Multilevel ensemble data assimilation

A thesis submitted for the Imperial College London degree of

Doctor of Philosophy in Mathematics

by

Alastair C. A. Gregory

Department of Mathematics

Imperial College London

This work is supported by the NERC funded Science and Solutions for a Changing Planet DTP [Reference Number: NE/L002515/1]

Supervisor: Dr. Colin Cotter

I certify that this thesis, and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

Signed: Alastair Gregory

Copyright

“I very much like you have been saved so many times by a view,
yesterday the sun whispered into the moon’s ear,
and the moon trembled, turning white with fear."

– Anthony Anaxagorou –

To Alan Woods, for all the inspiration a grandson could ever need.

Acknowledgements

I humbly thank Colin Cotter and Chris Ferro, for mentoring me throughout different parts of my education, Jim Haywood, for sparking my enthusiasm for research and finally my childhood mathematics teacher, Susan Harvey. There are many others that have inspired me, and dedicated time and energy into helping me throughout my education, in particular Alex Louch, Adam Butler, Susana Gomez and my parents Claire and Richard Gregory. For this I am extremely grateful; I could not have got to the point where I am now without the people mentioned above. On a professional note, I would like to thank the Natural Environmental Research Council, Grantham Institute of Climate Change, Mathematics of Planet Earth CDT, University of Reading, University of Exeter and Imperial College London for the funding and training that has been invested in me.

Abstract

This thesis aims to investigate and improve the efficiency of ensemble transform methods for data assimilation, using an application of multilevel Monte Carlo. Multilevel Monte Carlo is an interesting framework to estimate statistics of discretized random variables, since it uses a hierarchy of discretizations with a refinement in resolution. This is in contrast to standard Monte Carlo estimators that only use a discretization at a fine resolution. A linear combination of sub-estimators, on different levels of this hierarchy, can provide new statistical estimators to random variables at the finest level of resolution with significantly greater efficiency than a standard Monte Carlo equivalent. Therefore, the extension to computing filtering estimators for data assimilation is a natural, but challenging area of study. These challenges arise due to the fact that correlation must be imparted between ensembles on adjacent levels of resolution and maintained during the assimilation of data. The methodology proposed in this thesis, considers coupling algorithms to establish this correlation. This generates multilevel estimators that significantly reduce the computational expense of propagating ensembles of discretizations through time and space, in between stages of data assimilation.

An effective benchmark of this methodology is realised by filtering data into high-dimensional spatio-temporal systems, where a high computational complexity is required to solve the underlying partial differential equations. A novel extension of an ensemble transform localisation framework to finite element approximations within random spatio-temporal systems is proposed, in addition to a multilevel equivalent.

Table of contents

List of Figures

Chapter 1

Introduction

Chapter synopsis

The atmospheric processes that govern the weather are extremely complicated and nonlinear. The complexity of them creates a challenge for the field of forecasting: to predict the future state of something, given the uncertainty in an initial state or the dynamics within the underlying system (e.g. cloud parameterizations in weather forecasting). The thesis is concerned with precisely this, and in particular the field of ensemble and empirical forecasting. This chapter explores the mathematical and statistical techniques used in this field to quantify uncertainty of future forecastable states and provides a literature review of them. This review motivates the content in the remainder of the thesis.

1.1 Overview of thesis

This thesis blends together research areas of data assimilation, statistical estimation and numerical analysis. The product of this combination is a proof of concept for an efficient type of high-dimensional[1] ensemble forecasting methodology with the incorporation of data. Ensemble forecasting is essentially a toolbox of methods to quantify the uncertainty in random systems. This technique is frequently use by operational weather forecasting centers. The premise of ensemble forecasting is inherently simple (albeit sometimes extremely computationally expensive): construct an ensemble of N different 'realisations' of the random system in question and post-process statistics from this. However, the implementation of the post-processing procedure is in contrast not simple, and is an area of major research. In addition to this, the verification of the forecasts is an important area of research. This is concentrated upon in Chapter 3.

Statistics estimated from an ensemble forecast are important in understanding the uncertainty within a system that one is attempting to forecast. This estimation process is a broad field of research in itself, the Monte Carlo method is a general framework for this. There is a great range of theoretical literature exploring the accuracy and convergence of this method. The efficiency of these estimators (defined as a ratio of their accuracy to the computational cost of constructing them) is an area of great research importance. For example, variance reduction techniques can reduce the variance and error of these estimators, increasing their efficiency. Turn to Sec. 1.2.1 for more details on this. A major development in variance reduction research is the multilevel Monte Carlo (MLMC) method, and was proposed in Giles [2008].

Multilevel Monte Carlo uses a hierarchy of discretizations (e.g. time-stepping schemes) of a random variable at different 'levels' of resolution. From these different discretizations it uses a linear combination of independent Monte Carlo estimators on each level, with sample sizes dependent on the resolution. These linear combinations can significantly increase the efficiency of statistical estimation. Multilevel Monte Carlo has been applied to a great number of applications and variants of ensemble forecasting, many of these are discussed in Sec. 1.2.2. However one of the areas of ensemble forecasting that has only recently had MLMC applied to it is the area of data assimilation.

[1]The 'dimension' being referred to in the term 'high-dimensional' is not the geometric dimension (e.g. 2D, 3D state spaces) but rather the number of degrees of freedom of the state space.

Data assimilation is the technique that blends data with forecast models, and will be the subject of interest throughout this thesis. Data assimilation allows one to refine forecasts based on data (e.g. observations) taken from a certain trajectory that is perceived to be the truth. In the context of ensemble forecasting, a data assimilation framework is empirical, forming methodologies such as filters and sequential Monte Carlo. This is discussed in more detail in Sec. 1.3.1. Empirical data assimilation schemes often struggle to deal with practical cases that consider high-dimensional systems, due to the issue known as the 'curse of dimensionality'. For more information on this turn to Sec. 1.3.3. One of the major issues surrounding empirical data assimilation schemes in this high-dimensional context is the computational cost of propagating large ensembles of model simulations in time and space. These model simulations are often discretizations, and their computational cost grows significantly as their spatial and temporal resolution increases.

The application of MLMC to empirical data assimilation schemes in this context is therefore desirable to increase their efficiency. This is the aim of the thesis. The following sections in the introduction provide a more detailed review of the topics mentioned in this overview, starting with the foundation of ensemble forecasting: statistical estimation.

1.2 Statistical estimation

As stated in the previous section, the Monte Carlo method is traditionally used for statistical estimation. The main principle of the Monte Carlo method is that one can construct estimators to expected values of statistical quantities, e.g. moments, by simply taking means of ensemble forecasts. These estimators have rigorous error analysis, with the Central Limit Theorem (CLT) providing convergence rates with the number of ensemble members, N. More light is shed on this convergence in Sec. 2.2.2.

Unfortunately this convergence rate is relatively slow, and the absolute error from that of the true statistic scales as $\mathcal{O}(N^{-1/2})$. This makes large ensemble forecasts using the Monte Carlo method for probabilistic estimates infeasible for weather forecasts, for example. Typical ensemble sizes of forecasts from operational weather forecasting centers vary from 20 - 40. The reason for the infeasibility is, as highlighted previously, computational cost. Large weather models (or what is commonly called a forward prediction model) take a long time to simulate in isolation, let alone many at a time. One can increase the efficiency of the Monte Carlo method via variance reduction techniques.

1.2.1 Variance reduction

Using small ensemble sizes for ensemble forecasts (with a feasible computational cost) needs a small variance within the estimator in order to have sufficient accuracy. Variance reduction is achieved by considering the following thought:

How should we pick the members in our ensemble (and post-process them) to construct the most optimal / efficient forecast?

As is demonstrated in Sec. 2.5, the make-up of the ensemble (such as *antithetic variates*) together with some statistical tricks (such as *control variates*) can significantly reduce the variance of these estimators. This type of statistical manipulation is employed in the multilevel Monte Carlo method.

1.2.2 Multilevel methods

The computational bottleneck in ensemble forecasting and statistical estimation stems from the fact that the high-dimensional forward prediction model is discretized in time (and space). In most cases, the temporal (and spatial) resolution of the model determines the accuracy of the discretization, as a bias from the truth is associated with a particular discretization. For accurate approximations to forecast statistics, both the discretization accuracy (dependent on the model resolution) and the sample size of the ensemble (given the convergence of Monte Carlo estimates) need to increase simultaneously. See Sec. 2.4.4 for more details on this issue. This allows one to see the large computational costs associated with these estimators when the forward model cost grows at a sufficiently large rate with increasing resolution.

Variance reduction techniques, as previously discussed, can reduce this computational cost, but only by a scalar factor, and not by an order of magnitude (proportional to the discretization accuracy or sample size). Multilevel Monte Carlo (MLMC) [Giles, 2008, Cliffe et al., 2011, Giles, 2015] does this by trading off error due to the discretization bias and that due to the sample size. This is achieved by creating independent approximations to statistics on a hierarchy of different model temporal (and spatial) resolutions, with the sample sizes of each of these approximations proportional to the discretization bias. It then combines these approximations in a linear combination, as previously mentioned. The form of this linear combination, along with the appropriate analysis, is given in depth in Sec. 2.6.

Applications

It is the simple linear combination of independent Monte Carlo approximations that gives the MLMC framework the flexibility to work in many applications. Density estimation [Giles et al., 2015], quasi-Monte Carlo [Giles and Waterhouse, 2009], Sequential Monte Carlo methods [Beskos et al., 2017, Jasra et al., 2017a, Moral et al., 2017], ensemble Kalman filtering [Hoel et al., 2016, Chernov et al., 2016], Markov chain Monte Carlo [Ketelsen et al., 2013], atmospheric dispersion modelling [Katsiolides et al., 2018, Cook, 2013] and ensemble transform methods for Bayesian inference [Gregory et al., 2016, Gregory and Cotter, 2017b] have all had MLMC applied to them; the latter of these being the topic of this thesis. A great deal of the applications listed above, are discussed in the review article, Jasra et al. [2017b]. The adaptation of MLMC to sequential Monte Carlo and ensemble transform methods for Bayesian inference require coupling of ensembles from different model resolutions during resampling or transformation schemes [Sen et al., 2018]. This is a particularly challenging hurdle to overcome in some cases. The next section provides more detail on the last of these applications.

There is also a great opportunity for optimality and adaptivity within MLMC, given the range of parameters that appear in the framework, such as the sample sizes for each independent approximation. This optimality and adaptivity can control how efficient the method is; such examples are Pauli and Arbenz [2015] and Hoel et al. [2012]. Over the last few years, there has been increased interest in multilevel methods for spatio-temporal systems; multi-index Monte Carlo [Haji-Ali et al., 2016] has been introduced as the framework to describe this case. Here, model spatial and temporal resolution are part of hierarchies that can be refined independently of one another at different rates.

1.3 Incorporating data into ensemble forecasts

Ensemble forecasts can become more powerful when they are combined with data. Data assimilation schemes provide the ability to do this, and their approaches vary greatly, from linear (assuming Guassianity of the data and forecast distribution) to non-parametric methods. Examples of data assimilation tools used by weather forecasting practitioners are Variational methods (e.g. 3DVar and 4DVar) as well as the famous ensemble Kalman filter (EnKF). For more information on these, turn to Reich and Cotter [2015], Daley [1997].

Many modern day data assimilation techniques work by transforming ensemble forecasts into another ensemble which is closer to the observations. The EnKF uses a linear transform from one to the other, and approximates the posterior mean and covariance from this new ensemble (and thus assumes Gaussianity, although mean-field convergence has been investigated [Law et al., 2016] for nonlinear systems). Bayesian approaches to data assimilation are another major strand of literature, as explained in the next section.

1.3.1 Bayesian approach to data assimilation

Bayesian inference is one of the most highly-researched topics in statistics, and is centered around the famous Bayes' rule:

The probability of X conditioned on Y (posterior) is proportional to the product of the probability of Y conditioned on X (likelihood) and the prior probability of X.

Bayes' rule provides a very useful framework in data assimilation, especially where one is assimilating into a nonlinear system. By interpreting the prior probability mentioned above as the forecast distribution before assimilation of data, and taking Y as the data then one can find the posterior probability of the forecast conditioned on the assimilated data. When applied to ensemble forecasting, one can find likelihoods for each ensemble member and employ self-normalized importance sampling to compute this posterior distribution [Doucet and Johansen, 2011]. This is done practically by giving each ensemble member an *importance weight* dependent on their likelihood. The property of the Bayesian interpretation which makes it an attractive framework for data assimilation is the sequential nature of it:

Yesterday's posterior becomes today's prior [Rohde, 2014].

This allows one to sequentially assimilate data into the forecast in real-time as data becomes available, updating the forecast on-the-go. This is known as filtering. Given that one has a particle approximation to the posterior forecast distribution at any one time, it is a simple extension to compute posterior statistical estimates from this. Bearing this in mind, it is no surprise that the name 'sequential Monte Carlo' has been given to a whole host of algorithms that approach data assimilation in this manner. It is also no surprise that it is an attractive application of multilevel Monte Carlo, as highlighted previously in Sec. 1.2.2.

Sequential Monte Carlo and the particle filter

Particle filtering is one of the broader headings for sequential Monte Carlo algorithms. Resampling of the ensemble forecast is used occasionally after the assimilation of data to stop the importance weights of each ensemble member from becoming degenerate [2] [Douc et al., 2005, Cappé et al., 2007]. This is used to replace 'unlikely' ensemble members with more likely ensemble members, to even out weights. Theoretical work has come a long way in this field: Chopin [2004] provided a major step in this direction by establishing a CLT for sequential Monte Carlo methods.

Random resampling adds variance to estimators of posterior statistics in a particle filter [Hol et al., 2006]. Thus the quality of a particle filter is strongly dependent on the choice of resampling scheme and the frequency of resampling. For example, a technique known as residual resampling tends to perform better than the most commonly used multinomial resampling; it guarantees that ensemble members with high weights get resampled a specific amount of times rather than leaving this up to chance. More information about these resampling schemes is given in Sec. 2.7.3.

Couplings and optimal transport

Resampling between a weighted ensemble and an evenly weighted one can be interpreted as a coupling between two discrete distributions. Whilst there are many possible couplings between these two discrete distributions (multinomial resampling can be seen as an independent coupling, for example), some are more important than others in the field of Bayesian data assimilation.

An optimal coupling is often described as one that solves the Monge-Kantorovich problem [Villani, 2008]. This minimizes the Wasserstein metric (expected distance) between two distributions whilst satisfying the marginal constraints of each distribution. There are many algorithms to solve such a problem for discrete distributions, including the well-known Fast Earth Mover's Distance (Fast EMD) algorithm [Pele and Werman, 2009]. When both discrete distributions have even importance weights, this becomes an assignment problem and one can solve this using the Hungarian algorithm [Munkres, 1957]. By solving this problem and using this coupling, it can be shown that the covariance between the two discrete distri-

[2] One ensemble member will be associated with a very large weight whilst the others all are associated with negligible weights. This has found to lead to collapsing particle filters.

butions is maximized [Reich, 2013]. This is beneficial for resampling in a particle filter. The ensemble transform particle filter (ETPF) is based on this particular type of coupling, and is a variant of the class of linear ensemble transform filters discussed in the next section.

1.3.2 Linear ensemble transform methods

Linear ensemble transform methods are an ever-increasingly popular set of Bayesian based filtering algorithms. The class includes the ensemble Kalman filter (EnKF), the ensemble square root filter (ESRF), and most importantly to this thesis, the ETPF [Reich and Cotter, 2015, Reich, 2013]. It takes up a very interesting position, occupying an area between the EnKF and the classical particle filter.

The ETPF uses a deterministic transform instead of the random resampling scheme. The matrix that implements this deterministic transform is found by solving the aforementioned optimal transport problem. In this case, one tries to minimize the expected distance between the 'forecast' ensemble with associated importance weights and an evenly weighted 'analysis' ensemble. This creates an optimal coupling between forecast and analysis ensembles. Section 2.7.4 provides an in-depth look at the ETPF. Chustagulprom et al. [2016] proposes a variant of the method, which builds a hybrid bridging between the EnKF and the ETPF. This can be changed adaptively depending on how Gaussian the statistics are.

Weak convergence is established for the ETPF in Reich [2013], using properties of the optimal coupling between forecast and analysis ensembles. There is still some way to go, however, before the method possesses the theoretical rigor that was shown in Chopin [2004] for the classical particle filter. Unfortunately, this is primarily due to the optimal transport problem within the algorithm, which has great appeal in a practical sense, but less so in a theoretical one. The use of optimal transportation problems within the ETPF has another drawback: the computational cost of solving them. The next section discusses this aspect.

Fast algorithms to solve optimal transport problems

In one-dimensional cases the optimal transport problems within the ETPF can be solved very rapidly using a cheap algorithm. However, in cases of dimension greater than one, the computational cost increases rapidly with the number of samples in the ensemble [Reich and Cotter, 2015]. In cases where the sample size is great, and the forward prediction model

is cheap to simulate, then this computational cost can dominate that of the propagation of ensemble members. This leads to an inefficient filter. Importantly to the aims of this thesis, it also means that an application of MLMC to the ETPF, that can reduce the computational cost of this ensemble propagation, would be unnecessary. Localisation, which is detailed in Sec. 2.7.7, can be used to reduce the complexity of these optimal transport problems and alleviate this issue.

