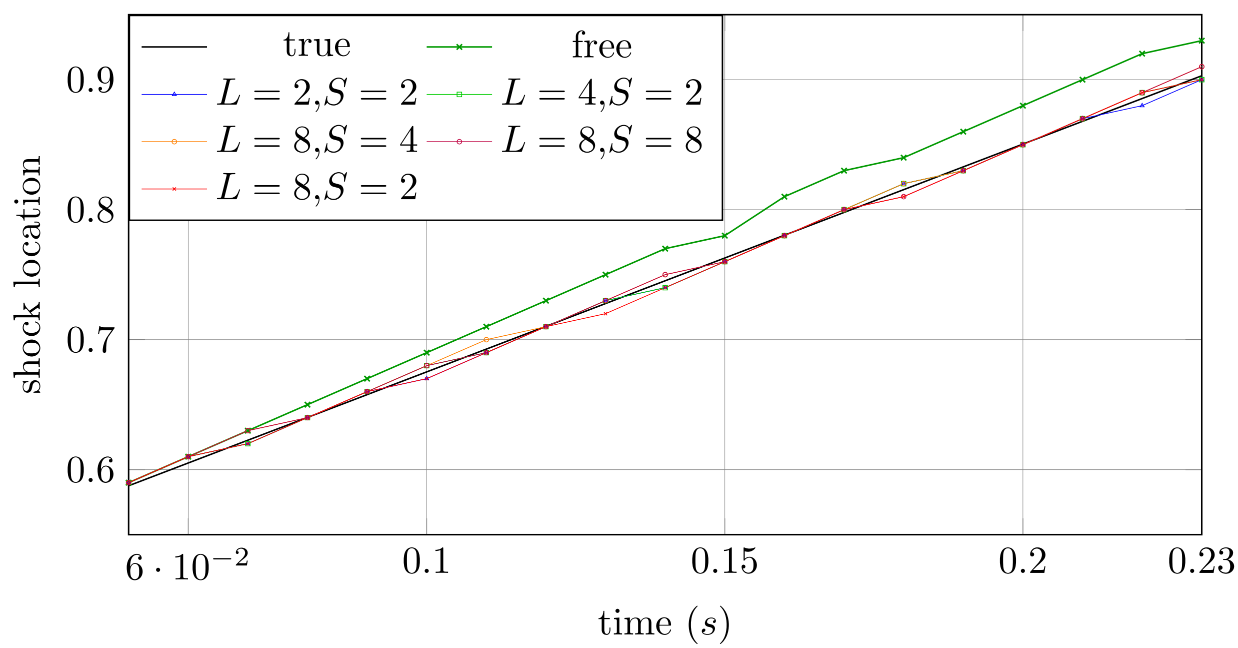

Figure 5.19: Pressure at different times and for different configurations of the MLES.

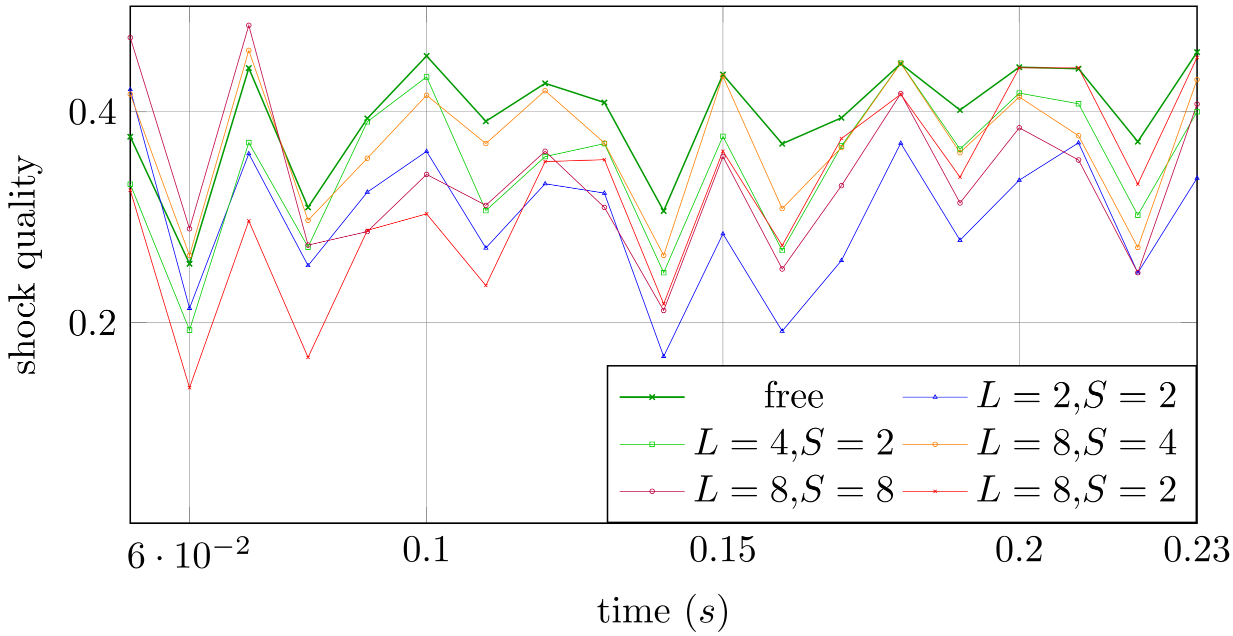

Figure 5.20: Shock quality at different times and for different configurations of the MLES.

5.2 Conclusions

Our findings suggest that the MLEF has limitations in effectively capturing shock dynamics compared to the MLES. This is due in part by the incorporation of the model in the minimization of the MLES, thus allowing the system's dynamics by an integral part of the analysis. While MLES shows promise in improving forecast accuracy, obtaining better performance with MLES remains challenging due to the intricacies involved in configuring DA parameters. Future research efforts should focus on refining these parameters and exploring alternative strategies to enhance the effectiveness of MLES in capturing shock dynamics. Additionally, further investigation is warranted to explore the full potential of MLES in addressing similar highly nonlinear problems, with emphasis on refining its implementation for practical applications.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This dissertation presents the creation and application of a novel DA method, the MLES, designed to improve the predictability of CFD modeling of complex fluid dynamics problems. By using the MLES, we integrate data into the CFD model to enhance fluid flow predictions and show its effective potential in improving CFD modeling as whole. CFD solvers like Chord serve as the foundation for the DA+CFD system due to its advanced features, such as ARK4 and MMB, which ensure reliability and computational efficacy. Performance evaluations against the MLEF demonstrate the effectiveness of the MLES and is supported by a performance study using the L96 model and Kuramoto-Sivashinsky equation. Both algorithms demonstrate a positive impact on the analysis states in each presented experiment, with the MLES performing at least comparably or slightly better than the MLEF. More significant impacts are expected as the problems become more complex. Numerous similar experiments indicate that the initial setup of model parameters for ensemble members and the control run significantly influences result quality, defined by proximity to true values. It has been shown that variations in observation variance have less impact than changes in model parameters, suggesting that empirical knowledge of the dynamical system outweighs the influence of gathered data. It's worth noting that when comparing the model parameter estimation, it assumes an imperfect initialization of parameters. This mimics reality, where parameters are unobservable and data is collected from physical experiments. Consequently, the notion of converging to true model parameters using synthesized data differs from actual physical scenarios. Nevertheless, the methodology remains applicable when using data from physical experiments and shows its potential for application in practical engineering flows.

Additionally, we introduce a novel approach by integrating MGinT with the MLES to enhance computational efficiency and practical applicability. This integration results in reduced minimization iterations on fine-time-grids, demonstrating improved computational efficiency. Performance evaluations against the MLES show the effectiveness of this approach, providing significant computational speedups. The integration of MGinT with MLES offers a promising avenue for advancing the capabilities of DA in improving the computational efficiency.

Furthermore, the efficacy of the MLES combined into the DA+CFD system is assessed through its application to a 1D shock tube, serving as a robust platform for assimilating measurements obtained from the true analytical solution. By extending the temporal range and incorporating more observations within the DA window, this study aims to uncover the limitations associated with managing nonlinearities while leveraging observational information effectively in the context of the shock tube problem. The ensemble size sensitivity analysis reveals the critical importance of selecting an appropriate N_e, striking a balance between computational efficiency and performance optimization. While larger ensemble sizes generally lead to improved results in terms of RMSE, the inherent variability in updates introduces uncertainty, impacting shock quality metrics. Despite challenges in interpreting shock location metrics, a compromise at $N_e = 16$ demonstrates optimal performance without significant loss, aligning with computational resource considerations. Transitioning to the exploration of increased nonlinearity, extending the DA window provides valuable insights into leveraging observational information while managing nonlinearities. This performance assessment of the shock tube problem shows the complexities associated with incorporating more observations, offering crucial considerations for future analyses seeking to balance observations and nonlinear model dynamics effectively.

6.2 Original Contributions

Key original contributions of this research include:

1. Development of a novel DA method, MLES, for application to complex CFD problems.

2. Integration of a novel approach, MGinT, to increase the computational efficiency of the MLES.

3. Application of the novel DA method, a hybrid ensemble-variational method capable of handling highly nonlinear problems, to Sods' shock tube problem.

6.3 Future Work

In this study, we have explored the influence of a hybrid ensemble-variational smoother DA method, the MLES, on the predictability of challenging flow dynamics, involving chaotic and discontinuous flows. Future research may center on

1. Improving the understanding of the uncertainty associated with model parameters for the development and evaluation of data-driven models.

2. Demonstrating a comprehensive understanding of the impact the DA parameters have on the non-Gaussian of turbulent-combustion problems.

Bibliography

[1] E. N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20:130–141, 1963.

[2] M. Yamada and K. Ohkitani. Lyapunov spectrum of a model of two-dimensional turbulence. *Phys. Rev. Lett.*, 60:983–986, Mar 1988.

[3] H. Kato and S. Obayashi. Approach for uncertainty of turbulence modeling based on data assimilation technique. *Comput. Fluids*, 85:2–7, 2012.

[4] A. J. Reagan, Y. Dubief, P. S. Dodds, and C. M. Danforth. Predicting flow reversals in a computational fluid dynamics simulated thermosyphon using data assimilation. *PLoS ONE*, 11(2), 2016.

[5] R. Diez. *Machine learning for RANS turbulence modelling of variable property flows*. PhD thesis, Delft University of Technology, 2018.

[6] K. Duraisamy, G. Iaccarino, and H. Xiao. Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.*, 51:357–377, 2019.

[7] E. Kalnay. *Atmospheric modeling, data assimilation and predictabilty*. Cambridge University Press, 1st edition, 2003.

[8] J. Lewis, S. Lakshmivarahan, and S. Dhall. *Dynamic Data Assimilation: A Least Squares Approach*. Cambridge University Press, 2006.

[9] G. Evensen. *Data assimilation: the ensemble Kalman filter*. Springer, 2007.

[10] X. Gao and J. Liu. Data assimilation for computational fluid dynamics modeling of combustion, May 2014. 26th International Conference on Parallel Computational Fluid Dynamics.