In addition to localisation, there has been research into using fast, iterative algorithms to solve optimal transport problems, namely the Sinkhorn iterative algorithm [Cuturi, 2013], within the ETPF. This algorithm works by finding an approximation to the solution of a optimal transport problem (see Sec. 2.7.5), which is sufficient for many applications. However, the error in this approximation causes the ETPF to underestimate the analysis ensemble variance. A solution to this problem is presented in de Wiljes et al. [2016], where a second-order accurate linear transform is proposed. This preservation allows the Sinkhorn iterative algorithm to be used, and is briefly explained in Sec. 2.7.6.

1.3.3 The curse of dimensionality

Whilst there is still a constant stream of research into modifying the Bayesian approach to data assimilation, covering particle filters and linear ensemble transform methods, one of the most important directions for future research is applying the approach to high-dimensional cases. This is not a trivial extension by any means. With increasing dimension, the likelihood becomes more degenerate, and the ensemble size needed to stabilize the particle filter at each assimilation step (after resampling) becomes exponentially great [Rebeschini and Van Handel, 2013].

Despite this, it is of great importance to extend these approaches to high-dimensional cases. They would provide far more flexibility in nonlinear forecast systems than the ones previously mentioned that are used by weather forecast practitioners, such as the EnKF and Variational methods. There have been two key developments in this field, Cheng and Reich [2013] and Poterjoy [2016], that concentrate on localising the likelihood associated with an ensemble forecast member.

This type of localisation has been inspired by the success of a similar tool used in the EnKF [Anderson, 2012, Bergemann and Reich, 2010]. It tries to alleviate the curse of dimensionality by only allowing likelihoods at a given point in space to be influenced only by

points local to the given point. Consider the following:

On any given day, the observed pointwise weather in Stanford University, CA, is unlikely to reduce the uncertainty of the weather forecast in Imperial College London, U.K.

This idea is intuitive, so why not apply it to data assimilation? The likelihood of the observed pointwise weather at Imperial College London, Y, given forecasted weather X, is unlikely to be significantly altered by X in California. A naive approach to localisation would be to then assign a likelihood to each individual component of a forecast state space. However, in a classical particle filter implementation, this would cause discontinuities in the resampled forecast. This idea therefore lacks the spatial regularity to work amongst these sequential Monte Carlo methods.

However, linear ensemble transform methods have been found to incorporate this type of localisation successfully, due to the linear transform that it uses. This transform, as supposed to the resampling step in classical particle filters, offers sufficient spatial regularity to localise the likelihood around each local component of a forecast state space, and alleviate the curse of dimensionality. A proof of concept of this localisation framework for the ETPF was given in Cheng and Reich [2013]. The localisation used here can also be employed to reduce the computational cost of the optimal transport problems in some cases as already mentioned. Both aspects of the localisation used in the ETPF are described in more detail in Sec. 2.7.7. Localisation is not the only methodology that has been considered when trying to extend Bayesian approaches to spatio-temporal data assimilation. Another key development, Beskos et al. [2014], attempts to use the structure of the likelihood (product of marginal component likelihoods) to filter sequentially in space as well as time (and resample when needed).

When considering these high-dimensional data assimilation problems that one would employ localisation for, it is important to remember the computational expense of the forecasting process. Of course, this problem is encountered in standard ensemble forecasting (without the assimilation of data) as previously mentioned. This inspires the main aim of this thesis: establishing an application of MLMC to filtering algorithms in order to alleviate the high computational cost associated with the forward model used by them and increase their efficiency. Towards the end of the thesis, a proof of concept is designed by extending the localisation strategy in Cheng and Reich [2013] to finite element approximations, in order to see the benefits of applying MLMC to data assimilation problems in high-dimensional

systems of partial differential equations.

1.4 Outlook

The remainder of this thesis is organised as follows. Chapter 2 will survey many of the aforementioned techniques, including statistical estimation, multilevel Monte Carlo and data assimilation. Chapter 3 will introduce verification and scoring for MLMC approximations before Chapter 4 applies MLMC to the ensemble transform particle filter. Finally Chapters 5 and 6 propose a methodology for using the ETPF and the multilevel version with systems involving finite element approximations to random fields, providing a proof of concept of the methods for high-dimensional systems.

Chapter 2

Uncertainty quantification and Bayesian data assimilation

Chapter synopsis

In this chapter, a survey on probability, uncertainty quantification and statistics is provided. These topics will be introduced in a manner that allow them to build up to the applications used in the later chapters of this thesis. In particular, the basics of statistical estimation (Sec. 2.2) and the discretization of random systems (Sec. 2.4) allow the topic of multilevel Monte Carlo to be introduced in Sec. 2.6. Later in the chapter, Bayesian based ensemble data assimilation (Sec. 2.7) and the ensemble transform particle filter (Sec. 2.7.4) are introduced. The latter will be the main application of multilevel Monte Carlo in later chapters. More detail on the following topics is given in Reich and Cotter [2015]. The following background sections are brief, and are included simply to introduce notation for the remainder of the thesis.

2.1 Preliminaries

This section introduces some notation that is frequently used in the field of uncertainty quantification. Let Ω be a sample space on a probability space. A continuous random variable, X, is the function, $X : \Omega \to E$, where E is some measurable space. In other words it assigns a measurable value to each outcome in the sample space. Most commonly, $E = \mathbb{R}$, and thus the random variable is continuous. However, E can also be a discrete space, for example, in a coin-toss example, where the sample space is simply

$$\Omega = \big\{\text{heads}, \text{tails}\big\}.$$

Given someone bets a stake, S, on this coin-toss being heads, a discrete random variable for the payoff in this example would be

$$X(\omega) = \begin{cases} S, & \omega = \text{heads}, \\ 0, & \omega = \text{tails}. \end{cases}$$

From here on in however, assume that X is continuous and $X : \Omega \to \mathbb{R}^{N_X}$, unless stated otherwise. Notice that random variables state nothing about the probabilities of the measurable values being taken. This aspect gets addressed in the next section. The following definitions can be generalised to dimensions greater than one ($N_X > 1$), however for the simplicity of presentation here we assume that $N_X = 1$.

2.1.1 Probability density functions

A probability density function (PDF), $\pi_X(x)$, of a continuous random variable X is expressed by a map $\pi : \mathbb{R}^{N_X} \to [0, 1]$. [1] It describes the probability of a random variable X taking a certain value. It must satisfy

$$\int_{-\infty}^{\infty} \pi_X(x) dx = 1.$$

[1] In the discrete case it is known as the probability mass function.

Example: normal distribution

Many PDFs have explicit functional forms, such as the normal (Gaussian) distribution, with mean μ and variance σ^2, denoted by $N(\mu, \sigma^2)$. This has PDF

$$\pi_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right).$$

In the case where $\mu = 0$ and $\sigma^2 = 1$, X is known as the standard normal distribution.

2.1.2 Cumulative distribution functions

The cumulative distribution function (CDF), $F_{\pi_X} : \mathbb{R}^{N_X} \to [0, 1]$, of the distribution π_X, describes the probability that X will take a value less than or equal to a specific value. It is given by

$$F_{\pi_X}(x) = \int_{-\infty}^{x} \pi_X(x)dx.$$

A CDF is a critical ingredient in sampling from distributions, primarily through the inverse transform sampling technique, which will be covered in the next section.

2.1.3 Sampling

The ability to sample outcomes from a probability distribution is common-place in uncertainty quantification. For example, in ensemble forecasting, one would like to build an ensemble of scenarios that could occur from a random process; therefore being able to generate the scenarios correctly is crucial. Given the probability distribution, π_X, one can denote independently, identically distributed (i.i.d.) random variables sampled from π_X as $X^i \sim X$, for $i = 1, 2, 3, ...$ These ensembles of random variables will also be denoted by the set notation $\left\{X^i\right\}_{i=1}^{N}$, for an ensemble of size N. Independently identically distributed samples mean that each random variable X^i has the same probability distribution π_X, and are mutually independent.

Many standard distributions such as Gamma, Beta or Gaussian distributions can be sampled from using readily available computational software. Typically this works by generating pseudo-random samples of the uniform distribution and inverting the standard distribution's CDF. This method is called probability inverse transform sampling.

Probability inverse transform sampling

The most common method of sampling from distributions is to use a probability inverse transform. Given a univariate random variable $X : \Omega \to \mathbb{R}$, with distribution π_X, then *inverse transform sampling* is the process of evaluating an inverse CDF, $F^{-1}_{\pi_X}(u)$, $u \in [0, 1]$, also known as the *quantile function*. In the case where the CDF of a random variable is strictly increasing and absolutely continuous (as is the case with F_{π_X}), then there exists a unique value $x \equiv F^{-1}(u)$ for which $F(x) = u$. One can sample from π_X by noting $F^{-1}_{\pi_X}(U) \sim X$, where $U \sim U[0, 1]$ (uniformly distributed random variable). Therefore a sample $X^i \sim X$ can be generated by

$$X^i = F^{-1}_{\pi_X}(u), \quad u \sim U[0, 1].$$

This is a standard, quick method to sample from a distribution π_X, however it only works when the inverse CDF is tractable. In many practical scenarios, this is not the case. In these cases, and where one is able to evaluate $\pi_X(x)$, other sampling methods such as Markov chain Monte Carlo and rejection sampling can be used [Gilks et al., 1995].

2.1.4 Couplings

Given two probability distributions, π_X, and π_Y, associated with the random variables $X : \Omega \to \mathbb{R}^{N_X}$ and $Y : \Omega \to \mathbb{R}^{N_Y}$ respectively, one can form joint probability distributions in order to define the probabilities of X, Y taking certain values when considered together. Let the joint probability density function, $\pi_{(X,Y)}$, of X and Y to be the map $\pi : \mathbb{R}^{N_X \times N_Y} \to [0, 1]$. One can recover the marginal PDFs, π_X and π_Y via the following equation

$$\pi_X(x) = \int_{-\infty}^{\infty} \pi_{X,Y}(x, y)dy,$$

and vice-versa for π_Y. This joint PDF can be referred to as a *coupling* between two distributions. A simple example of a coupling could be the independent coupling, i.e.

$$\pi_{(X,Y)}(x, y) = \pi_X(x)\pi_Y(y).$$

In many cases, which are explored later in the thesis, it might be appropriate to design couplings that can impose either positive or negative correlation between the marginals of X

and Y. Methods to do this include optimal transport [Villani, 2008]. Optimal transport is used throughout this thesis (as early as Sec. 2.7.4) to find transformations between forecast and analysis ensembles in data assimilation.

2.2 Statistical estimation

Statistics of random variables are important in quantifying the expected behavior and uncertainty of them and their associated distributions. For example, computing the mean quantity of a random variable allows expected outcomes to be derived. Statistics of random variables sometimes have analytic expressions, and in other, more practical cases they need to be estimated empirically. Whether one is required to do this depends entirely on the distribution associated with the random variable.

2.2.1 Statistical quantities

There are many statistical quantities of a random variable that one is practically interested in, and they can be defined through expected values.

Definition 1 (Expected values)**.** *An expected value of a measurable functional $g : \mathbb{R}^{N_X} \to \mathbb{R}^d$ of a random variable X, is defined by*

$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x)\pi_X(x)dx.$$

This definition motivates two very common statistical quantities of a random variable, X: the mean and variance.

Definition 2 (Mean)**.** *The mean of a random variable is $\mathbb{E}[X]$, defined by*

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x\pi_X(x)dx. \tag{2.1}$$

Definition 3 (Variance)**.** *The mean of a random variable is $\mathbb{V}[X]$, defined by*

$$\mathbb{V}[X] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \int_{-\infty}^{\infty} (x-\mu)^2\pi_X(x)dx, \tag{2.2}$$

where $\mu = \mathbb{E}[X]$.

In a great deal of practical cases, these integrals are intractable, and have to be estimated discretely. Statistical estimation is most commonly done via the Monte Carlo.

2.2.2 Monte Carlo

As previously mentioned in Chapter 1, statistics to forecast distributions can be estimated via the Monte Carlo method. An ensemble forecast is a ensemble of i.i.d. random variables, typically simulations of a random system. Given an ensemble forecast, Monte Carlo methods can be employed to create estimators, with rigorous error bounds, to statistics of the system for which the ensemble members are realisations of. Consider the estimator $\bar{X}$ of the statistic $\mathbb{E}[g(X)]$. Here, as in the last section, $g : \mathbb{R}^{N_X} \to \mathbb{R}^d$ is an arbitrary, measurable function of X, that is simply used to generalise quantities of interest. If $\bar{X}$ is unbiased, then $\mathbb{E}[\bar{X}] = \mathbb{E}[g(X)]$. With that in mind, denote $\mathbb{E}[\bar{X} - g(X)]$ as the bias of the estimator $\bar{X}$.

Definition 4. *A Monte Carlo estimator of a statistic $\mathbb{E}[g(X)]$ can be computed via sampling N i.i.d. realisations of X, denoted by $\left\{X^i\right\}_{i=1}^N$. The estimator is then given by*

$$\bar{X}_N = \frac{1}{N}\sum_{i=1}^{N} g(X^i). \tag{2.3}$$

This is unbiased, since

$$\mathbb{E}[\bar{X}_N] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} g(X^i)\right] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[g(X)] = \mathbb{E}[g(X)].$$

The Central Limit Theorem (Lindeber-Levy) provides us with the tools to work out the variance within an estimator of this form and therefore deduce a rate of convergence; see (2.16). Assume $\mathbb{V}[g(X)] = \sigma^2 < \infty$ and that $\mathbb{E}[g(X)] = \mu$, then as $N \to \infty$, the random variable $\sqrt{N}(\bar{X} - \mu)$ converges in distribution to $N(0, \sigma^2)$. I.e.

$$\left(\bar{X}_N - \mu\right) \sim \frac{S}{\sqrt{N}},$$

where $S \sim N(0, \sigma^2)$. Typically the variance of $S/\sqrt{N}$, which is σ^2/N, is referred to as the *sampling variance* or *estimator variance*. Sometimes it is important to estimate the variance

of S, and one can do this via the sample variance

$$\frac{1}{N}\sum_{i=1}^{N}(g(X^i)-\bar{X}_N)^2.$$

It can be shown that this estimate is biased with respect to (2.2) by a factor of $(N-1)/N$. An unbiased estimate can therefore by given by

$$\frac{1}{N-1}\sum_{i=1}^{N}(g(X^i)-\bar{X}_N)^2. \tag{2.4}$$

2.2.3 Error analysis of estimators

Error is quite a general term when referring to statistical estimators. Although it can mean the reciprocal of any metric of accuracy, one most often considers the mean square error (MSE) or root mean square error (RMSE). One can suitably analyse convergence rates when using these two metrics. Let the MSE of (2.3) be given by

$$\mathbb{E}\left[\left(\bar{X}_N-\mathbb{E}[g(X)]\right)^2\right],$$

and the RMSE be given by

$$\sqrt{\mathbb{E}\left[\left(\bar{X}_N-\mathbb{E}[g(X)]\right)^2\right]}.$$

One typically estimates the RMSE using

$$\sqrt{\frac{1}{N_{\mathbb{E}}}\sum_{j=1}^{N_{\mathbb{E}}}\left(\bar{X}_N^j-\mathbb{E}[g(X)]\right)^2},$$

where $\left\{\bar{X}_N^j\right\}_{j=1}^{N}$ are $N_{\mathbb{E}}$ independent simulations of the estimator $\bar{X}_N$. When the estimator varies over a temporal grid, t_k, with $k\in[1,N_t]$ (e.g. $\bar{X}_{N,t_k}$), then the time-averaged RMSE can be used as an appropriate accuracy estimate,

$$\sqrt{\frac{1}{N_t}\sum_{k=1}^{N_t}\left(\bar{X}_{N,t_k}-\mathbb{E}[g(X_{t_k})]\right)^2}.$$

These metrics motivate the introduction of the (rates of) convergence for estimators.

Convergence of estimators

There are two types of estimator convergence which will be used throughout this thesis.

Definition 5 (Convergence in probability)**.** *Convergence in probability, also known as the 'consistency of estimators', is used to show consistency of estimators, in the limit of infinite samples, and is defined by*

$$\lim_{N\to\infty} P\left(|\bar{X}_N - \mathbb{E}[g(X)]| > \epsilon\right) = 0,$$

for a small $\epsilon > 0$. This is often described as the weak convergence, in the number of samples, of an estimator.

Definition 6 (Mean square convergence)**.** *Mean square convergence, also known as '$\mathbb{L}_2$ convergence', is a stronger sense of convergence than the above, and is defined by*

$$\lim_{N\to\infty} \mathbb{E}\left[(\bar{X}_N - \mathbb{E}[g(X)])^2\right] = 0.$$

It guarantees that the MSE tends to 0 in the limit of infinite samples.