[11] H. Kato, A. Yoshizawa, G. Ueno, and S. Obayashi. A data assimilation methodology for reconstructing turbulent flows around aircraft. *J. Comput. Phys.*, 283:559–581, 2015.

[12] X. Gao, Y. Wang, N. Overton, M. Zupanski, and X. Tu. Data-assimilated computational fluid dynamics modeling of convection-diffusion-reaction problems. *J. Comput. Sci.*, 21:38–59, 2017.

[13] C. He, Y. Liu, and L. Gan. A data assimilation model for turbulent flows using continuous adjoint formulation. *Phys. Fluids*, 30:105108, 2018.

[14] Y. Wang, S. Walters, S. Guzik, and X. Gao. CFD modeling of bluff-body stabilized premixed flames with data assimilation. AIAA 2020-0352, 2020 AIAA SciTech Forum, January 2020.

[15] A. C. Lorenz. Analysis methods for numerical weather prediction. *Q. J. R. Meteorol. Soc.*, 112:1177–1194, 1986.

[16] P. J. VanLeeuwen. Particle filtering in geophysical system. *Mon. Wea. Rev.*, 2137:4089–4114, 2009.

[17] P. L. Houtekamer and H. L. Mitchell. A sequential ensemble kalman filter for atmospheric data assimilation. *Mon. Wea. Rev.*, 129(1):123–137, 2001.

[18] T. Hamill, J. S. Whitaker, and C. Snyder. Distance-dependent filtering of background error covariance estimates in an ensemble kalman filter. *Mon. Wea. Rev.*, 129:2776–2790, 2001.

[19] G. Evensen. The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean Dyn.*, 53:343–367, 2003.

[20] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. J. Patil, and J. A. Yorke. A local ensemble kalman filter for atmospheric data assimilation. *Tellus A*, 56(5):415–428, 2004.

[21] D. T. Pham, J. Verron, and M. C. Roubaud. A singular evolutive extended kalman filter for data assimilation in oceanography. *J. Mar. Syst.*, 16:323–340, 1998.

[22] J. L. Anderson and S. L. Anderson. A monte carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Mon. Wea. Rev.*, 127:2741–2758, December 1999.

[23] J. Anderson. Spatially and temporally varying adaptive covariance inflation for ensemble filters. *Tellus A*, 61(1):72–83, 2009.

[24] O. Talagrand and P. Courtier. Variational assimilation of meteorological observations with the adjoint vorticity equation. i: Theory. *J. Meteorol. Soc. Japan*, 113:1311–1328, 1987.

[25] T. Schlatter. Variational assimilation of meteorological observations in the lower atmosphere: A tutorial on how it works. *J. Atmos. Sol. Terr. Phys.*, 62(12):1057–1070, 2000.

[26] Y. Wang, S. Guzik, M. Zupanski, and X. Gao. The maximum likelihood ensemble filter for computational flame and fluid dynamics. *IMA J. Appl. Math.*, 86(4):631–661, 2021.

[27] A. C. Lorenz. The potential of the ensemble kalman filter for nwp – a comparison with 4d-var. *Q. J. R. Meteorol. Soc.*, 129:3183–3203, 2003.

[28] E. Kalnay, H. Li, T. Miyoshi, S. C. Yang, and J. Ballabrera-Poy. 4-d-var or ensemble kalman filter? *Tellus A*, 59(5):758–773, 2007.

[29] R. N. Bannister. A review of operational methods of variational and ensemble-variational data assimilation. *Q. J. R. Meteorol. Soc.*, 143:607–633, 2017.

[30] S. E. Cohn, N. S. Sivakumaran, and R. Todling. A fixed-lag kalman smoother for retrospective data assimilation. *Mon. Wea. Rev.*, 122:2838–2867, 1994.

[31] G. Evensen. An ensemble kalman smoother for nonlinear dynamics. *Mon. Wea. Rev.*, 128:1852–1867, 2000.

[32] Y. Chen and D. S. Oliver. Ensemble randomized maximum likelihood method as an iterative ensemble smoother. *Math. Geosci.*, 44(1):1–26, 2012.

[33] M. Bocquet and P. Sakov. An iterative ensemble kalman smoother. *Q. J. R. Meterol. Soc.*, 140:1521–1535, 2014.

[34] D. J. Posselt, D. Hodyss, C. H. Bishop, and S. B. Pope. Errors in ensemble kalman smoother estimates of cloud microphysical parameters. *Mon. Wea. Rev.*, 142:1631–1654, 2014.

[35] G. Evensen. Analysis of iterative ensemble smoothers for solving inverse problems. *Comput. Geosci.*, 22(6):885–908, 2018.

[36] M. Zupanski. Maximum likelihood ensemble filter: Theoretical aspects. *Mon. Wea. Rev.*, 133:1710–1726, 2005.

[37] M. Asch, M. Bocquet, and M. Nodet. *Data Assimilation: Methods, Algorithms, and Applications*. SIAM, 2016.

[38] C. Grudzien and M. Boquet. A fast, single-iteration ensemble Kalman smoother for sequential data assimilation. *Geosci. Model Dev.*, 15:7641–7681, 2022.

[39] X. Gao, S. M. J. Guzik, and P. Colella. A fourth-order boundary treatment for viscous fluxes on cartesian grid finite-volume methods. AIAA 2014-1277, 52nd AIAA Aerospace Sciences Meeting, 2014.

[40] S. M. Guzik, X. Gao, L. D. Owen, P. McCorquodale, and P. Colella. A freestream-preserving fourth-order finite-volume method in mapped coordinates with adaptive-mesh refinement. *Comput. Fluids*, 123:202–217, 2015.

[41] S. M. Guzik, X. Gao, and C. Olschanowsky. A high-performance finite-volume algorithm for solving partial differential equations governing compressible viscous flows on structured grids. *Comput. Math Appl.*, 72:2098–2118, 2016.

[42] X. Gao, L. D. Owen, and S. M. J. Guzik. A parallel adaptive numerical method with generalized curvilinear coordinate transformation for compressible Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 82:664–688, 2016.

[43] L. D. Owen, S. M. Guzik, and X. Gao. A high-order adaptive algorithm for multispecies gaseous flows on mapped domains. *Comput. Fluids*, 170:249–260, 2018.

[44] L. D. Owen, X. Gao, and S. M. Guzik. Techniques for improving monotonicity in a fourth-order finite-volume algorithm solving shocks and detonations. *J. Comput. Phys.*, 415, 2020.

[45] N. Overton-Katz. *Geometry Considerations for High-Order Finite-Volume Methods on Structured Grids with Adaptive Mesh Refinement*. Ph.D. dissertation, Colorado State University, 2022.

[46] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, and M. Visbal. High-order CFD methods: current status and perspective. *Int. J. Numer. Methods Fluids*, 72(8):811–845, 2013.

[47] L. D. Owen. *A Fourth-Order Solution-Adaptive Finite-Volume Algorithm for Compressible Reacting Flows on Mapped Domains*. Ph.D. dissertation, Colorado State University, 2019.

[48] C. Kennedy and M. Carpenter. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Appl. Numer. Math.*, 44:139–181, 2003.

[49] J. Christopher, S. M. Guzik, and X. Gao. High-order implicit-explicit additive runge–kutta schemes for numerical combustion with adaptive mesh refinement. *Int. J. Numer. Methods Fluids*, 94(8):1082–1110, 2022.

[50] M. Jardak, I. M. Navon, and M. Zupanski. Comparison of sequential data assimilation methods for the Kuramoto–Sivashinsky equation. *International Journal for Numerical Methods in Fluids*, 62(4):374–402, December 2010.

[51] E. N. Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1. Shinfield Park, Reading, ECMWF, 1996.

[52] R. Menard, S. E. Cohn, L.-P. Chang, and P. M. Lyster. Assimilation of stratospheric chemical tracer observations using a kalman filter. part i: Formulation. *Mon. Wea. Rev.*, 128:2654–2671, 2000.

[53] G. I. Sivashinsky. Nonlinear analysis of hydrodynamic instability in laminar flames–i. derivation of basic equations. *Acta Astronomica*, 4:1177–1206, 1977.

[54] Y. Kuramoto. Diffusion-induced chaos in reaction systems. *Supplement of the Progress of Theoretical Physics*, (64):346–367, 1978.

[55] J. S. Whitaker and A. F. Loughe. The relationship between ensemble spread and ensemble mean skill. *Mon. Wea. Rev.*, 126:3292–3302, 1998.

[56] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.*, 27(1):1–31, 1978.

[57] T. Houba, R. Munipalli, and M. E. Harvazinski. Development of data assimilation methods for combustion. In *AIAA SCITECH 2023 Forum*, page 1281, 2023.

[58] T. H. Pulliam and D. W. Zingg. *Fundamental Algorithms in Computational Fluid Dynamics*. Scientific Computation. Spinger International Publishing, New Delhi, India, 1 edition, 2014.

[59] J. I. Polanco. BSplineKit.jl: a collection of B-spline tools in Julia, 2021.

[60] P. McCorquodale and P. Colella. A high-order finite-volume method for conservation laws on locally refined grids. *Comm. App. Math. Comput. Sci.*, 6(1):1–25, 2011.

[61] SK Godunov. A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb.*, 47:271, 1959.

Appendix A

Gradient and Hessian Derivations

A.1 MLEF

$\boldsymbol{\nabla}_{\psi} J(\boldsymbol{\psi})$ derivation

The following will show the rigorous derivation of the gradient for the MLEF in terms of ψ. Given the costfunction

$$J(\boldsymbol{\psi}) = \frac{1}{2}\boldsymbol{\psi}^{\mathrm{T}}\boldsymbol{\psi} + \frac{1}{2}\Big[\mathbf{O} - \mathcal{H}(\mathbf{Q}_f + \mathbf{X}_f\boldsymbol{\psi})\Big]^{\mathrm{T}}\mathbf{R}^{-1}\Big[\mathbf{O} - \mathcal{H}(\mathbf{Q}_f + \mathbf{X}_f\boldsymbol{\psi})\Big], \qquad \text{(A.1)}$$

we take the gradient w.r.t. $\boldsymbol{\psi}$. We focus on the two terms individually.

term 1, $\frac{1}{2}\boldsymbol{\psi}^{\mathbf{T}}\boldsymbol{\psi}$

Starting with the first term and using the definitions of gradients of matrices,

$$\frac{1}{2}\frac{\partial \boldsymbol{\psi}^{\mathrm{T}}\mathbf{I}\boldsymbol{\psi}}{\partial \boldsymbol{\psi}} = \big[2\boldsymbol{\psi}^{\mathrm{T}}\mathbf{I}\big] = \boldsymbol{\psi}^{\mathrm{T}} \qquad \text{(A.2)}$$

term 2, $\frac{1}{2}\Big[\mathbf{O} - \mathcal{H}(\mathbf{Q}_f + \mathbf{X}_f\boldsymbol{\psi})\Big]^{\mathbf{T}}\mathbf{R}^{-1}\Big[\mathbf{O} - \mathcal{H}(\hat{\mathbf{Q}}_f + \mathbf{X}_f\boldsymbol{\psi})\Big]$