One can use the Central Limit Theorem from Sec. 2.2.2 to explore the rates of convergence of (2.3). Note that the MSE equals the sampling variance in the absence of bias and therefore the MSE of the estimator in (2.3) can be constrained to

$$\mathbb{E}\left[\left(\bar{X}_N - \mathbb{E}[g(X)]\right)^2\right] \leq \frac{c_1}{N}, \tag{2.5}$$

where c_1 is a constant. This states, that for a MSE of $\mathcal{O}(\epsilon^2)$, for a small ϵ, one requires a sample size for (2.3) of $\mathcal{O}(\epsilon^{-2})$. In this respect, one often refers to (2.3) as having a $\mathcal{O}(N^{-1})$ rate of (mean square) convergence.

2.3 Stochastic differential equations

In most practical cases, the random variable X is the solution to a stochastic differential equation (SDE), and therefore varies over time, i.e. X_t. Stochastic differential equations typically have a trajectory modelled by a Brownian motion, and motivate a great deal of the rest of the thesis. To start with, it is important to define the following. A Brownian motion (or

a Wiener process), $W_t \in \mathbb{R}^{N_X}$, is a continuous-in-time stochastic process at time t [Higham, 2001]. It is used in many applications, from finance [Giles, 2008] to atmospheric dispersion and turbulence parameterizations [Cook, 2013], to incorporate stochasticity into systems.

Definition 7 (Brownian motion)**.** *A Brownian motion, $W_t \in \mathbb{R}^{N_X}$, satisfies the following properties:*

(i) $W_0 = 0$,

(ii) W_t *is continuous in* t,

(iii) $W_{t+s} - W_t$, *for* $s > 0$, *is independent of what has occurred before then,* W_τ, $\tau \leq t$,

(iv) $W_{t+s} - W_t$ *is normally distributed, with mean 0 and variance* s.

These properties allow us to deduce that $\mathbb{E}[W_t] = 0$ and $\mathbb{V}[W_t] = t$. Using a Brownian motion, an SDE can be written in the form of

$$dX_t = a(X_t, t)dt + b(X_t, t)dW_t, \tag{2.6}$$

with $X_t \in \mathbb{R}^{N_X}$, and $W_t \in \mathbb{R}^{N_X}$ is a Brownian motion. Here, a and b are arbitrary functions. By re-writing the SDE in (2.6) in terms of integrals over the time interval $[t, t + s]$, the key components of it can be pulled apart:

$$\int_t^{t+s} dX_u = X_{t+s} - X_t = \int_t^{t+s} a(X_u, u)du + \int_t^{t+s} b(X_u, u)dW_u. \tag{2.7}$$

The second integral is clearly just one over time. The last integral in the expression above is called an Îto integral.

Depending on what a and b are, there are several well known types of SDEs that have explicit properties. For example, consider the case of $a = \theta(\mu - X_t)$ and $b = \sigma$, where $\sigma > 0$, μ and θ are constants. This represents the well-known Ornstein-Uhlenbeck (OU) process; explained in more detail later in the chapter. The limiting probability distribution, as $t \to \infty$, of the trajectories of this process can be described by the Fokker-Planck equation (given in Reich and Cotter [2015] for example). For example, the stationary distribution for trajectories in this limit is a Gaussian distribution with mean μ and variance $\sigma^2/2\theta$. Typically, as with their deterministic ODE counterparts, most SDEs need to be discretized over time (and possibly space); this leads us on to the discretization of random variables.

2.4 Discretization of random variables

Let X^h represent the random variable X with some discretization parameter h (such as a time-step or spatial resolution). The end of this section shows how this parameter adds bias, in addition to the sampling variance of a statistical estimator, in contributing towards the MSE of an estimator involving these discretized random variables.

2.4.1 Numerical methods to solve SDEs

Suppose one wishes to numerically solve the ordinary differential equation (ODE)

$$\frac{dX_t}{dt} = f(X_t, t). \tag{2.8}$$

There are a whole host of available of options to do this, including the forward Euler method. When there is a stochastic term in the differential equation, there are corresponding methods employed to numerically solve them. Two will be introduced: the Euler-Maruyama method and the Milstein method. For a more in-depth look into them, Higham [2001] presents an excellent review.

Underlying Brownian paths

Consider the case where $b(X_t) = 1$ and $a(X_t) = 0$. Then $dX_t = dW_t$ and X_t is a Brownian motion, as introduced earlier in the chapter. The discretization of this can be used to introduce the Euler-Maruyama method. By using the fourth property of a Brownian motion one notes that

$$\delta W := W_{t+h} - W_t \sim N(0, h).$$

Therefore a discretization of a Brownian motion is equivalent to simply making a summation of increments $\delta W \sim N(0, h)$, for an arbitrary time-step h. Denote the discretization as W_t^h. It is otherwise known as a Gaussian random walk. As $h \to 0$, these increments get smaller and smaller, with X_t^h tending to the continuous process, W_t. Figure 1 shows a single discrete Brownian path, W_t^h, $t \in [0, 3]$, discretized with time-steps $h = 0.25$, $h = 0.5$ and $h = 1$. Given a fine discretization of a path W_t^h, one can compute coarser versions of the path, W_t^{nh},

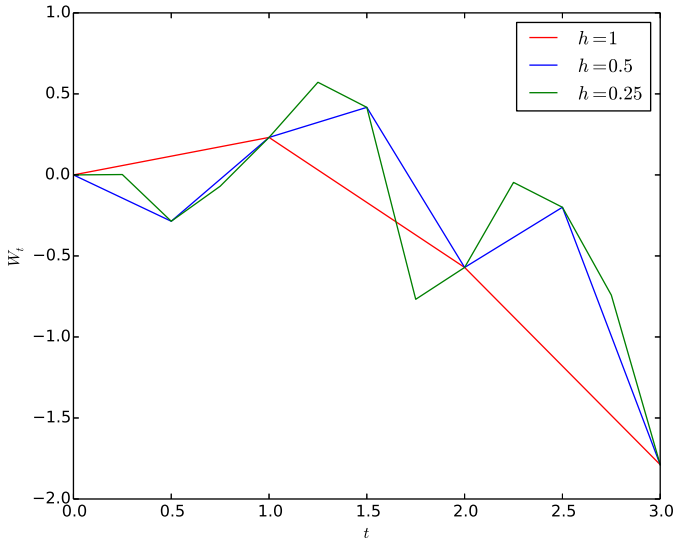


Figure 1: A single Brownian path discretized with a number of different time-steps.

for $n = 2, 3, \ldots$, using

$$W^{nh}_{t+nh} = \sum_{i=1}^{n} W^{h}_{t+ih} - W^{h}_{t+(i-1)h}. \tag{2.9}$$

Euler-Maruyama method

The Euler-Maruyama scheme is a natural extension of the forward Euler scheme for ordinary differential equations. An iteration of the Euler-Maruyama scheme to solve the SDE given in (2.6) is

$$X^{h}_{t+h} = X^{h}_{t} + a(X^{h}_{t}, t)h + b(X^{h}_{t}, t)\delta W, \tag{2.10}$$

where $\delta W \sim N(0, h)$ and X^{h}_{t} is a Euler-Maruyama discretization of X_t. This iteration integrates the SDE over the time interval $[t, t + h]$, by using a single time-step, of size h. More will be discussed about the accuracy and order of error for this method later on in the chapter.

Stratonovich integrals

Apart from the Îto integral in (2.7), there is another stochastic integral type, namely the Stratonovich integral. It is denoted by

$$\int_0^T X_t \circ dW_t.$$

It differs from that of the Îto integral in the last term of (2.7) by the fact that it is the limit, as $h \to 0$, of

$$\sum_{i=0}^{k-1} \left(\frac{X_{(i+1)h} - X_{ih}}{2} \right) \delta W,$$

where $\delta W \sim N(0, h)$ and $T = kh$. This type of integral is considered again in Chapter 5.

Milstein method

The Milstein method adds a correction term to the Euler-Maruyama method. This can improve its convergence properties, which will be described in the next section. An iteration of the Milstein scheme is given by

$$X_{t+h}^h = X_t^h + a(X_t^h, t)h + b(X_t^h, t)\delta W + \frac{1}{2} b(X_t^h, t) b'(X_t^h, t)(\delta W^2 - h),$$

where $\delta W \sim N(0, h)$ and X_t^h is a Milstein discretization of X_t. Again, this iteration integrates the SDE over the time interval $[t, t + h]$, using a single time-step, of size h.

2.4.2 Weak and strong convergence of numerical methods

Earlier in the chapter, the convergence of statistical estimators was considered. In the same way, one can evaluate the numerical convergence of discretized random variables. At the end of the chapter, these numerical convergence rates are shown to impact the convergence rates of statistical estimators involving discretized random variables.

Definition 8 (Weak convergence of numerical approximations)**.** *The weak convergence rate (discretization bias) of a numerical approximation, X^h, is defined by*

$$\left| \mathbb{E}\left[g(X^h) - g(X) \right] \right| \leq c_g h^\alpha, \tag{2.11}$$

where α is a positive constant and $c_g < \infty$ is a constant for any polynomial g, e.g. $g(x) = x^2$ (second moment).

Whilst this can be see as the *error of means* [Higham, 2001] and invariant of the individual paths of X, there is a rate of convergence for the numerical approximation of paths of X: strong convergence.

Definition 9 (Strong convergence of numerical approximations)**.** *The strong convergence rate of a numerical approximation, X^h, is defined by the following*

$$\mathbb{E}\left[\left|X^h - X\right|\right] = \mathcal{O}(h^p), \tag{2.12}$$

where $p \leq \alpha$ is a positive constant. This is a stronger condition than weak convergence.

The Euler-Maruyama method has a weak numerical convergence rate of $\alpha = 1$, given that the deterministic explicit Euler method converges with order 1. Apart from in cases of additive noise (explained in the next section), the scheme has a strong numerical convergence rate of $p = 1/2$. As a demonstration of these numerical convergence types, it is useful to consider the Geometric Brownian Motion,

$$dX_t = \mu X_t dt + \sigma X_t dW_t,$$

and both the Euler-Maruyama and Milstein approximations of it. They are given by

$$X^h_{t+h} = X^h_t + \mu X^h_t h + \sigma X^h_t \delta W,$$

and

$$X^h_{t+h} = X^h_t + \mu X^h_t h + \sigma X^h_t \delta W + \frac{1}{2}\sigma^2 X^h_t (\delta W^2 - h),$$

respectively, where $\delta W \sim N(0, h)$. The parameter values of $\mu = 2$, $\sigma = 1.5$ and $X_0 = 1$ are taken and h varies from 2^{-4} to 2^{-8}. The strong and weak rates of numerical convergence, where expected values are estimated via a Monte Carlo approximation over 10^6 samples, for both numerical schemes are plotted in Figure 2. The approximations are taken over such a large sample size simply to eliminate sampling error to be negligible in our results. As one can see, the extra correction term in the Milstein method provides an increase (from the Euler-Maruyama method in this case) in the strong convergence rate to $p = 1$, the same as

the weak convergence.

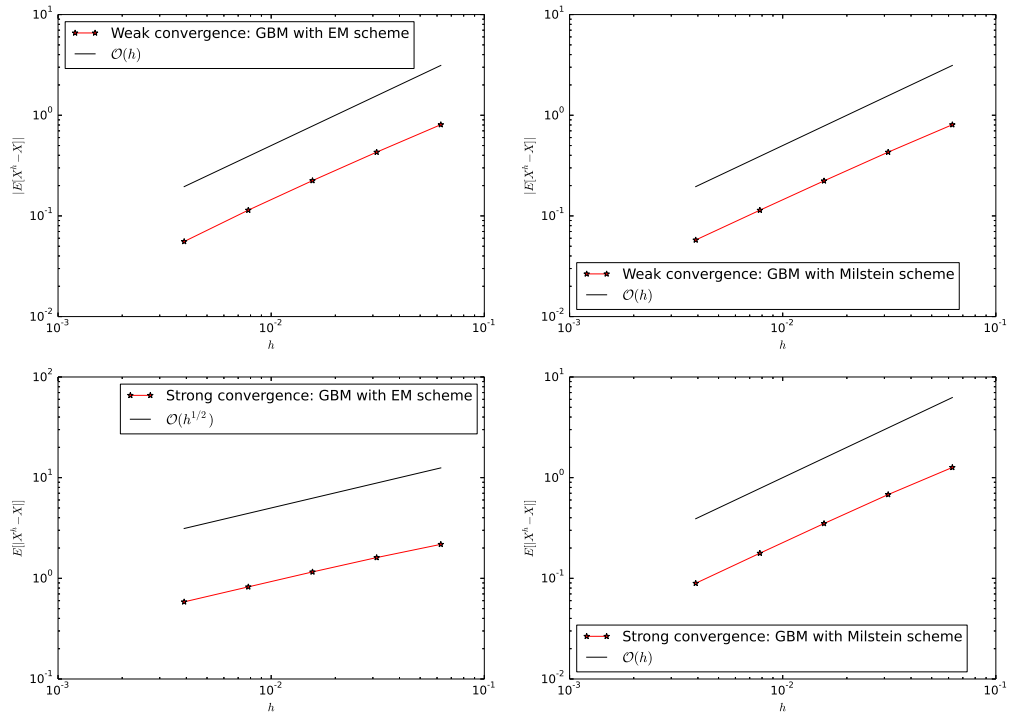


Figure 2: Strong and weak convergence rates for Euler-Maruyama and Milstein numerical schemes approximating the solution to a Geometric Brownian Motion. Both schemes achieve first order weak convergence. However the Milstein scheme achieves a strong convergence rate that is twice the value of the rate obtained by the Euler-Maruyama scheme.

The Geometric Brownian Motion is used later in this chapter for a demonstration of multilevel Monte Carlo; the rates of weak and strong numerical convergence are important for the analysis within this demonstration.

2.4.3 Ornstein Uhlenbeck processes

The general form for an Ornstein Uhlenbeck (OU) process is given by

$$dX_t = -V'(X_t)dt + \sigma dW_t, \tag{2.13}$$

where $X_t \in \mathbb{R}$, $W_t \in \mathbb{R}$ is a one-dimensional Brownian motion. Two common cases of the potential V are:

(i) $V(X_t) = \theta\left(\mu - \frac{\kappa}{2}X_t\right)X_t$,

(ii) $V(X_t) = \frac{\kappa}{4}X_t^4 - \frac{\kappa}{2}X_t^2$,

for constants κ, μ and θ. The former is traditionally known as the linear case, where the process reverts back towards a mean μ in the long-time limiting distribution. The latter is an example of a nonlinear OU process, where the trajectories 'jump' between two attractors, -κ and κ.

Implementing numerical schemes on the Ornstein Uhlenbeck process

One can solve the OU process via the two numerical schemes presented earlier in this chapter. It shall be seen that using the Euler Maruyama method is the same as using the Milstein method, due to the *additive* stochasticity in this system. Additive stochasticity means that the state is not within the Îto integral. Another type of stochasticity is multiplicative. Take $b(X_t) = \sigma X_t$ in (2.6). The stochasticity in the corresponding system is *multiplicative*; that is to say the term in the Îto integral involves a product of the noise and state. Back to the case of additive stochasticity, one can solve the OU process, using the Euler-Maruyama scheme via:

$$X_{t+h}^h = X_t^h - hV'(X_t^h) + \sigma\delta W.$$

Interestingly, this is equivalent to solving it using the Milstein scheme via:

$$\begin{aligned} X_{t+h}^h &= X_t^h - hV'(X_t^h) + \sigma\delta W + \frac{1}{2}\sigma\frac{d\sigma}{dX_t^h}(\delta W^2 - h) \\ &= X_t^h - hV'(X_t^h) + \sigma\delta W + 0 = X_t^h - hV'(X_t^h) + \sigma\delta W. \end{aligned}$$

Therefore, both schemes produce a weak and strong numerical convergence rate of $\mathcal{O}(h)$ [Higham, 2001].

2.4.4 Error analysis of discretized estimators

As in Sec. 2.2.3, it is important to now discuss the convergence of estimators to statistics of discretized random variables. With this in mind, consider the discretized estimator

$$\bar{X}_N^h = \frac{1}{N}\sum_{i=1}^{N} g(X^{h,i}), \tag{2.14}$$

where h is the discretization parameter and $\left\{X^{h,i}\right\}_{i=1}^{N}$ are i.i.d. sample paths of X^h. The MSE of the estimator above can be split up into the two contributions from the discretization bias and sampling error. This can be done by the following Lemma.