By using the matrix derivative rule,

$$\frac{\partial f}{\partial \boldsymbol{\psi}} = \mathbf{h}^T(\mathbf{A} + \mathbf{A}^{\mathrm{T}})\frac{\partial \mathbf{h}}{\partial \boldsymbol{\psi}}, \qquad \text{(A.3)}$$

where $\mathbf{h}$ is a function of $\boldsymbol{\psi}$. Letting

$$f = \frac{1}{2}\Big[\mathbf{O} - \mathcal{H}(\mathbf{Q}_f + \mathbf{X}_f\boldsymbol{\psi})\Big]^{\mathrm{T}}\mathbf{R}^{-1}\Big[\mathbf{O} - \mathcal{H}(\hat{\mathbf{Q}}_f + \mathbf{X}_f\boldsymbol{\psi})\Big] \qquad \text{(A.4)}$$

and

$$\mathbf{h}(\boldsymbol{\psi}) = \mathbf{O} - \mathcal{H}(\mathbf{Q}_f + \mathbf{X}_f \boldsymbol{\psi}) \tag{A.5}$$

and

$$\mathbf{R}^{-1} = \mathbf{A} \tag{A.6}$$

we need to find $\frac{\partial \mathbf{h}}{\partial \boldsymbol{\psi}}$. Let

$$\mathbf{H}_{\psi} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{\psi}} \tag{A.7}$$

and we get

$$\frac{\partial \mathbf{h}}{\partial \boldsymbol{\psi}} = -\mathbf{H}_{\psi} \mathbf{X}_f. \tag{A.8}$$

We get, for term 2,

$$-\frac{1}{2} \left[\mathbf{O} - \mathcal{H}(\mathbf{Q}_f + \mathbf{X}_f \boldsymbol{\psi})\right]^{\mathrm{T}} (\mathbf{R}^{-1} + \mathbf{R}^{-\mathrm{T}}) \mathbf{H}_{\psi} \mathbf{X}_f = -\left[\mathbf{O} - \mathcal{H}(\mathbf{Q}_f + \mathbf{X}_f \boldsymbol{\psi})\right]^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{H}_{\psi} \mathbf{X}_f. \tag{A.9}$$

Here we assume $\mathbf{R}^{-1}$ is symmetric. Rearrange and take the transpose to get

$$-\left(\mathbf{R}^{-1/2} \mathbf{H}_{\psi} \mathbf{X}_f\right)^{\mathrm{T}} \mathbf{R}^{-1/2} \left[\mathbf{O} - \mathcal{H}(\mathbf{Q}_f + \mathbf{X}_f \boldsymbol{\psi})\right]. \tag{A.10}$$

term 1 + term 2

Putting the two terms together, the result is (we take the transpose of the first term to assure a column vector):

$$\boldsymbol{\nabla}_{\psi} J = \boldsymbol{\psi} - \left(\mathbf{R}^{-1/2} \mathbf{H}_{\psi} \mathbf{X}_f\right)^{\mathrm{T}} \mathbf{R}^{-1/2} \left[\mathbf{O} - \mathcal{H}(\mathbf{Q}_f + \mathbf{X}_f \boldsymbol{\psi})\right] . \tag{A.11}$$

$\nabla_\psi^2 J(\psi)$ derivation

Now for the Hessian. We take the gradient of $\nabla_{\boldsymbol{\psi}} J(\psi)$ w.r.t. $\boldsymbol{\psi}$.

term 1, ψ

This is trivial, we get $\mathbf{I}$.

term 2, $\left(\mathbf{R}^{-1/2}\mathbf{H}_\psi \mathbf{X}_f\right)^{\mathbf{T}} \mathbf{R}^{-1/2}\left[\mathbf{O}-\mathcal{H}(\mathbf{Q}_f+\mathbf{X}_f\boldsymbol{\psi})\right]$

Apply the chain rule:

$$\frac{\partial \mathbf{f}}{\partial \boldsymbol{\psi}} = \frac{\partial \mathbf{f}}{\partial \mathbf{h}}\frac{\partial \mathbf{h}}{\partial \boldsymbol{\psi}} \tag{A.12}$$

where $\mathbf{h}$ is the same as before, and $\mathbf{f}$ is term 2. We get

$$\frac{\partial \mathbf{f}}{\partial \mathbf{h}} = \left(\mathbf{R}^{-1/2}\mathbf{H}_{\boldsymbol{\psi}}\mathbf{X}_f\right)^{\mathrm{T}} \mathbf{R}^{-1/2} \tag{A.13}$$

and

$$\frac{\partial \mathbf{h}}{\partial \boldsymbol{\psi}} = -\mathbf{H}_{\boldsymbol{\psi}}\mathbf{X}_f\,. \tag{A.14}$$

term 1 + term 2

Bringing the two terms together, the result is

$$\nabla_{\boldsymbol{\psi}}^2 J = \mathbf{I} + \left(\mathbf{R}^{-1/2}\mathbf{H}_{\boldsymbol{\psi}}\mathbf{X}_f\right)^{\mathrm{T}} \mathbf{R}^{-1/2}\mathbf{H}_{\boldsymbol{\psi}}\mathbf{X}_f\,. \tag{A.15}$$

Second change of variable

We now introduce the second change of variable:

$$\boldsymbol{\psi} = \left(\mathbf{I} + \mathbf{C}_f\right)^{-\mathrm{T}/2}\boldsymbol{\xi}\,, \tag{A.16}$$

where

$$\mathbf{C}_f = \mathbf{Z}_f^{\mathrm{T}}\mathbf{Z}_f \tag{A.17}$$

and the columns of $\mathbf{Z}_f$ are

$$\mathbf{Z}_{f,i} = (\mathbf{R}^{-1/2}\mathbf{H}_{\psi}\mathbf{X}_f)_i \,. \tag{A.18}$$