Lemma 1. *Consider the estimator in (2.14). Then*

$$\mathbb{E}\left[\left(\bar{X}_N^h - \mathbb{E}[g(X)]\right)^2\right] = \mathbb{E}\left[\left(\bar{X}_N^h - \mathbb{E}[g(X^h)]\right)^2\right] + \mathbb{E}\left[g(X^h) - g(X)\right]^2 \tag{2.15}$$

Proof.

$$\begin{aligned}
\mathbb{E}\left[\left(\bar{X}_N^h - \mathbb{E}[g(X)]\right)^2\right] &= \mathbb{E}\left[\left(\left(\bar{X}_N^h - \mathbb{E}[g(X^h)]\right) - \left(\mathbb{E}[g(X)] - \mathbb{E}[g(X^h)]\right)\right)^2\right] \\
&= \mathbb{E}\left[\left(\bar{X}_N^h - \mathbb{E}[g(X^h)]\right)^2\right] + \mathbb{E}\left[\left(\mathbb{E}[g(X)] - \mathbb{E}[g(X^h)]\right)^2\right] \\
&\quad - 2\mathbb{E}\left[\left(\bar{X}_N^h - \mathbb{E}[g(X^h)]\right)\left(\mathbb{E}[g(X)] - \mathbb{E}[g(X^h)]\right)\right] \\
&= \mathbb{E}\left[\left(\bar{X}_N^h - \mathbb{E}[g(X^h)]\right)^2\right] + \mathbb{E}\left[g(X^h) - g(X)\right]^2
\end{aligned}$$

□

Therefore using the Central Limit Theorem from Sec. 2.2.2 and Lemma 1, the MSE of the estimator in (2.14) can be constrained to

$$\mathbb{E}\left[\left(\bar{X}_N^h - \mathbb{E}[g(X)]\right)^2\right] \leq \frac{c_1}{N} + c_2 h^{2\alpha}, \tag{2.16}$$

where c_1 and c_2 are constants and α is the rate of discretization bias as in (2.11). This is an important bound, as it shows that the error within a discretized estimator is controlled by both the discretization and the number of samples. In particular, both the number of samples and the discretization accuracy need to be increased for the estimator to converge.

2.5 Variance reduction techniques

Whilst the discretization bias term in the (2.16) can be improved by using higher order discretization techniques, the sampling error term is hindered by the fixed rate of decay, $\mathcal{O}(N^{-1})$. However the constant c_1 can be improved to achieve (potentially significant) variance reductions and thus lower error. To do this, variance reduction techniques can be em-

ployed. Typically they play on the idea of correlation and covariance, as the following two techniques do.

2.5.1 Control variates

Consider the following random variable

$$Z = X + c(Y - \mathbb{E}[Y]), \tag{2.17}$$

where c is a constant, and Y is another random variable (known as the control variate) with a known expectation. The first thing to note is that the estimator is unbiased for any c, as $\mathbb{E}[Z] = \mathbb{E}[X]$. An example of such a control variate could be a Brownian motion, where the expectation is known and one can easily sample paths of it. One can construct an estimator of $\mathbb{E}[Z]$, in (2.17), by

$$\bar{Z} = \frac{1}{N}\left(\sum_{i=1}^{N} X^i + c\left(\sum_{i=1}^{N} Y^i - \mathbb{E}[Y]\right)\right),$$

where $\left\{X^i\right\}_{i=1}^{N} \sim X$ and $\left\{Y^i\right\}_{i=1}^{N} \sim Y$. Studying the variance of $\bar{Z}$,

$$\mathbb{V}[\bar{Z}] = \frac{1}{N}\left(\mathbb{V}[X] + c^2\mathbb{V}[Y] + 2c\,\mathrm{Cov}[X, Y]\right),$$

one notes that to minimise this variance the optimal value of c is

$$c^* = -\frac{\mathrm{Cov}[X, Y]}{\mathbb{V}[Y]}.$$

Given this value, the variance of $\bar{Z}$ is

$$\mathbb{V}[\bar{Z}] = \frac{\mathbb{V}[X]}{N} - \frac{\mathrm{Cov}[X, Y]^2}{N\mathbb{V}[Y]},$$

which is a reduction from $\mathbb{V}[\bar{X}]$, where $\bar{X}_N$ is given in (2.3), for any $\mathrm{Cov}[X, Y] \neq 0$. The larger this covariance, the greater the variance reduction. In practice, one can estimate $\mathrm{Cov}[X, Y]$ (and $\mathbb{V}[Y]$, although this might well be known as well) to generate the optimal value c^*.

2.5.2 Antithetic variates

Antithetic variates play on the identity

$$\mathbb{V}[X+Y] = \mathbb{V}[X] + \mathbb{V}[Y] + 2\,\mathrm{Cov}[X,Y]$$

to reduce the variance of an estimator $\bar{X}_N$, in (2.3). Consider the following estimator

$$\bar{X}' = \frac{1}{2N'}\sum_{i=1}^{N'}(X_1^i + X_2^i),$$

where $\{X_1^i\}_{i=1}^{N'} \sim X$, $\{X_2^i\}_{i=1}^{N'} \sim X$ and $2N' = N$. Studying the variance of $\bar{X}'$, it is found that

$$\mathbb{V}[\bar{X}'] = \frac{N'}{4(N')^2}\mathbb{V}[X_1 + X_2].$$

Further,

$$\mathbb{V}[\bar{X}'] = \frac{1}{2N}\big(2\mathbb{V}[X] + 2\,\mathrm{Cov}[X_1, X_2]\big),$$

and so $\mathbb{V}[\bar{X}'] < \mathbb{V}[\bar{X}_N]$ (for the same computational cost), if $\mathrm{Cov}[X_1, X_2] < 0$. Crucially, $\bar{X}'$ is also unbiased since

$$\mathbb{E}[\bar{X}'] = \frac{1}{2}\big(\mathbb{E}[X_1] + \mathbb{E}[X_2]\big) = \mathbb{E}[X].$$

Now how does one construct X_1 and X_2 such that $\mathrm{Cov}[X_1, X_2] < 0$? In complicated systems of X, this can be difficult, however it could be as simple as taking $X_1 \sim -W_t$, where $X_2 \sim W_t$ and W_t is a standard Brownian motion. Antithetic variates are useful, and have been applied to lots of areas of uncertainty quantification, including sequential Monte Carlo [Olsson and Bizjajeva, 2016].

A major development in variance reduction is the novel multilevel Monte Carlo technique. This is a generalisation of the control variate idea in Sec. 2.5.1 and uses the numerical discretization aspect of Sec. 2.4 as well.

2.6 Multilevel Monte Carlo

In the following section, the statistical framework of multilevel Monte Carlo (MLMC), which will be used throughout the remainder of this thesis, will be described and introduced.

This framework was coined in Giles [2008] for the efficient computation of estimators for statistics of discretized random variables, representing the underlying stock price in option pricing. Since then, the range of literature using this framework has exploded into many areas of uncertainty quantification including Markov chain Monte Carlo [Ketelsen et al., 2013], atmospheric dispersion modelling [Katsiolides et al., 2018] and data assimilation [Gregory et al., 2016, Jasra et al., 2017a].

The idea is based on control variates, previously introduced in this chapter, and so in principle it is a variance reduction method for Monte Carlo estimators. It is important to remember that this methodology is very general and can be applied in a great variety of settings, as already highlighted by the range of applications it has appeared in. For a fixed size of MSE, the multilevel estimator can return significant computational cost reductions from its standard Monte Carlo counterpart.

2.6.1 The high price paid for uncertainty quantification

Section 2.2.3 motivates the use of such a methodology; it is clear there is a very high computational complexity associated with producing accurate estimators via the Monte Carlo method when considering high-dimensional state spaces for discretized random variables. To increase the accuracy of the estimator, one must increase the resolution and the number of samples used simultaneously. The following proposition explains this.

Proposition 1. *Consider a Monte Carlo estimator of a discretized random variable, in (2.14), then the computational complexity / cost of it is*

$$\textit{Cost}(\bar{X}_N^h) = \mathcal{O}(Nh^{-\gamma}).$$

This is simply N times the cost of producing one sample of $g(X^h)$, where γ is a constant. In a temporally discretized system, where h is a time-step, then $\gamma = 1$. Theoretically, given the proposition above, the cost of producing a Monte Carlo estimator of the form in (2.14), with MSE $\mathcal{O}(\epsilon^2)$, for a small ϵ, is

$$\text{Cost}(\bar{X}_N^h) = \mathcal{O}(\epsilon^{-2-(\gamma/\alpha)}). \tag{2.18}$$

Here α is the rate of weak convergence for the numerical discretization scheme. For a tempo-

rally discretized system, where $\gamma = 1$, this complexity is feasible for high accuracy estimators, however when γ is higher (encountered in the implementation of the quasi-geostrophic equations in Chapters 5 and 6), this complexity can become very large. Statistical techniques, such as the multilevel framework discussed in this chapter, are therefore aimed at reducing this complexity whilst fixing the order of accuracy above.

Computational cost is estimated by runtime (seconds) on a HP Z420 Workstation Linux OS computer (Python implementation) throughout the rest of this thesis.

2.6.2 A hierarchy of discretization levels

Recall h relates to a discretization parameter (assumed to be temporal resolution, e.g. time-step, in this section). A hierarchy of these parameters can be governed by the rule $h_l \propto m^{-l}$, with $m > 1$. Denote the 'level' of the discretized random variable X^{h_l} as l. It is important to the multilevel method that the hierarchy of levels creates a hierarchy of accuracy as well. A theorem formally stated later in the chapter contains a condition that makes this clear. This condition is centered around weak convergence stated in Sec. 2.4.2; the assumption of this hierarchy of levels is that

$$\left|\mathbb{E}\left[X^{h_l} - X\right]\right| = \mathcal{O}\left(h_l^{\alpha}\right),$$

for an $\alpha > 0$. Another important assumption that is made within the multilevel literature is that $\gamma > 0$. Put simply, there has to be some price-to-pay in computational complexity for the increase in accuracy one gets from increasing l.

2.6.3 A multilevel estimator

Suppose one wishes to estimate $\mathbb{E}[g(X^{h_L})]$, for a sufficiently large L, and for a scalar functional $g : \mathbb{R}^{N_X} \to \mathbb{R}$, then the standard Monte Carlo estimator in (2.14) can be used,

$$\bar{X}_N^{h_L} = \sum_{i=1}^{N} g(X^{h_L,i}).$$

However $\mathbb{E}[g(X^{h_L})]$ can also be written as

$$\mathbb{E}[g(X^{h_L})] = \mathbb{E}[g(X^{h_0})] + \sum_{l=1}^{L} \mathbb{E}[g(X^{h_l})] - \mathbb{E}[g(X^{h_{l-1}})]. \tag{2.19}$$

Consider estimating this 'telescoping sum of expectations' with the following linear combination of Monte Carlo estimators, μ_l, with sample sizes N_l, for $l = 0, ..., L$ respectively,

$$\bar{X}^{h_L}_{N_0,...,N_L} = \sum_{l=0}^{L} \mu_l, \tag{2.20}$$

where

$$\mu_l = \begin{cases} \frac{1}{N_0} \sum_{i=1}^{N_0} g(X^{h_0,i}), & l = 0, \\ \frac{1}{N_l} \sum_{i=1}^{N_l} g(X^{h_l,i}) - g(\hat{X}^{h_{l-1},i}), & l \geq 1, \end{cases} \tag{2.21}$$

and $\left\{\hat{X}^{h_l,i}\right\}_{i=1}^{N_{l+1}} \sim X^{h_l}$ but are independent of the samples $\left\{X^{h_l,i}\right\}_{i=1}^{N_l}$. Note that under i.i.d. sampling of the discretized random variables, as in the standard Monte Carlo estimator, this multilevel estimator is unbiased. That is to say,

$$\mathbb{E}\left[\bar{X}^{h_L}_{N_0,...,N_L}\right] = \mathbb{E}\left[g(X^{h_L})\right].$$

The estimators μ_l, for $l = 0, ..., L$, are referred to as 'difference estimators', due to the expected difference between the coarse and fine approximations they estimate. The variance of the multilevel estimator, $\bar{X}^{h_L}_{N_0,...,N_L}$, is then

$$\mathbb{V}\left[\bar{X}^{h_L}_{N_0,...N_L}\right] = \sum_{l=0}^{N_l} \frac{\mathbb{V}_l}{N_l}, \tag{2.22}$$

where

$$\mathbb{V}_l = \begin{cases} \mathbb{V}[g(X^{h_0})], & l = 0, \\ \mathbb{V}\left[g(X^{h_l}) - g(X^{h_{l-1}})\right], & l \geq 1, \end{cases} \tag{2.23}$$

assuming that the μ_l's are independent of one another and that the $X^{h_l,i}$'s and the $\hat{X}^{h_l,i}$'s are sampled i.i.d. for $i = 1, ..., N_l$. An important requirement of the multilevel estimator is to have some balance of the contributions from each term within the summation in (2.22). That is to say if $\mathbb{V}_l$ was to asymptotically decrease as $l \to \infty$, N_l likewise could asymptotically decrease. Manipulating this balance, to reduce the computational cost of computing (2.20) as much as possible, is primarily dependent on choosing N_l carefully, and is discussed in more detail in a later section.

2.6.4 Efficiency of the multilevel estimator

The true power of the multilevel estimator can be realised from an efficiency perspective. Efficiency, in this thesis, is measured via the ratio of computational complexity with respect to accuracy; this is defined as the Cost/Accuracy Ratio (CAR). For example, consider two estimators, a and b, both estimating $\mathbb{E}[c]$. If they both have the same order of accuracy, say

$$\mathbb{E}\left[(a - \mathbb{E}[c])^2\right] = \mathcal{O}(\epsilon^2), \qquad \mathbb{E}\left[(b - \mathbb{E}[c])^2\right] = \mathcal{O}(\epsilon^2),$$

for some small ϵ, then if $\text{Cost}(a) = \mathcal{O}(\epsilon^{-p})$ and $\text{Cost}(b) = \mathcal{O}(\epsilon^{-q})$, with $p < q$, a is more efficient than b. This motivates a formal definition of the CAR and thus efficiency to be given.

Definition 10 (Cost/Accuracy Ratio)**.** *An estimator, θ, of $\mathbb{E}[X]$, has a CAR value given by*

$$\textit{CAR}(\theta) = \frac{\textit{Cost}(\theta)}{\mathbb{E}\left[(\theta - \mathbb{E}[X])^2\right]^{-1}} = \mathbb{E}\left[(\theta - \mathbb{E}[X])^2\right] \textit{Cost}(\theta).$$

If an estimator has a lower CAR value than another, it is more efficient.

Using this, the CAR of a is $\mathcal{O}(\epsilon^{2-p})$ and the CAR of b is $\mathcal{O}(\epsilon^{2-q})$. Moving back to the multilevel estimator, if one obtains a MSE for (2.20) of

$$\mathbb{E}\left[\left(\bar{X}^{h_L}_{N_0,\dots N_L} - \mathbb{E}[g(X)]\right)^2\right] = \mathcal{O}(\epsilon^2), \tag{2.24}$$

then Giles [2008], Cliffe et al. [2011] state that if

$$\left|\mathbb{E}\left[g(X^{h_l}) - g(X^{h_{l-1}})\right]\right| = \mathcal{O}(h_l^{\alpha}), \tag{2.25}$$

and

$$\mathbb{V}\left[g(X^{h_l}) - g(X^{h_{l-1}})\right] = \mathcal{O}(h_l^{\beta}), \tag{2.26}$$

for some positive constants β, γ and $\alpha \geq (1/2)\min(\beta, \gamma)$ then there exist values of L and

N_l, for $l = 0, ..., L$, such that (2.20) has a complexity cost of

$$\text{Cost}(\bar{X}^{h_L}_{N_0,...,N_L}) = \begin{cases} \mathcal{O}(\epsilon^{-2}), & \beta > \gamma \\ \mathcal{O}(\epsilon^{-2}\log(\epsilon)^2), & \beta = \gamma \\ \mathcal{O}(\epsilon^{-2-(\gamma-\beta)/\alpha}), & \beta < \gamma. \end{cases} \tag{2.27}$$

For a proof of this, see Cliffe et al. [2011]. Finally, it can be derived that the complexity cost growth rates in (2.27) dictate the following CAR rates:

$$\text{CAR}(\bar{X}^{h_L}_{N_0,...,N_L}) = \begin{cases} \mathcal{O}(1), & \beta > \gamma \\ \mathcal{O}(\log(\epsilon)^2), & \beta = \gamma \\ \mathcal{O}(\epsilon^{(\beta-\gamma)/\alpha}), & \beta < \gamma \end{cases} \tag{2.28}$$

which are less than that of the Monte Carlo estimator,

$$\text{CAR}(\bar{X}^{h_L}_{N}) = \mathcal{O}(\epsilon^{-(\gamma/\alpha)}), \tag{2.29}$$

for any $\beta > 0$, and $\epsilon < e^{-1}$.

2.6.5 The multilevel Monte Carlo algorithm

There is an adaptive algorithm to select the values of N_l and L that obtain the efficiency advantages stated in (2.28) [Giles, 2008]. For a general setting, this proceeds as follows, starting with $L = 0$:

(1) Compute μ_L and estimate $\mathbb{V}_L$ with a small number of samples n.

(2) Calculate the optimal N_l, for $l = 0, ..., L$ via

$$N_l = \left\lceil 2\epsilon^{-2}\sqrt{\mathbb{V}_l h_l^{\gamma}} \left(\sum_{l=0}^{L} \sqrt{\mathbb{V}_l h_l^{-\gamma}} \right) \right\rceil. \tag{2.30}$$

(3) Generate extra samples, if needed, given the updated N_l, for $l = 0, ..., L$, and re-estimate μ_l.

(4) Check for convergence by $|\mu_L| \leq (M-1)\frac{\epsilon}{\sqrt{2}}$.

(5) If converged, stop the algorithm and estimate $\bar{X}^{h_L}_{N_0,...N_L}$ via (2.20). If not converged, return to step (1) and set $L = L + 1$.