Note that $\mathbf{Z}_f$ and $\mathbf{Z}$ are different and the difference will be explained later. The new cost function is

$$J(\boldsymbol{\xi}) = \frac{1}{2}\left(\left(\mathbf{I}+\mathbf{C}_f\right)^{-\mathrm{T}/2}\boldsymbol{\xi}\right)^T\left(\left(\mathbf{I}+\mathbf{C}_f\right)^{-\mathrm{T}/2}\boldsymbol{\xi}\right) \tag{A.19}$$

$$+\frac{1}{2}\left[\mathbf{O}-\mathcal{H}(\mathbf{Q}_f+\mathbf{X}_f\left(\mathbf{I}+\mathbf{C}_f\right)^{-\mathrm{T}/2}\boldsymbol{\xi})\right]^{\mathrm{T}}\mathbf{R}^{-1}\left[\mathbf{O}-\mathcal{H}(\mathbf{Q}_f+\mathbf{X}_f\left(\mathbf{I}+\mathbf{C}_f\right)^{-\mathrm{T}/2}\boldsymbol{\xi})\right]. \tag{A.20}$$

$\boldsymbol{\nabla}_{\boldsymbol{\xi}}J(\boldsymbol{\xi})$ derivation

Finding the gradient is straightforward. We can use the chain rule:

$$\frac{\partial J(\boldsymbol{\xi})}{\partial\boldsymbol{\xi}} = \frac{\partial J(\boldsymbol{\xi})}{\partial\boldsymbol{\psi}}\frac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{\xi}} \,. \tag{A.21}$$

If, in the term $\frac{\partial J(\boldsymbol{\xi})}{\partial\boldsymbol{\psi}}$, we replace $\boldsymbol{\xi}$ with $\boldsymbol{\psi}$ in $J(\boldsymbol{\xi})$, we simply get $\boldsymbol{\nabla}_{\boldsymbol{\psi}}J(\boldsymbol{\psi})$. For the other partial derivative,

$$\frac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{\xi}} = \left(\mathbf{I}+\mathbf{C}_f\right)^{-\mathrm{T}/2} \,. \tag{A.22}$$

The result is

$$
\begin{aligned}
\boldsymbol{\nabla}_{\boldsymbol{\xi}} J &= (\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}/2}\, \boldsymbol{\nabla}_{\psi} J(\boldsymbol{\psi}) && \text{(A.23)}\\
&= (\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}/2} \left[\boldsymbol{\psi} - \left(\mathbf{R}^{-1/2}\mathbf{H}_{\psi}\mathbf{X}_f\right)^{\mathrm{T}} \mathbf{R}^{-1/2} \left[\mathbf{O}-\mathcal{H}(\mathbf{Q}_f+\mathbf{X}_f\boldsymbol{\psi})\right]\right] && \text{(A.24)}\\
&= (\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}}\, \boldsymbol{\xi} - (\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}/2} \left(\mathbf{R}^{-1/2}\mathbf{H}_{\psi}\mathbf{X}_f\right)^{\mathrm{T}} \mathbf{R}^{-1/2} \left[\mathbf{O}-\mathcal{H}(\mathbf{Q}_f+\mathbf{X}_f\,(\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}/2}\,\boldsymbol{\xi})\right] && \text{(A.25)}\\
&= (\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}}\, \boldsymbol{\xi} - (\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}/2}\, \mathbf{Z}^{\mathrm{T}}\mathbf{R}^{-1/2} \left[\mathbf{O}-\mathcal{H}(\mathbf{Q}_f+\mathbf{X}_f\,(\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}/2}\,\boldsymbol{\xi})\right] && \text{(A.26)}\\
&= (\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}}\, \boldsymbol{\xi} - (\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}/2}\, \mathbf{Z}^{\mathrm{T}}\mathbf{R}^{-1/2} \left[\mathbf{O}-\mathcal{H}(\hat{\mathbf{Q}})\right]\,. && \text{(A.27)}
\end{aligned}
$$

Note that we purposefully don't use $\mathbf{Z}_f$, but rather $\mathbf{Z}$, but we continue to use $\mathbf{C}_f$.

$\boldsymbol{\nabla}^2_{\boldsymbol{\xi}} J(\boldsymbol{\xi})$ derivation

The Hessian is dealt with in a similar way.

$$
\frac{\partial\left[\boldsymbol{\nabla}_{\boldsymbol{\xi}} J\right]}{\partial\boldsymbol{\xi}} = \frac{\partial\left[\boldsymbol{\nabla}_{\boldsymbol{\xi}} J\right]}{\partial\boldsymbol{\psi}}\frac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{\xi}}. \tag{A.28}
$$

where

$$
\frac{\partial\left[\boldsymbol{\nabla}_{\boldsymbol{\xi}} J\right]}{\partial\boldsymbol{\psi}} = \frac{\partial\left[(\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}/2}\, \boldsymbol{\nabla}_{\psi} J(\boldsymbol{\psi})\right]}{\partial\boldsymbol{\psi}} = (\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}/2}\, \boldsymbol{\nabla}^2_{\psi} J(\boldsymbol{\psi}) \tag{A.29}
$$

and

$$
\frac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{\xi}} = (\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}/2}\,. \tag{A.30}
$$

And so we have

$$
\boldsymbol{\nabla}^2_{\boldsymbol{\xi}} J(\boldsymbol{\xi}) = (\mathbf{I}+\mathbf{C}_f)^{-\mathrm{T}/2}\, \boldsymbol{\nabla}^2_{\psi} J(\boldsymbol{\psi})\, (\mathbf{I}+\mathbf{C}_f)^{-1/2} \tag{A.31}
$$

Approximating $\mathbf{R}^{-1/2}\mathbf{H}\mathbf{X}_{f,i}$

We don't want to explicitly use the tangent linear of $\mathbf{H}$. We can get around this by expanding $\mathbf{H}\mathbf{X}_{f,i}$ using a Taylor series:

$$\mathcal{H}\left(\bar{\mathbf{Q}} + \frac{1}{\sqrt{N_e - 1}}(\mathbf{Q}_{f,i} - \bar{\mathbf{Q}})\right) = \mathcal{H}(\bar{\mathbf{Q}}) + \mathbf{H}\frac{1}{\sqrt{N_e - 1}}(\mathbf{Q}_{f,i} - \bar{\mathbf{Q}}) + (\mathcal{O})^2\left(\frac{1}{\sqrt{N_e - 1}}(\mathbf{Q}_{f,i} - \bar{\mathbf{Q}})\right) \tag{A.32}$$

Notice $\mathbf{X}_f$ can be substituted, and so we get

$$\mathbf{H}\mathbf{X}_{f,i} \approx \mathcal{H}(\bar{\mathbf{Q}} + \mathbf{X}_{f,i}) - \mathcal{H}(\bar{\mathbf{Q}})\,. \tag{A.33}$$

If we say, instead of using the normalizing constant of $\frac{1}{\sqrt{N_e-1}}$, we can actually reduce the error by using a small value $\epsilon << 1$. Now we have $\epsilon(\mathbf{Q}_{f,i} - \bar{\mathbf{Q}})$ to use in the expansion, and we get

$$\mathbf{H}\epsilon\mathbf{X}_{f,i} \approx \mathcal{H}(\bar{\mathbf{Q}} + \epsilon\mathbf{X}_{f,i}) - \mathcal{H}(\bar{\mathbf{Q}}) \tag{A.34}$$

$$\mathbf{H}\mathbf{X}_{f,i} \approx \frac{1}{\epsilon}\left[\mathcal{H}(\bar{\mathbf{Q}} + \epsilon\mathbf{X}_{f,i}) - \mathcal{H}(\bar{\mathbf{Q}})\right]\,. \tag{A.35}$$

Note we can technically lump $\frac{1}{\sqrt{N_e-1}}$ into the ϵ constant and thus is not shown.

C

There is a deliberate difference between $\mathbf{Z}_f$ and $\mathbf{Z}$. The columns of $\mathbf{Z}$ are redefined as

$$\mathbf{Z}_i = \mathbf{R}^{-1/2}\frac{1}{\epsilon}\left[\mathcal{H}(\bar{\mathbf{Q}} + \epsilon\mathbf{X}_{f,i}) - \mathcal{H}(\bar{\mathbf{Q}})\right]\,, \tag{A.36}$$

which also changes $\mathbf{C}$. The columns of $\mathbf{Z}_f$ are

$$\mathbf{Z}_{f,i} = \mathbf{R}^{-1/2}\frac{1}{\epsilon}\left[\mathcal{H}(\bar{\mathbf{Q}}_f + \epsilon\mathbf{X}_{f,i}) - \mathcal{H}(\bar{\mathbf{Q}}_f)\right]\,, \tag{A.37}$$

which also changes $\mathbf{C}_f$. So in the equations, when we see $\mathbf{C}_f$, we use the appropriate forecast control.

$\boldsymbol{\nabla}_{\psi} J(\boldsymbol{\psi})$ and $\boldsymbol{\nabla}^2_{\psi} J(\boldsymbol{\psi})$

We need to redefine $\boldsymbol{\nabla}_{\psi} J(\boldsymbol{\psi})$ and $\boldsymbol{\nabla}^2_{\psi} J(\boldsymbol{\psi})$ in terms of this new $\mathbf{Z}$. They are

$$\begin{aligned} \boldsymbol{\nabla}_{\psi} J(\boldsymbol{\psi}) &= \boldsymbol{\psi} - \mathbf{Z}^{\mathrm{T}} \mathbf{R}^{-1/2} \left[\mathbf{O} - \mathcal{H}(\mathbf{Q}_f + \mathbf{X}_f \boldsymbol{\psi})\right] && \text{(A.38)} \\ &= \boldsymbol{\psi} - \mathbf{Z}^{\mathrm{T}} \mathbf{R}^{-1/2} \left[\mathbf{O} - \mathcal{H}(\bar{\mathbf{Q}})\right] , && \text{(A.39)} \end{aligned}$$

and

$$\boldsymbol{\nabla}^2_{\psi} J = \mathbf{I} + \mathbf{Z}^{\mathrm{T}} \mathbf{Z} . \tag{A.40}$$

$\boldsymbol{\nabla}_{\xi} J(\boldsymbol{\xi})$ and $\boldsymbol{\nabla}^2_{\xi} J(\boldsymbol{\xi})$

These don't change if you calculate them using

$$\boldsymbol{\nabla}_{\xi} J = (\mathbf{I} + \mathbf{C}_f)^{-\mathrm{T}/2} \, \boldsymbol{\nabla}_{\psi} J(\boldsymbol{\psi}) . \tag{A.41}$$

and

$$\boldsymbol{\nabla}^2_{\xi} J(\boldsymbol{\xi}) = (\mathbf{I} + \mathbf{C}_f)^{-\mathrm{T}/2} \, \boldsymbol{\nabla}^2_{\psi} J(\boldsymbol{\psi}) \, (\mathbf{I} + \mathbf{C}_f)^{-1/2} \tag{A.42}$$