One can estimate the variances $\mathbb{V}_l$ by using the sample variance in (2.4). This actually *bounds* the MSE by ϵ^2 rather than establishing an order of accuracy. The generality of instead using an order of accuracy here is useful for the application of these estimators to data assimilation later in the thesis, where estimators are only consistent and have small biases (see self-normalized sequential importance sampling estimators in Sec. 2.7.2).

In this algorithm, N_l is chosen adaptively based on the variance reduction rate, with respect to the cost-per-sample rate, γ. Thus for a positive β, one expects that N_l decreases asymptotically as $l \to \infty$.

2.6.6 Variance reduction between fine and coarse estimators

From the efficiency rates in (2.28), it is clear that the greater β is in the condition (2.26), the greater the efficiency of the multilevel Monte Carlo algorithm will be, up until the optimal case of $\beta > \gamma$. Indeed, this β parameter is very significant in the quality of multilevel Monte Carlo applications, and henceforth it will be referred to as the variance reduction parameter. The reason for which the efficiency is greater when β is large, is that the sample sizes for the estimators, μ_l, with $l = 0, ..., L$, decay faster as $l \to L$ as apparent in (2.30). Therefore estimators at a finer resolution, which cost more to compute, are made up of less samples. This creates a balance in computational cost of each difference estimator. As a demonstration of this, consider the case where $\beta = \gamma = \alpha = 1$, and thus $\mathbb{V}_l = \mathcal{O}(h_l)$. Following the algorithm in the previous section, this creates a decay of

$$N_l = \mathcal{O}\left((L+1)\epsilon^{-2}\sqrt{h_l^2}\right) = \mathcal{O}((L+1)h_l\epsilon^{-2}),$$

and an overall variance of

$$\mathbb{V}[\bar{X}^{h_L}_{N_0,...,N_L}] = \sum_{l=0}^{L} \frac{\mathbb{V}_l}{N_l} = \sum_{l=0}^{L} \mathcal{O}((L+1)^{-1}h_l h_l^{-1}\epsilon^2) = \mathcal{O}(\epsilon^2). \tag{2.31}$$

Using (2.15), (2.19), (2.22) and also (2.31) one can constrain the MSE of the multilevel Monte Carlo estimator to

$$\mathbb{E}\left[\left(\bar{X}^{h_L}_{N_0,\dots N_L} - \mathbb{E}[g(X)]\right)^2\right] \leq c_1\epsilon^2 + c_2 h_L^{2\alpha},$$

where c_1 and c_2 are constants. Therefore this estimator achieves a MSE of $\mathcal{O}(\epsilon^2)$, as the overall bias is $\left|\mathbb{E}[g(X^{h_L}) - g(X)]\right| = \mathcal{O}(h_L) = \mathcal{O}(\epsilon)$ (from the convergence condition in the algorithm). However the computational cost is only

$$\text{Cost}(\bar{X}^{h_L}_{N_0,\dots,N_L}) = \mathcal{O}\left(\sum_{l=0}^{L} N_l h_l^{-1}\right) = \mathcal{O}((L+1)^2\epsilon^{-2}h_l^{-1}h_l)) = \mathcal{O}\left(\epsilon^{-2}\log(\epsilon)^2\right),$$

given that

$$m^{-L} = \mathcal{O}(\epsilon) \quad \Longrightarrow \quad -L\log(m) = \mathcal{O}(\log(\epsilon)) \quad \Longrightarrow \quad L+1 = \mathcal{O}\left(\log(\epsilon^{-1})\right),$$

if $\epsilon < e^{-1}$ and $m = \mathcal{O}(1)$. This creates a CAR value of $\mathcal{O}\left(\log(\epsilon)^2\right)$; this is less than the CAR value of the Monte Carlo estimator for $\gamma = 1$, which is $\mathcal{O}(\epsilon^{-1})$.

One can gain more of an insight into this parameter variance reduction parameter β by re-writing $\mathbb{V}_l$, for $l = 1, ..., L$, as

$$\mathbb{V}_l = \mathbb{V}[g(X^{h_l})] + \mathbb{V}[g(X^{h_{l-1}})] - 2\,\text{Cov}[g(X^{h_l}), g(X^{h_{l-1}})].$$

By enforcing positive correlation between $\left\{X^{h_l,i}\right\}_{i=1}^{N_l}$ and $\left\{\hat{X}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ one can reduce $\mathbb{V}_l$ (see variance reduction techniques, Sec. 2.5). If this correlation is sufficiently strong, an asymptotic rate of this reduction as $l \to \infty$, can be established. A way of enforcing correlation between $\left\{X^{h_l,i}\right\}_{i=1}^{N_l}$ and $\left\{\hat{X}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ is by using the same underlying random input in the system for each pair of samples. This could be a case of using the same initial condition, or the same underlying Brownian path for an SDE, as in Sec. 2.4.1. For example, given $2h_l = h_{l-1}$, if $X_t^{h_l} \sim W_t^{h_l}$ and $\hat{X}_t^{h_{l-1}} \sim W_t^{h_{l-1}}$, then one can correlate $\left\{X^{h_l,i}\right\}_{i=1}^{N_l}$ and $\left\{\hat{X}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ by sampling $\left\{X^{h_l,i}\right\}_{i=1}^{N_l} \sim W_t^{h_l}$ and setting

$$\hat{X}^{h_{l-1},i}_{t+h_{l-1}} = \sum_{j=1}^{2}\left(X^{h_l,i}_{t+jh_l} - X^{h_l,i}_{t+(j-1)h_l}\right).$$

The parameters $\alpha = 1$ and $\gamma = 1$ used in the analysis above, correspond to using an Euler-Maruyama scheme to discretize a Geometric Brownian Motion [Giles, 2008], as seen previously in the chapter. For such a simple case, a rate of β can be found from the strong rate of convergence (assuming that one correlates $\left\{X^{h_l,i}\right\}_{i=1}^{N_l}$ and $\left\{\hat{X}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ by using the same underlying Brownian paths, via (2.9)). Suppose g is a scalar function, and is Lipschitz continuous, i.e. $|g(a) - g(b)| \leq c^2\|a - b\|$, where c is a constant. Note that

$$\mathbb{V}[g(X^{h_l}) - X] \leq \mathbb{E}\left[\left(g(X^{h_l}) - g(X)\right)^2\right] \leq c^2\mathbb{E}\left[\left\|X^{h_l} - X\right\|^2\right],$$

and therefore

$$\mathbb{V}[g(X^{h_l}) - X] = \mathcal{O}(h_l), \tag{2.32}$$

since the strong rate of convergence for the Euler-Maruyama scheme for the Geometric Brownian Motion is $p = 1/2$. Now consider

$$g(X^{h_l}) - g(X^{h_{l-1}}) = \left(g(X^{h_l}) - g(X)\right) - \left(g(X^{h_{l-1}}) - g(X)\right).$$

Recall that $\mathbb{V}[A - B] = \mathbb{V}[A] + \mathbb{V}[B] - 2\,\mathrm{Cov}[A, B]$, for random variables A and B, and also $\mathrm{Cov}[A, B] = \rho(A, B)\mathbb{V}[A]^{1/2}\mathbb{V}[B]^{1/2}$, where $\rho(A, B)$ is the correlation coefficient between A and B. Then if $\rho\left(g(X^{h_l}), g(X^{h_{l-1}})\right) > 0$ it holds that

$$\mathbb{V}[g(X^{h_l}) - g(X^{h_{l-1}})] \leq \left(\mathbb{V}[g(X^{h_l}) - g(X)]^{1/2} + \mathbb{V}[g(X^{h_{l-1}}) - g(X)]^{1/2}\right)^2.$$

This finally gives us $\mathbb{V}[g(X^{h_l}) - g(X^{h_{l-1}})] = \mathcal{O}(h_l^\beta)$, where $\beta = 1$. In this case, following the MLMC algorithm, N_l would decrease asymptotically with a linear rate. If the Milstein scheme is used for the analysis above, the strong rate of convergence would increase to $p = 1$. Therefore, (2.32) becomes $\mathbb{V}[g(X^{h_l}) - X] = \mathcal{O}(h_l^2)$, and the variance reduction rate increases to $\beta = 2$ in the same way as above. Following the efficiency rates in (2.28), this variance reduction rate gives us a more efficient estimator than with $\beta = 1$.

2.6.7 An example

The effectiveness of the multilevel Monte Carlo method will be demonstrated with a simple problem. Consider the Geometric Brownian Motion, X_t, discussed previously in the chapter, discretized via the Milstein method; our goal is create the most efficient estimator of $\mathbb{E}[X_T]$

as possible, with $T = 2$ and MSE of $\mathcal{O}(\epsilon^2)$. Both standard Monte Carlo and multilevel Monte Carlo estimators will be considered. Take $\mu = 0.2$, $\sigma = 0.15$ and $X_0 = 1$. For the multilevel Monte Carlo estimator, the coarse and fine samples in each difference estimator, (2.21), will be coupled by using the same underlying Brownian motion, via (2.9). Of course, for this problem, $\mathbb{E}[X_T]$, is known, it is indeed $e^{\mu t}$. Mean square errors away from this are estimated over $N_{\mathbb{E}} = 25$ simulations of the estimators. The algorithm described earlier in this chapter will be used to construct the multilevel Monte Carlo estimator, whilst standard Monte Carlo estimators will be computed by the following algorithm:

(1) Choose $L = \left\lceil \left(\log(\epsilon^{-1}\sqrt{2})/\alpha \log(m)\right) \right\rceil$.

(2) Estimate $\mathbb{V}[X_T^{h_L}]$ using a small number of samples n.

(3) Calculate $N = \left\lceil 2\epsilon^{-2}\mathbb{V}[X_T^{h_L}] \right\rceil$.

(4) If needed, generate $N - n$ extra samples and estimate $\mathbb{E}[X_T^{h_L}]$ using (2.14).

Firstly, one can estimate the statistics, μ_l and $\mathbb{V}_l$, for a multilevel Monte Carlo estimator of $\mathbb{E}[X_T]$ to observe the asymptotic analysis, with increasing l, that was presented earlier. Take $h_l = m^{-l-1}$ and $m = 2$ for this example. These statistical quantities, for fixed $l \in [0, 7]$ and $N_l = 100000$ are shown in Figure 3. Note the 'jump' down from $l = 0$ to $l = 1$, where expected differences and variance of differences are being computed rather than simply $\mathbb{E}[X_T^{h_0}]$ and $\mathbb{V}[X_T^{h_0}]$ on the coarsest level. The quantities, $|\mathbb{E}[X_T^{h_l} - X_T^{h_{l-1}}]|$ and $\mathbb{V}[X_T^{h_l} - X_T^{h_{l-1}}]$, then continue to decay asymptotically at rates of $\mathcal{O}(h_l^{\alpha})$ and $\mathcal{O}(h_l^{\beta})$, with $\alpha = 1$ and $\beta = 2$ respectively. This asymptotic behaviour is expected given the weak and strong convergence rates of the Milstein numerical method.

Secondly, the efficiency of the multilevel Monte Carlo and standard Monte Carlo estimators of $\mathbb{E}[X_T]$ is considered for various values of ϵ. Figure 4 shows the computational cost of these estimators against the MSEs of them. Given that $\gamma = 1$ for the Geometric Brownian Motion, and the variance reduction rate of $\beta = 2$ is achieved, the computational cost of the MLMC estimator scales as $\mathcal{O}(\epsilon^{-2})$, instead of $\mathcal{O}(\epsilon^{-3})$ for the standard MC estimator. These scalings result in the theoretical CAR values of $\mathcal{O}(1)$ and $\mathcal{O}(\epsilon^{-1})$ respectively.

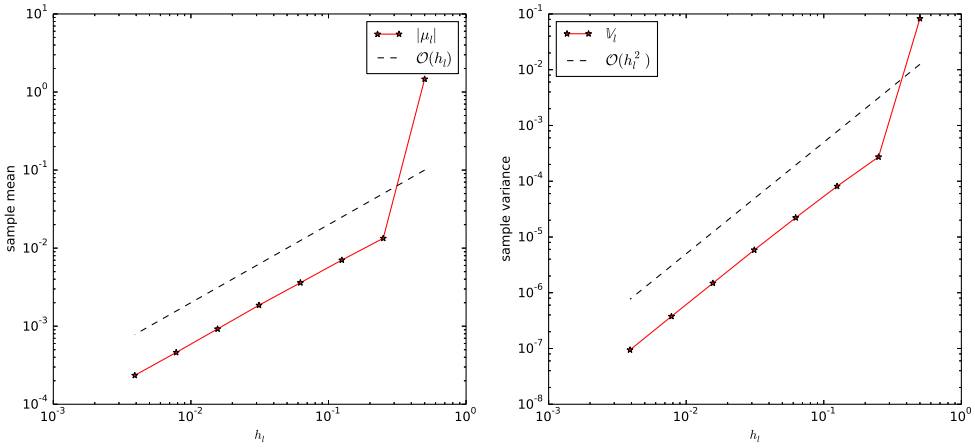


Figure 3: Estimates of the statistical quantities, $|\mu_l|$ and $\mathbb{V}_l$, for $l \in [0, 7]$, from a multilevel Monte Carlo estimator of the expected value of a Geometric Brownian Motion. Following the asymptotes shown by the black dashed lines, the quantity $|\mu_l|$ decays at a rate of $\mathcal{O}(h_l)$ whilst the quantity $\mathbb{V}_l$ decays at a rate of $\mathcal{O}(h_l^2)$.

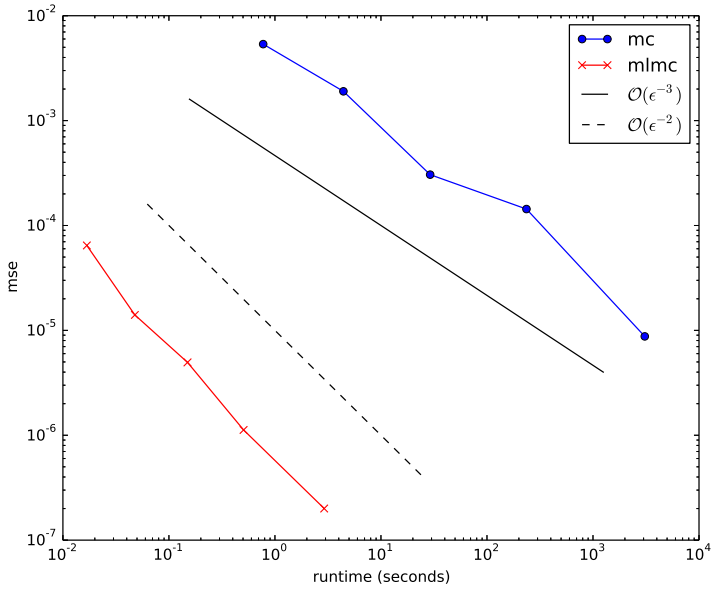


Figure 4: The runtime (seconds) of Monte Carlo (MC) and multilevel Monte Carlo (MLMC) estimators against their MSEs away from the expected value of a Geometric Brownian Motion, $\mathbb{E}[X_T]$, for various values of ϵ. Following the asymptotes shown by the black dashed and solid lines, the computational cost of the MLMC and MC estimators scale as $\mathcal{O}(\epsilon^{-2})$ and $\mathcal{O}(\epsilon^{-3})$ respectively.

2.6.8 Vector functionals

Now consider the case where $g : \mathbb{R}^{N_X} \to \mathbb{R}^d$, $d > 1$, and d is finite. One can extend the MSE metric for the multilevel estimator in (2.20) to this case:

$$\mathbb{E}\left\| \bar{X}^{h_L}_{N_0,\dots,N_L} - \mathbb{E}[X] \right\|^2 = \sum_{m=1}^{d} \mathbb{E}\left[\left(\bar{X}^{h_L}_{N_0,\dots,N_L}(m) - \mathbb{E}\left[g\big(X\big)(m) \right] \right)^2 \right], \tag{2.33}$$

where $X(m)$ is the m'th component of X. Assuming that $d = \mathcal{O}(1)$ (and therefore not dependent on ϵ), then

$$\mathbb{E}\left\| \bar{X}^{h_L}_{N_0,\dots,N_L} - \mathbb{E}[X] \right\|^2 = \mathcal{O}(\epsilon^2),$$

is obtained if the MSE of all components of X is $\mathcal{O}(\epsilon^2)$. For the multilevel Monte Carlo algorithm given earlier in the chapter, N_l is adaptively set to actually bound the error by ϵ^2. This algorithm can be modified to bound the error by $d\epsilon^2$ for vector functionals. It is suggested in Giles [2015] that one uses

$$\mathbb{V}_l = \max_m \left(\mathbb{V}_{l,m} \right),$$

where

$$\mathbb{V}_{l,m} = \begin{cases} \mathbb{V}[g\left(X^{h_0}\right)(m)], & l = 0, \\ \mathbb{V}\left[g\left(X^{h_l}\right)(m) - g\left(X^{h_{l-1}}\right)(m) \right], & l \geq 1, \end{cases} \tag{2.34}$$

for the formula for the optimal sample sizes, N_l, in (2.30). This means that the variance of all components will be bounded by $\epsilon^2/2$ at the very least.

In the case that d is dependent on ϵ, e.g. $d = \mathcal{O}(\epsilon)$, which may occur when X^{h_l} is spatially discretized, the above analysis does not hold. This occurs later on in this thesis, and an alternative metric will be given that involves integral quantities for functions of space and time.

This chapter will now consider ensemble data assimilation, another major research area within ensemble forecasting. Multilevel Monte Carlo will be applied to this area of research later in the thesis.

2.7 Bayesian ensemble data assimilation

Bayesian methods are an extremely important field in statistics, and motivate many data assimilation schemes. Bayes' rule is the underlying law of this field, given simply by

$$\pi_{X|Y}(x|y_{\text{data}}) = \frac{\pi_Y(y_{\text{data}}|x)\pi_X(x)}{\int \pi_Y(y_{\text{data}}|x')\pi_X(x')dx'}. \tag{2.35}$$

Bayesian inference is a term to describe using the above rule to infer information about a random variable $X : \Omega \to \mathbb{R}^{N_X}$, with associated distribution π_X, given observed data y_{data}, with associated distribution π_Y. Here, $\pi_X(x)$ is the prior belief of X, $\pi_Y(y_{\text{data}}|x)$ is the likelihood of y_{data} given X and $\pi_{X|Y}(x|y_{\text{data}})$ is the posterior belief of X given y_{data}. The next section will use Bayes' rule to introduce the Bayesian interpretation of a specific type of data assimilation: filtering.

2.7.1 Filtering

Filtering will be concentrated upon on throughout this thesis. Filtering is used when X and y_{data} are temporal (and possibly spatial) processes, X_t and $y_{\text{data},t}$, at time $t \geq 0$, with associated distributions π_{X_t} and π_{Y_t} respectively. To propagate the process X_t from time t to time $s > t$, a (possibly random) forward prediction model is required:

$$X_s = M(t, X_t, \omega), \qquad \omega \in \Omega. \tag{2.36}$$

Then Bayes' rule is applied sequentially over numerous 'assimilation steps' t_k, for $k = 1, 2, 3, ...$, with $\Delta t = t_{k+1} - t_k$. For example, at time t_1, the posterior of X_{t_1} given the data y_{data,t_1} is

$$\pi_{X_{t_1}|Y_{t_1}}(x|y_{\text{data},t_1}) = \frac{\pi_{Y_{t_1}}(y_{\text{data},t_1}|x)\pi_{X_{t_1}}(x)}{\int \pi_{Y_{t_1}}(y_{\text{data},t_1}|x')\pi_{X_{t_1}}(x')dx'}.$$

Then recursively at an arbitrary time t_k, $k > 1$, the posterior of X_{t_k} given the data at every time up to and including the k'th assimilation step, $y_{\text{data},t_1:t_k}$, is

$$\pi_{X_{t_1}|Y_{t_1:t_k}}(x|y_{\text{data},t_1:t_k}) = \frac{\pi_{X_{t_k}|Y_{t_1:t_{k-1}}}(x|y_{\text{data},t_1:t_{k-1}})\pi_{Y_{t_k}}(y_{\text{data},t_k}|x)}{\int \pi_{X_{t_k}|Y_{t_1:t_{k-1}}}(x'|y_{\text{data},t_1:t_{k-1}})\pi_{Y_{t_k}}(y_{\text{data},t_k}|x')dx'}. \tag{2.37}$$

The distributions $\pi_{X_{t_1}}$ and $\pi_{X_{t_k}|Y_{t_1:t_{k-1}}}$, for $k > 1$, are known as the forecast distributions in a data assimilation context. If the underlying prediction model M is Markovian, then the forecast distributions for $k > 1$ are given by the Chapman-Kolmogorov equation:

$$\pi_{X_{t_k}|Y_{t_1:t_{k-1}}}(x|y) = \int \pi_{X_{t_k}|X_{t_{k-1}}}(x|x_{t_{k-1}})\pi_{X_{t_{k-1}}|Y_{t_1:t_{k-1}}}(x_{t_{k-1}}|y)dx_{t_{k-1}}, \tag{2.38}$$

where $y := y_{\text{data},t_1:t_{k-1}}$ and $\pi_{X_{t_k}|X_{t_{k-1}}}$ is the distribution associated with the random variable X_{t_k} given the prediction model M and the state of the random variable $X_{t_{k-1}}$. The posterior distributions $\pi_{X_{t_k}|Y_{t_1:t_k}}$ are known as analysis distributions or filtering distributions in this context. Define the random variable associated with this posterior distribution as $\tilde{X}_{t_k} : \Omega \to \mathbb{R}^{N_X}$. This will be referred to as the analysis random variable. With this is mind, let the distribution $\pi_{\tilde{X}_{t_k}}$ be equivalent to the distribution $\pi_{X_{t_k}|Y_{t_1:t_k}}$. In a weather forecasting scenario, one is interested in obtaining analysis distributions for the future weather, based on real-time observations, y_{data}, from sources such as satellite data and in-situ measurements. There are a number of challenges surrounding the filtering problem in such practical scenarios.

Firstly, the forecast distributions are typically not known explicitly. Indeed, the normalizing integral constants of these distributions are also not typically known. In this case, it is required to approximate these distributions using ensembles and self-normalize within the ensemble to approximate the normalization constant. If the forecast distribution and likelihood are at least known to be Gaussian, there is an efficient approximate filtering algorithm known as the Ensemble Kalman Filter (EnKF) to find analysis distributions. For more details on this see Reich and Cotter [2015], Daley [1997]. However, commonly they are not. Therefore other algorithms have to be used, such as sequential importance sampling and particle filtering. This is because the EnKF does not converge to the true Bayesian posterior in these cases.

Secondly, the dimensions of the forecast distribution and the likelihood are typically large in these practical cases. This causes an issue known as degeneracy in an empirical approximation of the filtering distribution. It will be looked at in the next section.

2.7.2 Sequential importance sampling

The former of these challenges lead us on to the basic outline for many Bayesian filtering algorithms: sequential importance sampling. Earlier in the chapter, the ensemble forecast of the random variable X was defined by an i.i.d. sampled ensemble, $\{X^i\}_{i=1}^N \sim X$, which can form an approximation to the distribution of X, π_X. E.g.

$$\pi_X(x) \approx \frac{1}{N}\sum_{i=1}^{N} \delta_x(X^i). \tag{2.39}$$

Here, δ is the Dirac delta measure. Statistics can also be estimated from this ensemble as previously covered, e.g. Monte Carlo estimators. If X is a function of time (and possibly space), X_t, then this ensemble can be interpreted as an approximation to the forecast distribution in the previous section, π_{X_t}. It is referred to as the *forecast ensemble*, and likewise X_t is referred to as the *forecast random variable*.

Given some data y_{data,t_k}, at assimilation times t_k, for $k = 1, 2, 3, \ldots$, one can recursively define the so-called *importance weight* of the i'th ensemble member as

$$w^i_{t_k} = w^i_{t_{k-1}} \pi_{Y_{t_k}}(y_{\text{data},t_k} | X^i_{t_k}), \tag{2.40}$$

where $w^i_0 = 1/N$ and therefore one assumes all N ensemble members are sampled i.i.d. from $\pi_{X_{t_0}}$ at time $t_0 = 0$. Assume that the data is given by

$$y_{\text{data},t_k} = H(X^{\text{ref}}_{t_k}) + \phi, \tag{2.41}$$

where $H : \mathbb{R}^{N_X} \to \mathbb{R}^{N_Y}$ is some local operator, X^{ref}_t is some reference trajectory of X_t that one would like to forecast, and $\phi \sim N(0, R)$ is some Gaussian measurement error with mean zero and covariance R. Then the importance weight of the i'th ensemble member is

$$w^i_{t_k} = w^i_{t_{k-1}} \frac{1}{\sqrt{2\pi}|R|^{N_Y/2}} e^{-\frac{1}{2}\left(y_{\text{data},t_k} - H(X^i_{t_k})\right)^T R^{-1}\left(y_{\text{data},t_k} - H(X^i_{t_k})\right)}. \tag{2.42}$$

An approximation to the filtering posterior (analysis) distribution in (2.37) can then be given by a weighted sum of Dirac delta measures,

$$\pi_{\tilde{X}_{t_k}}(x | y_{\text{data},t_1:t_k}) = \pi_{X_{t_k}|Y_{t_1:t_k}}(x | y_{\text{data},t_1:t_k}) \approx \frac{\sum_{i=1}^N w^i_{t_k} \delta_x(X^i_{t_k})}{Z}, \tag{2.43}$$

where Z is some normalization constant. This is the basic sequential importance sampling approximation. A self-normalized sequential importance sampling approximation is given by

$$Z = \sum_{j=1}^{N} w_{t_k}^j,$$

such that

$$\int \pi_{\tilde{X}_{t_k}}(x|y_{\text{data},t_1:t_k})dx = 1.$$

Then, the approximation in (2.43) becomes

$$\pi_{\tilde{X}_{t_k}}(x|y_{\text{data},t_1:t_k}) \approx \frac{\sum_{i=1}^{N} w_{t_k}^i \delta_x(X_{t_k}^i)}{\sum_{j=1}^{N} w_{t_k}^j}. \tag{2.44}$$

Statistics of this distribution can be estimated too. The expectation of $g(\tilde{X}_{t_k})$, can be approximated by the weighted Monte Carlo estimate

$$\bar{X}_{N,t_k} = \sum_{i=1}^{N} \tilde{w}_{t_k}^i g(X_{t_k}^i), \tag{2.45}$$

where

$$\tilde{w}_{t_k}^i = \frac{w_{t_k}^i}{\sum_{j=1}^{N} w_{t_k}^j}.$$

This estimator converges in probability to the true analysis mean (in other words it is a consistent estimator) [Doucet and Johansen, 2011]. However, it is not unbiased, i.e.

$$\mathbb{E}[\bar{X}_{N,t_k}] \neq \mathbb{E}[g(\tilde{X}_{t_k})],$$

since the posterior approximation is self-normalized.

Sequential importance sampling is a simple methodology for constructing filtering estimates of posterior (analysis) statistics, and can be simulated in an on-line manner. For example, consider the very simple recursive algorithm:

(1) Propagate $\left\{X_{t_{k-1}}^i\right\}_{i=1}^N$ with prediction model M to time t_k.

(2) Update importance weights

$$w_{t_k}^i = w_{t_{k-1}}^i \pi_{Y_{t_k}}(y_{\text{data},t_k}|X_{t_k}^i),$$

for $i = 1, ..., N$.

(3) Normalize the weights using $\tilde{w}^i_{t_k} = w^i_{t_k} / \sum_{j=1}^N w^j_{t_k}$, for $i = 1, ..., N$.

(4) Estimate $\mathbb{E}[g(\tilde{X}_{t_k})]$ via (2.45), and iterate $k = k + 1$.

This is an example of a sequential Monte Carlo algorithm. Despite its simplicity, sequential importance sampling is not an ideal nonlinear data assimilation tool in practice. As the number of assimilation steps increase, the importance weights tend to degenerate. This means that one weight becomes one, and the rest take the value of zero. In this case, the filter becomes unstable. Degeneracy can also occur with a small number of assimilation steps when the number of degrees of freedom for the data, N_y, is large. The problem of degeneracy, both with the number of assimilation steps and the number of degrees of freedom, is demonstrated with the following example.

Degeneracy describes the increase in variation amongst weights. A suitable metric to evaluate this variation is the effective sample size, defined below:

$$N_{eff} = \frac{1}{\sum_{i=1}^N (\tilde{w}^i_{t_k})^2}$$

This can take values anywhere between 1 and N. When for any given n,

$$\begin{cases} \tilde{w}^i_{t_k} = 1, & i = n, \\ \tilde{w}^j_{t_k} = 0, & j \neq n, \end{cases} \tag{2.46}$$

the effective sample size takes the minimum value of 1, and the filter is completely degenerate. As an example, consider the following discrete system of X_{t_k}, for $k = 1, 2, ..., N_t$, with the initial condition $X_{t_0} = 0$:

$$X_{t_k} = X_{t_{k-1}} + \xi,$$

where $\xi \sim N(0, 0.01)$ and observations are given by

$$y_{\text{data},t_k} = X^{\text{ref}}_{t_k} + \phi,$$

with $\phi \sim N(0, 0.01)$, and some reference trajectory $X^{\text{ref}}_{t_k} \sim X_{t_k}$. A sequential importance sampling estimator of $\mathbb{E}[\tilde{X}_{t_k}]$, with $N = 500$, is simulated over $N_t = 50$ assimilation steps, and the effective sample size at each step is shown in Figure 5. The effective sample size

decays as the number of assimilation steps increase showing degeneracy in the ensemble importance weights.

Now consider using the (sequential) importance sampling estimator over one assimilation step, for the discrete system

$$X_{t_1} \sim N(\mu, \Sigma),$$

where μ is a zero vector length d, and $\Sigma = 0.01I$. Here I is the $d \times d$ identity matrix. Observations are given by

$$y_{\text{data},t_1} = X^{\text{ref}}_{t_1} + \phi,$$

with $\phi \sim N(\mu, \Sigma)$, and some reference trajectory $X^{\text{ref}}_{t_1} \sim X_{t_1}$. The importance sampling estimator is taken over 500 ensemble members, and estimates $\mathbb{E}[\tilde{X}_{t_1}]$, over increasing number of degrees of freedom, d. The effective sample sizes after one assimilation step for each of the simulations is shown in Figure 6. As with increasing number of assimilation steps in the previous experiment, the effective sample size decays as the number of degrees of freedom increases.

These two forms of degeneracy can be avoided with resampling and localisation techniques respectively. The latter is an active area of research, and is a major extension to the type of filter investigated later in the thesis [Cheng and Reich, 2013]. The former is a very well known solution to degeneracy with an increasing number of assimilation steps. Particle filters are a general framework to incorporate resampling into sequential importance sampling estimators. Different types of resampling are the main difference between variants of the particle filter [Douc et al., 2005].

2.7.3 Particle filtering

Particle filtering is a modified framework to the sequential importance sampling methodology described in the last section. The use of the word 'particle' refers to the individual members of the ensemble approximating the posterior distribution after assimilation. At a certain assimilation step t_k, it aims to replace the forecast ensemble $\left\{X^i_{t_k}\right\}_{i=1}^N$ and importance weights $\left\{\tilde{w}^i_{t_k}\right\}_{i=1}^N$, with an evenly weighted, 'resampled' *analysis ensemble* $\left\{\tilde{X}^i_{t_k}\right\}_{i=1}^N$. The analysis ensemble provides an ensemble forecast of the aforementioned analysis distribution, $\pi_{\tilde{X}_{t_k}}$. The new estimate for the first moment of $\tilde{X}_{t_k}$ at each assimilation step as

supposed to (2.45) is then

$$\bar{X}_{N,t_k} = \frac{1}{N}\sum_{i=1}^{N} \tilde{X}^i_{t_k}. \tag{2.47}$$

This replacement is concerned with replicating highly weighted ensemble members whilst removing low weighted members in the analysis ensemble. This replacement should satisfy

$$\mathbb{E}\left[\sum_{i=1}^{N} \tilde{w}^i_{t_k} X^i_{t_k}\right] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} \tilde{X}^i_{t_k}\right].$$

This means that the replacement is unbiased. Despite being unbiased, many resampling schemes contribute variance to the estimate of the analysis mean; the consequences of this on the theoretical error bounds of the estimate are investigated in Chopin [2004]. As previously mentioned, it is this stage of the particle filtering framework in which most particle filter variants differ. There are many different types of resampling, such as multinomial and residual resampling schemes [Douc et al., 2005].

Multinomial resampling

Multinomial resampling is the most simple resampling scheme. It is very cheap, however it contributes the most variance. The analysis ensemble members are simply sampled from the multinomial distribution with N draws and probability vector $(\tilde{w}^1_{t_k}, ..., \tilde{w}^N_{t_k})$, defined as $Mult(N; \tilde{w}^1_{t_k}, ..., \tilde{w}^N_{t_k})$. To sample from this distribution, sample $u \sim U[0,1]$ and resample $X^j_{t_k}$ where j is given by

$$j = \min\left\{ n \in (1, ..., N); \sum_{i=1}^{n} \tilde{w}^i_{t_k} \geq u \right\}.$$

Residual resampling

Residual resampling aims to produce a proportion of the analysis ensemble deterministically from the weighted forecast ensemble. It then uses multinomial resampling to resample the rest of the ensemble. More specifically, if an ensemble member $X^i_{t_k}$ has a weight of $\tilde{w}^i_{t_k}$, then it should be resampled an expected number of times: $\tilde{w}^i_{t_k} N$. However an ensemble member cannot be resampled a non-integer amount of times. Therefore one should take the

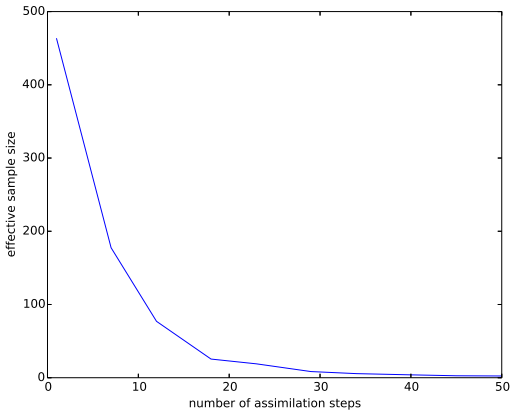


Figure 5: Effective sample size for the sequential importance sampling estimator of $\mathbb{E}[\tilde{X}_{t_k}]$, with $N = 500$, against the number of assimilation steps. This decays exponentially with an increasing number of assimilation steps.