A.2 MLES

For the MLES, we now have the model in the cost function. If we let

$$\mathcal{F}_{0:k}(\cdot) = \mathcal{H}(\mathcal{M}_{0:k}(\cdot)) , \tag{A.43}$$

the derivations for the gradients and Hessians are similar to that of the MLEF:

$$\boldsymbol{\nabla}_{\psi} J = \boldsymbol{\psi} - \sum_{k=L-S+1}^{L} \mathbf{Z}_k^{\mathrm{T}} \mathbf{R}_k^{-1/2} \left[\mathbf{O}_k - \mathcal{F}_{0:k}(\bar{\mathbf{Q}}_0) \right] . \tag{A.44}$$

$$\boldsymbol{\nabla}_{\boldsymbol{\psi}}^2 J = \mathbf{I} + \sum_{k=L-S+1}^{L} \mathbf{Z}_k^{\mathrm{T}} \mathbf{Z}_k . \tag{A.45}$$

$$\boldsymbol{\nabla}_{\boldsymbol{\xi}} J = \left(\mathbf{I} + \mathbf{C}_f\right)^{-\mathrm{T}/2} \boldsymbol{\nabla}_{\boldsymbol{\psi}} J \tag{A.46}$$

$$= \left(\mathbf{I} + \mathbf{C}_f\right)^{-\mathrm{T}} \boldsymbol{\xi} - \left(\mathbf{I} + \mathbf{C}_f\right)^{-\mathrm{T}/2} \sum_{k=L-S+1}^{L} \mathbf{Z}_k^{\mathrm{T}} \mathbf{R}_k^{-1/2} \left[\mathbf{O}_k - \mathcal{F}_{0:k}(\hat{\mathbf{Q}}_0) \right] . \tag{A.47}$$

$$\boldsymbol{\nabla}_{\boldsymbol{\xi}}^2 J(\boldsymbol{\xi}) = \left(\mathbf{I} + \mathbf{C}_f\right)^{-\mathrm{T}/2} \boldsymbol{\nabla}_{\boldsymbol{\psi}}^2 J(\boldsymbol{\psi}) \left(\mathbf{I} + \mathbf{C}_f\right)^{-1/2} . \tag{A.48}$$

The time index k refers to the observation locations, and running the forward model (i.e. $\mathcal{M}_{0:k}(\cdot)$) represents as such. Redefining $\mathbf{Z}$ and $\mathbf{C}$:

$$\mathbf{Z}_{k,i} = \mathbf{R}_k^{-1/2} \frac{1}{\epsilon} \left[\mathcal{F}_{0:k}(\bar{\mathbf{Q}}_0 + \epsilon \mathbf{X}_{f,i}) - \mathcal{F}_{0:k}(\bar{\mathbf{Q}}_0) \right] , \tag{A.49}$$

and

$$\mathbf{C} = \sum_{k=L-S+1}^{L} \mathbf{Z}_k^{\mathrm{T}} \mathbf{Z}_k , \tag{A.50}$$

and

$$\mathbf{C}_f = \sum_{k=L-S+1}^{L} \mathbf{Z}_{k,f}^{\mathrm{T}} \mathbf{Z}_{k,f} , \tag{A.51}$$

with

$$\mathbf{Z}_{k,f,i} = \mathbf{R}_k^{-1/2}\frac{1}{\epsilon}\left[\mathcal{F}_{0:k}(\bar{\mathbf{Q}}_{f,0} + \epsilon\mathbf{X}_{f,i}) - \mathcal{F}_{0:k}(\bar{\mathbf{Q}}_{f,0})\right] , \tag{A.52}$$

Again, this is only calculated once at the first minimization iteration, and updated when the optimal state value is found to reinitialize the ensemble.

A.3 Additional Minimization details

We need to calculate the gradient and Hessian in terms of ψ. There are some things to consider when performing minimization.

1. Only calculate $\mathbf{C}_f$ in the *first* minimization iteration using the forecast value, and use that value for the rest of the minimization iterations. When there are $\mathbf{Z}$ terms (i.e. with no subscript f), you must calculate those differently for each iteration. For $\boldsymbol{\nabla}_{\boldsymbol{\xi}} J$, there is $\mathbf{Z}^{\mathrm{T}}$ that must be updated, but the $\mathbf{C}_f$ terms need only calculating once. For $\boldsymbol{\nabla}^2_{\boldsymbol{\xi}} J$, we get

$$\boldsymbol{\nabla}^2_{\boldsymbol{\xi}} J = (\mathbf{I} + \mathbf{C}_f)^{-\mathrm{T}/2}\left(\mathbf{I} + \mathbf{Z}^{\mathrm{T}}\mathbf{Z}\right)(\mathbf{I} + \mathbf{C}_f)^{-1/2} . \tag{A.53}$$

   The Hessian would be the identity which is only be valid when $\mathcal{H}$ is linear. We must use $\mathbf{Z}$.

2. At the end of minimization, we must re-calculate $\mathbf{C}$ (call it $\mathbf{C}_a$) using the final, updated $\bar{\mathbf{Q}}_a$. To update the ensemble, we need

$$\mathbf{X}_a = \sqrt{N_e - 1}\mathbf{X}_f\left(\mathbf{I} + \mathbf{C}_a\right)^{-\mathrm{T}/2} \tag{A.54}$$

   note the $\sqrt{N_e - 1}$ to cancel the original $\frac{1}{\sqrt{N_e-1}}$ on $\mathbf{X}_f$. We can now update the ensemble,

$$\mathbf{Q}_{a,i} = \bar{\mathbf{Q}}_a + \mathbf{X}_{a,i} , \tag{A.55}$$

   and continue with forecasting.