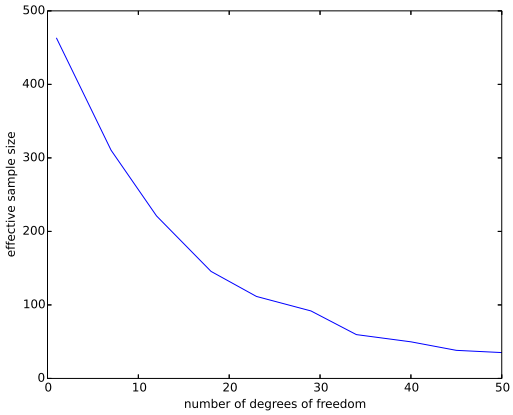


Figure 6: Effective sample size for the sequential importance sampling estimator of $\mathbb{E}[\tilde{X}_{t_1}]$, with $N = 500$, against the number of degrees of freedom. This decays exponentially with an increasing number of degrees of freedom.

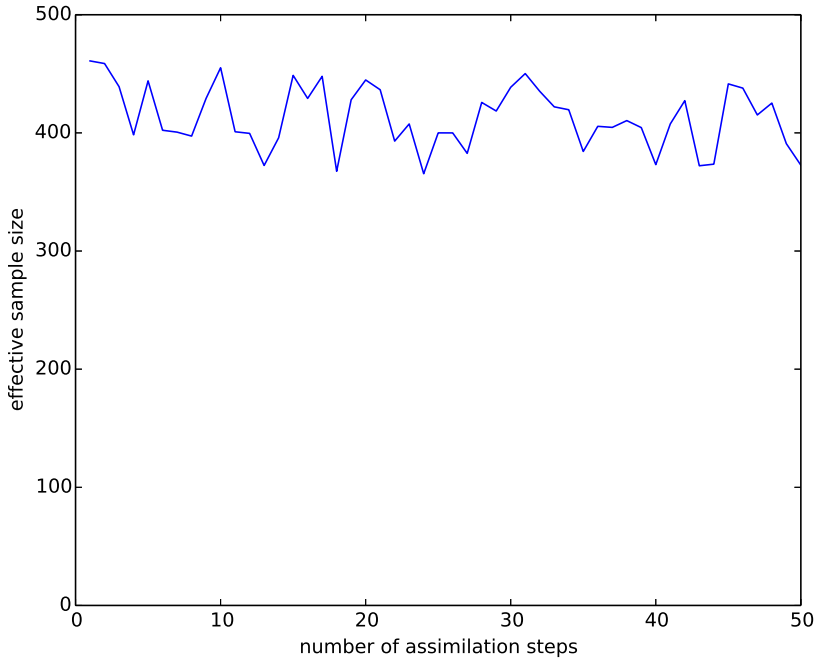


Figure 7: Same as in Figure 5 only with a particle filtering estimator for $\mathbb{E}[\tilde{X}_{t_k}]$, where resampling is implemented on every resampling step. The effective sample size does not decay with an increasing number of assimilation steps.

lower integer bound of $\tilde{w}^i_{t_k} N$ and guarantee that many copies of the i'th ensemble member to appear in the analysis ensemble. With this in mind, let $\lfloor \tilde{w}^i_{t_k} N \rfloor$ copies of the i'th ensemble member be in the analysis ensemble, and then compute

$$N' = N - \sum_{i=1}^{N} \lfloor \tilde{w}^i_{t_k} N \rfloor.$$

Then draw the remaining N' analysis ensemble members via a multinomial distribution, $Mult(N'; \tilde{v}^1_{t_k}, ..., \tilde{v}^N_{t_k})$, where

$$\tilde{v}^i_{t_k} = \frac{\tilde{w}^i_{t_k} - \lfloor \tilde{w}^i_{t_k} N \rfloor}{\sum_{j=1}^{N} \left(\tilde{w}^j_{t_k} - \lfloor \tilde{w}^j_{t_k} N \rfloor \right)},$$

for $i = 1, ..., N$. Whilst the multinomial resampling scheme has a small probability that a highly weighted ensemble member will not get resampled, the residual resampling scheme tries to guarantee that it will be resampled a certain number of times at least. Due to the deterministic selection of a proportion of the analysis ensemble, the variance contributed to the overall posterior estimate from resampling is less than that of the multinomial resampling scheme [Hol et al., 2006].

A general algorithm for the particle filter

Using the resampling modification to the sequential importance sampling algorithm in the previous section, the general particle filter algorithm is shown below.

(1) Propagate $\left\{ X^i_{t_{k-1}} \right\}_{i=1}^{N}$ with prediction model M to time t_k.

(2) Update importance weights

$$w^i_{t_k} = w^i_{t_{k-1}} \pi_{Y_{t_k}}(y_{\text{data},t_k} | X^i_{t_k}), \tag{2.48}$$

for $i = 1, ..., N$.

(3) Normalize the weights using $\tilde{w}^i_{t_k} = w^i_{t_k} / \sum_{j=1}^{N} w^j_{t_k}$, for $i = 1, ..., N$.

(4) Resample the analysis ensemble $\left\{ \tilde{X}^i_{t_k} \right\}_{i=1}^{N}$ from the weighted forecast ensemble $\left\{ X^i_{t_k} \right\}_{i=1}^{N}$.

(5) Reset the importance weights to even weights, $w^i_{t_k} = 1/N$, for $i = 1, ..., N$.

(6) Reset the forecast ensemble $X^i_{t_k} = \tilde{X}^i_{t_k}$, for $i = 1, ..., N$, and iterate $k = k + 1$.

The particle filter is now used to create an estimator for $\mathbb{E}[\tilde{X}_{t_k}]$ for increasing k, where X_{t_k} is as in the examples in Sec. 2.7.2. The particle filter is run with 500 samples. The effective sample size for each assimilation step is shown in Figure 7. Unlike with the sequential importance sampling estimator (where resampling is not implemented), the effective sample size does not decay and stays relatively constant.

Resampling is not required to occur on every assimilation step, in fact it is detrimental to the particle filter if resampling is done when it is not needed (i.e. weights are approximately even) [Doucet and Johansen, 2011]. One can use the effective sample size to determine whether or not to resample at a particular assimilation step. For example, if one updates the importance weights $\left\{\tilde{w}^i_{t_k}\right\}_{i=1}^N$, one should resample only if

$$N_{eff} < \tau N,$$

where $\tau \in [1/N, 1]$. If $\tau = 1/N$, the filter will never resample (as $N_{eff} \geq 1$) and corresponds to sequential importance sampling. On the other hand, if $\tau = 1$, the filter resamples on every assimilation step, corresponding to the algorithm above.

2.7.4 Ensemble transform methods for Bayesian data assimilation

There has been a great deal of research into linear ensemble transform filters (LETFs) recently, and their use in a Bayesian approach to nonlinear filtering. A particular type of linear ensemble transform filter, called the ensemble transform particle filter (ETPF), replaces the resampling step in the particle filter algorithm with a deterministic transform step.

Linear ensemble transform filters

Linear ensemble transform filters (LETFs) are a framework for implementing Bayesian data assimilation [Reich and Cotter, 2015]. Consider the following definition.

Definition 11 (Linear ensemble transform filters)**.** *A LETF forms the analysis ensemble* $\left\{\tilde{X}^i_{t_k}\right\}_{i=1}^N$ *at time* t_k*, for* $k = 1, 2, 3, ...$*, from a linear transform to the forecast ensemble* $\left\{X^i_{t_k}\right\}_{i=1}^N$*, i.e.*

$$\tilde{X}^j_{t_k} = \sum_{i=1}^N D_{i,j} X^i_{t_k}.$$

Here, $D \in \mathbb{R}^{N\times N}$ depends on the observations, y_{data}, in some way and satisfies

$$\sum_{i=1}^{N} D_{i,j} = 1, \tag{2.49}$$

for each $j = 1, ..., N$.

For example, the EnKF is a type of LETF, where one uses a transform to the analysis ensemble to estimate the posterior mean and covariance. The transform is as follows:

$$\tilde{X}^j_{t_k} = \sum_{i=1}^{N} D^{\text{EnKF}}_{i,j} X^i_{t_k},$$

where

$$D^{\text{EnKF}}_{i,j} = \delta_{i,j} - \frac{1}{N-1}\left(\left(X^i_{t_k} - \bar{X}^{\text{EnKF}}_{N,t_k}\right)^T (C+R)^{-1}(X^j_{t_k} - y_{\text{data},t_k})\right),$$

and C is the sample covariance matrix of X_{t_k} [Reich and Cotter, 2013]. Here it is assumed that the operator H is the identity matrix and that $\bar{X}^{\text{EnKF}}_{N,t_k}$ is the forecast mean, namely $\frac{1}{N}\sum_{i=1}^{N} X^i_{t_k}$.

The ensemble transform particle filter

The ensemble transform particle filter (ETPF) is a particularly interesting variant of the LETF. It uses a deterministic transformation that takes one from the weighted forecast ensemble $\left\{X^i_{t_k}\right\}_{i=1}^N$ in a particle filter to the evenly weighted analysis ensemble $\left\{\tilde{X}^i_{t_k}\right\}_{i=1}^N$. Using the framework for the LETF described in the previous section, one can write this transform as

$$\tilde{X}^j_{t_k} = \sum_{i=1}^{N} P_{i,j} X^i_{t_k}, \tag{2.50}$$

for $j = 1, ..., N$. The matrix $P \in \mathbb{R}^{N\times N}$, can be written as a product NT, where T satisfies $\sum_{i=1}^{N}\sum_{j=1}^{N} T_{i,j} = 1$. Here, T is often referred to as a coupling matrix. It also has to satisfy the marginal constraint

$$\sum_{i=1}^{N} T_{i,j} = \frac{1}{N}, \tag{2.51}$$

given the condition in (2.49) of $\sum_{i=1}^{N} P_{i,j} = 1$ and recalling that the analysis ensemble members are evenly weighted. The transform in (2.50) allows a new estimator for statistics of the posterior to be defined, as supposed to (2.45):

$$\bar{X}_{N,t_k} = \frac{1}{N}\sum_{i=1}^{N} g(\tilde{X}^i_{t_k}). \tag{2.52}$$

To preserve the mean of the first moment (g being the identity) in the weighted form of (2.45), the marginal constraint

$$\sum_{j=1}^{N} T_{i,j} = \tilde{w}^i_{t_k}, \tag{2.53}$$

is required such that

$$\frac{1}{N}\sum_{j=1}^{N} \tilde{X}^j_{t_k} = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{N} P_{i,j} X^i_{t_k} = \sum_{j=1}^{N}\sum_{i=1}^{N} T_{i,j} X^i_{t_k} = \sum_{i=1}^{N} \tilde{w}^i_{t_k} X^i_{t_k}, \tag{2.54}$$

from [Reich, 2013]. Denote the space of couplings that such a T belongs to by $\mathcal{T} \in \mathbb{R}^{N\times N}$. The transform T associated with the ETPF solves the Monge-Kantorovich problem [Villani, 2008], and is found by using optimal transportation. The so-called Monge-Kantorovich problem concerns finding a $T \in \mathcal{T}$ that minimizes a certain cost metric,

$$\arg\min_{T\in\mathcal{T}} T_{i,j} \left\| X^i_{t_k} - X^j_{t_k} \right\|^2. \tag{2.55}$$

The solution minimises the expected Euclidean difference between X_{t_k} and $\tilde{X}_{t_k}$ and in turn maximises the covariance between them via

$$\mathbb{E}_T[\|X_{t_k} - \tilde{X}_{t_k}\|^2] = \mathbb{E}[\|X_{t_k}\|^2] + \mathbb{E}[\|\tilde{X}_{t_k}\|^2] - 2\mathbb{E}[X_{t_k}]^T\mathbb{E}[\tilde{X}_{t_k}] - 2Tr\left(\mathrm{Cov}_T[X_{t_k}, \tilde{X}_{t_k}]\right).$$

Here, the statistical operators $\mathbb{E}_T$ and Cov_T are the expectation and covariance with respect to the coupling T. This coupling therefore minimises the variance of the difference between X_{t_k} and $\tilde{X}_{t_k}$, i.e. $\mathbb{V}_T[X_{t_k} - \tilde{X}_{t_k}]$. Whereas the classical particle filter resampling step adds variance to the posterior approximation, this transformation attempts to minimize the variance added to such an approximation. The Monge-Kantorovich problem in practice can be solved using optimal transportation algorithms.

These algorithms vary in cost, and can be quite expensive. In one-dimensional cases of

one degree of freedom, where $X_{t_k} \in \mathbb{R}$, there is a cheap algorithm that solves the problems in $\mathcal{O}(N \log N)$ complexity; for an example of the psuedocode see Reich and Cotter [2015]. However, in higher dimensional cases, the cost is solely dependent on N and can only be solved in $\mathcal{O}(N^3 \log N)$ complexity. This expense can therefore be a computational bottleneck of the ETPF in some cases; a technique called localisation can be employed here to provide some complexity relief (see Sec. 2.7.7).

The transformation in (2.50) will now be studied in more detail. Let the ensembles $\left\{Z_1^i\right\}_{i=1}^N$ and $\left\{Z_2^i\right\}_{i=1}^N$ be i.i.d. random variables associated with the distributions π_{Z_1} and π_{Z_2} respectively. Furthermore suppose they are assigned weights $\left\{p^i\right\}_{i=1}^N$ and $\left\{q^i\right\}_{i=1}^N$ respectively, by importance sampling *via* a third probability distribution π_{Z_3}, all absolutely continuous with respect to the Lesbegue measure. Suppose one wishes to resample $\left\{Z_2^i\right\}_{i=1}^N$ with weights $\left\{q^i\right\}_{i=1}^N$, to obtain a new ensemble $\left\{\tilde{Z}_2^i\right\}_{i=1}^N$ with weights $\left\{p^i\right\}_{i=1}^N$. To do this, consider the following transform (a general version of the one in (2.50)):

$$\tilde{Z}_2^j = \sum_{i=1}^N T_{i,j}(p^j)^{-1} Z_2^i, \tag{2.56}$$

for $j = 1, ..., N$. Here, $T \in \mathcal{T}$ is a coupling matrix satisfying

$$\sum_{i=1}^N T_{i,j} = p^j, \qquad \sum_{j=1}^N T_{i,j} = q^i.$$

Notice that this is equivalent to the transform used in (2.50) with $p^i = 1/N$, for $i = 1, ..., N$. In Reich and Cotter [2015], the following was proved. If the ensemble $\left\{\tilde{Z}_2^i\right\}_{i=1}^N$ is computed via the transform in (2.56), then the maps $\Psi_N : Z_1^i \to \tilde{Z}_2^i$, given by

$$\tilde{Z}_2^i = \Psi_N(Z_1^i), \qquad i = 1, ..., N,$$

weakly converge to a map $\Psi : \mathbb{R}^{N_X} \to \mathbb{R}^{N_X}$ as $N \to \infty$. In addition to this, the random variable given by $\tilde{Z}_2 = \Psi(Z_1)$ is associated with the distribution π_{Z_2}. Hence, for arbitrary test functionals g, weak convergence for estimates to $\mathbb{E}[g(Z_2)]$ can be obtained. I.e.

$$\sum_{i=1}^N p^i g(\tilde{Z}_2^i) \to \sum_{i=1}^N q^i g(Z_2^i) \to \mathbb{E}[g(Z_2)], \tag{2.57}$$

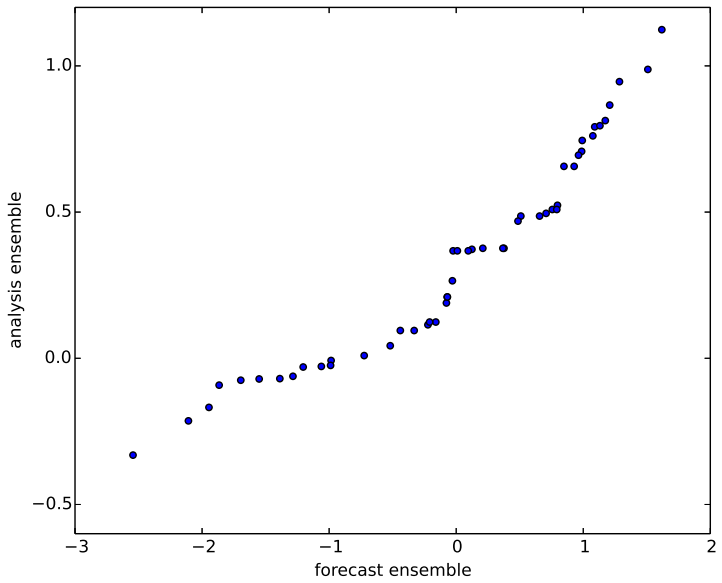


Figure 8: The forecast ensemble $\left\{X_{t_k}^i\right\}_{i=1}^N \sim N(0,1)$ and analysis ensemble $\left\{\tilde{X}_{t_k}^i\right\}_{i=1}^N$, after importance weighting with an observation sampled from the distribution $N(0.35, 0.1)$. They are positively correlated due to the transformation, from forecast to analysis ensemble in (2.50), that maximises the covariance between them.

as $N \to \infty$. Returning to the transformation in (2.50), this theory results in weak convergence for the estimate to posterior statistics in (2.52), so that

$$\frac{1}{N}\sum_{i=1}^{N} g(\tilde{X}_{t_k}^i) \to \sum_{i=1}^{N} \tilde{w}_{t_k}^i g(X_{t_k}^i) \to \mathbb{E}[g(\tilde{X}_{t_k})], \tag{2.58}$$

as $N \to \infty$.

As a demonstration of the transformation in the ETPF, consider an ensemble of $N = 50$ particles sampled from the forecast distribution $X_{t_k} \sim N(0,1)$. The observation is sampled from the distribution $N(0.35, 0.1)$. The forecast ensemble is weighted, and the transformation in (2.50) is used to obtain an evenly weighted analysis ensemble, $\left\{\tilde{X}_{t_k}^i\right\}_{i=1}^N$. The forecast and analysis ensembles are shown in Figure 8. Note that they are positively correlated due to the coupling, that solved the optimal transport problem, maximising the covariance between them.

An algorithm for the ensemble transform particle filter

This ensemble transform particle filtering framework inspires a modification to the algorithm presented in Sec. 2.7.3.

(1) Propagate $\left\{X^i_{t_{k-1}}\right\}^N_{i=1}$ with prediction model M to time t_k.

(2) Update importance weights

$$w^i_{t_k} = \pi_{Y_{t_k}}(y_{\text{data},t_k}|X^i_{t_k}), \tag{2.59}$$

for $i = 1, ..., N$.

(3) Normalize the weights using $\tilde{w}^i_{t_k} = w^i_{t_k} / \sum_{j=1}^N w^j_{t_k}$, for $i = 1, ..., N$.

(4) Using a coupling matrix T, found by solving (2.55) whilst satisfying (2.51) and (2.53), compute the analysis ensemble $\left\{\tilde{X}^j_{t_k}\right\}^N_{j=1}$ by

$$\tilde{X}^j_{t_k} = \sum_{i=1}^N NT_{i,j}X^i_{t_k}.$$

(5) Reset the forecast ensemble $X^i_{t_k} = \tilde{X}^i_{t_k}$, for $i = 1, ..., N$, and iterate $k = k + 1$.

A matrix interpretation of the ensemble transform particle filter

In order to understand the extensions of the ETPF that are investigated in the following two sections, it is beneficial to frame the associated transform and posterior approximations in matrix algebra. Firstly, note that the forecast ensemble $\left\{X^i_{t_k}\right\}^N_{i=1}$ can be written as the matrix $\mathbf{X}_{t_k} = (X^1_{t_k}, ..., X^N_{t_k})^T \in \mathbb{R}^{N_X \times N}$. In addition to this, by writing the importance weights as $\tilde{\mathbf{w}}_{t_k} = (\tilde{w}^1_{t_k}, ..., \tilde{w}^N_{t_k})^T \in \mathbb{R}^{N \times 1}$, one can re-write the weighted posterior mean in (2.45) as

$$\bar{X}_{N,t_k} = \mathbf{X}_{t_k}\tilde{\mathbf{w}}_{t_k}.$$

The analysis ensemble $\tilde{\mathbf{X}}_{t_k} = (\tilde{X}^1_{t_k}, ..., \tilde{X}^N_{t_k})^T \in \mathbb{R}^{N_X \times N}$, can be found by transforming the forecast ensemble like so

$$\tilde{\mathbf{X}}_{t_k} = \mathbf{X}_{t_k}P. \tag{2.60}$$

In addition to this, by defining $\mathbf{1} = (1, ..., 1)^T \in \mathbb{R}^{N\times 1}$, one can re-write the ETPF estimator to the analysis mean as

$$\bar{X}_{N,t_k} = \frac{1}{N}\tilde{\mathbf{X}}_{t_k}\mathbf{1}.$$

From the last section, $T = (1/N)P$ allows

$$\frac{1}{N}T\mathbf{1} = \tilde{\mathbf{w}}_{t_k}$$

to be written and therefore the transformation is *first-order accurate*:

$$\frac{1}{N}\tilde{\mathbf{X}}_{t_k}\mathbf{1} = \mathbf{X}_{t_k}\tilde{\mathbf{w}}_{t_k}.$$

This is equivalent to (2.54). Here, first-order accuracy of the transformation refers to the weighted forecast mean and the analysis mean being equal.

2.7.5 The Sinkhorn approximation to an optimal coupling

The optimal transport algorithm in the transform mentioned in the last section can be expensive when $N_X > 1$. Fortunately, there is an iterative algorithm that can achieve approximations to a slightly modified problem for significantly less complexity: the Sinkhorn algorithm [Cuturi, 2013]. The regularized cost matrix, $\mathbf{K} \in \mathbb{R}^{N\times N}$, is defined as

$$\mathbf{K}_{i,j} = \exp\left(-\lambda C(X^i_{t_k}, X^j_{t_k})\right), \tag{2.61}$$

for $i, j = 1, ..., N$, where λ is a positive constant and $C(x, y) = \|x - y\|^2$. One now finds the transform matrix $T_{i,j}$ that minimises the modified problem

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\left(T_{i,j}C(X^i_{t_k}, X^j_{t_k}) + \frac{1}{\lambda}T_{i,j}\log\left(\frac{T_{i,j}}{T^0_{i,j}}\right)\right),$$

where $T^0 = \tilde{\mathbf{w}}_{t_k}\mathbf{1}^T$ [Cuturi, 2013, de Wiljes et al., 2016]. As $\lambda \to \infty$, the modified problem tends back to the full optimal transport problem in the last section. The dual fixed point iterations of the $N \times 1$ vectors $\mathbf{u}$ and $\mathbf{v}$ are used:

$$\left(\frac{N\tilde{\mathbf{w}}_{t_k}}{\mathbf{Kv}}\right) \to \mathbf{u}, \qquad \left(\frac{1}{\mathbf{Ku}}\right) \to \mathbf{v}.$$

In practice, one iterates $\mathbf{u}$ first, starting with $\mathbf{v} = \mathbf{1}$, until $\left\|\tilde{\mathbf{w}}^*_{t_k} - \tilde{\mathbf{w}}_{t_k}\right\| < \epsilon$ is achieved, where $\tilde{\mathbf{w}}^*_{t_k} = T\mathbf{1}$ and

$$T = \text{diag}(\mathbf{u})\mathbf{K}\text{diag}(\mathbf{v}).$$

As λ becomes large, the algorithm takes longer to converge.

2.7.6 A second-order accurate correction to the ensemble transform particle filter

The Sinkhorn algorithm is used within de Wiljes et al. [2016], alongside a *second-order accurate* correction to the transform matrix, T, for the ETPF. Here, second-order accuracy refers to the weighted forecast covariance and the analysis covariance being equal. This is required as the analysis ensemble underestimates the forecast ensemble spread when the Sinkhorn algorithm is used, and in the long-time limit could cause the filter to collapse. The study proposed a correction matrix, Δ, to the transform matrix, $P = NT$, in (2.60). It satisfies $\Delta\mathbf{1} = 0$, such that (2.60) is still first-order accurate, and in addition satisfies the following condition:

$$N\left(\text{diag}(\tilde{\mathbf{w}}_{t_k}) - \tilde{\mathbf{w}}_{t_k}\tilde{\mathbf{w}}^T_{t_k}\right) - BB^T = B\Delta^T + \Delta B^T + \Delta\Delta^T,$$

where $B = P - \tilde{\mathbf{w}}_{t_k}\mathbf{1}^T$. This condition allows the analysis covariance and the weighted forecast covariance matrices to be equal, thus preserving ensemble spread. The corrected transformation therefore replaces (2.60) with

$$\tilde{\mathbf{X}}_{t_k} = \mathbf{X}_{t_k}(P + \Delta). \tag{2.62}$$

The correction matrix Δ, assumed to be symmetric, can be found by numerically solving a continuous-time algebraic Riccati equation

$$A = B\Delta^T + \Delta B^T + \Delta\Delta, \tag{2.63}$$

where $A = N(\text{diag}(\tilde{\mathbf{w}}_{t_k}) - \tilde{\mathbf{w}}_{t_k}\tilde{\mathbf{w}}_{t_k}^T) - BB^T$. The following Euler discretization, with a step-size of δt, can be used to numerically solve (2.63):

$$\Delta_{(l+1)\delta t} = \Delta_{l\delta t} + \delta t \left(-B\Delta_{l\delta t} - \Delta_{l\delta t}B^T + A - \Delta_{l\delta t}\Delta_{l\delta t}\right).$$

The initial condition $\Delta_0 = \mathbf{0}$ is used and the numerical method is stopped when

$$\left\|\Delta_{(l+1)\delta t} - \Delta_{l\delta t}\right\|_\infty < 10^{-3}.$$

Take the final iterate $\Delta_{(l+1)\delta t}$ to be the Δ used in (2.62). Due to this correction, this filter is more stable than the standard ETPF. In addition to this, the Sinkhorn algorithm in the last section has a complexity of $\mathcal{O}(N^2)$, and so is a reduction from that of solving the optimal transport problems using exact algorithms.

This extension of the ETPF is not considered further, given that the computational cost of solving the optimal transport problems within the ETPF framework is not found to hinder the benchmark results presented in this thesis. Either this is due to a reasonably low order of magnitude of sample sizes being used, as in Chapters 5 and 6, or because of localisation, a technique that will be discussed now.

2.7.7 Localisation

Localisation, in the context of filtering, is used to stabilise the filter in the case of high-dimensional state spaces and alleviate degeneracy. It is frequently used in the EnKF for high-dimensional and spatially-extended systems [Anderson, 2012], and can also be applied to the ETPF [Cheng and Reich, 2013]. The type of localisation used alongside the ETPF is known as 'R-localisation' or 'radius localisation' as it assumes that only a localised area within a particular radius of each state component is considered during assimilation.

In addition to stabilising the ETPF, localisation can also be used to alleviate the high computational cost of the optimal transport problem in the transformation stage. Both benefits will be discussed in the following two sections. An important thing to remember about localisation is that using it will result in inconsistent estimators for the analysis statistics $\mathbb{E}[g(\tilde{X}_{t_k})]$. That is to say that the claim in (2.58) does not hold. Despite this, empirical results suggest that the ETPF with localisation implemented can effectively compensate between accuracy and stability in some systems, including the chaotic and highly nonlinear

Lorenz equations [Cheng and Reich, 2013].

Cost metric localisation

In the most extreme form, applying localisation to the ETPF means that one can reduce the computational cost of constructing a multivariate coupling matrix T to the cost of constructing N_X separate univariate coupling matrices, T^m, for $m = 1, ..., N_X$, for each component of X_{t_k}. More generally, localisation allows one to construct an individual transformation in (2.50) for each of the N_X components of a multivariate X_{t_k}. The modification to the ETPF framework required for localisation, concerns constructing local cost metrics instead of the one in (2.55) that are based on Euclidean differences of local components only. The alternative cost metrics use a localisation matrix, alongside a user-defined radius parameter, $r_{loc,c}$, to determine these local components. A simple example of a localisation matrix C [Cheng and Reich, 2013] that describes an idealised decay in spatial correlations amongst state components of X_{t_k} could be

$$C_{m,n} = \begin{cases} 1 - \frac{1}{2}\left(\frac{s_{m,n}}{r_{loc,c}}\right), & \left(\frac{s_{m,n}}{r_{loc,c}}\right) \leq 2, \\ 0, & \text{otherwise.} \end{cases} \tag{2.64}$$

Here $m, n = 1, \ldots, N_X$ are the indices of the spatial components of X_{t_k}. Let $\wp(m)$ denote the spatial position of component m of X_{t_k}, and then define

$$s_{m,n} = \|\wp(m) - \wp(n))\|_2. \tag{2.65}$$

Refer to Reich and Cotter [2015] for a similar example of $s_{m,n}$, where spatial periodicity of the domain of X_{t_k} is accounted for. This can be used to decompose the cost metric in (2.55) into N_X separate metrics. For each component $m = 1, ..., N_X$ in X_{t_k}, one finds the coupling matrix $T^m \in \mathcal{T}^m \in \mathbb{R}^{N \times N}$, that satisfies

$$\sum_{i=1}^{N} T^m_{i,j} = \frac{1}{N}, \qquad \sum_{j=1}^{N} T^m_{i,j} = \tilde{w}^i_{t_k}(m), \tag{2.66}$$

and solves

$$\arg \min_{T^m \in \mathcal{T}^m} \sum_{i=1}^{N} \sum_{j=1}^{N} T^m_{i,j} f_m(X^i_{t_k}, X^j_{t_k}), \tag{2.67}$$

where

$$f_m(X^i_{t_k}, X^j_{t_k}) = \sum_{n=1}^{N_X} C_{m,n}\big(X^i_{t_k}(n) - X^j_{t_k}(n)\big)^2. \tag{2.68}$$

Here, $X(m)$ is the m'th component of the random variable X, and $\tilde{w}_{t_k}(m)$ is a separate importance weight for each component of X_{t_k}. This technicality will be useful in the next section, but with simplicity in mind for now, take $\tilde{w}_{t_k}(m) = \tilde{w}_{t_k}$, for $m = 1, ..., N_X$. Using the transformation

$$\tilde{X}^j_{t_k}(m) = \sum_{i=1}^{N} NT^m_{i,j} X^i_{t_k}(m), \tag{2.69}$$

for $m = 1, ..., N_X$, to compute the components of the analysis ensemble members, one can define an approximation to posterior statistics in the usual way. I.e.

$$\bar{X}^N_{t_k} = \frac{1}{N}\sum_{j=1}^{N} g(\tilde{X}^j_{t_k}). \tag{2.70}$$

Note that the cost functions in (2.68) no longer achieve the minimum of (2.55). This means that implementing localisation results in inconsistent estimates of posterior statistics. When $r_{loc,c} = 0$, exhibiting the most computationally efficient scenario, one is required to solve N_X univariate optimal transport problems. Therefore this is equivalent to transforming all components individually. One can simply use the cheap univariate algorithm described in Reich and Cotter [2015], with a complexity of $\mathcal{O}(Nlog(N))$, for these transformations. In total, for all N_X transport problems, this results in an overall complexity of $\mathcal{O}(N_X Nlog(N))$. In practice, when $r_{loc,c} = 0$, one should re-order the component sets in the analysis ensemble members into the rank structure of the forecast ensemble members. This preserves the approximate copula structure of the forecast ensemble [Schefzik et al., 2013]. Such extreme localisation is useful when the computational cost of the forward model, M, is heavily dominated by the complexity of solving the optimal transport problems.

Likelihood localisation

In Sec. 2.7.2 it was found that degeneracy within the importance weights associated with the forecast ensemble $\{X^i_{t_k}\}_{i=1}^N$ can be caused by the high dimension of the state space. A localisation method, slightly different to the one explained above, can prevent this degeneracy in the importance weights. It is implemented by modifying the likelihood of each

forecast ensemble member in $\left\{X^i_{t_k}\right\}^N_{i=1}$. The degeneracy it alleviates is an issue inherent to sequential Monte Carlo (SMC) methods in general for high-dimensional systems [Rebeschini and Van Handel, 2013]. Standard SMC methods, with computationally feasible sample sizes, fail to track signals / reference trajectories of high-dimensional systems due to exponentially decaying effective sample size. Localisation in the likelihood can not directly be applied to standard particle filters, due to the lack of spatial regularity in random resampling schemes [Cheng and Reich, 2013]. There have been methods proposed to get around this issue, such as tempering of likelihoods and spatial resampling [Beskos et al., 2014]. These methods typically concentrate on maintaining the theoretical properties of particle filters, such as consistency, whilst alleviating the curse of dimensionality. On another hand, the form of the linear transform in (2.50) for the ETPF can provide sufficient spatial regularity to produce analysis ensembles under localisation in the likelihood [Reich and Cotter, 2015]. This is done via the following modification to the importance weight update of the forecast ensemble members in the ETPF. Here it is assumed that $N_X = N_Y$ and observations are given by (2.41). For each component, $m = 1, ..., N_X$, of the i'th forecast ensemble member, one can generate a separate importance weight, $\tilde{w}^i_{t_k}(m)$,

$$\tilde{w}^i_{t_k}(m) \propto \frac{1}{\sqrt{2\pi}|R|^{N_X/2}} e^{-\frac{1}{2}\left(H(X^i_{t_k})-y_{\text{data},t_k}\right)^T(\tilde{C}_m)R^{-1}\left(H(X^i_{t_k})-y_{\text{data},t_k}\right)}. \tag{2.71}$$

Let $\wp^H(m)$ denote the spatial position of component m of $H(X_{t_k})$ and also

$$s^H_{m,n} = \left\|\wp^H(m) - \wp^H(n)\right\|_2.$$

Then define the diagonal matrices $\tilde{C}_m$, for $m = 1, ..., N_X$, by

$$(\tilde{C}_m)_{n,n} = \begin{cases} 1 - \frac{1}{2}\left(\frac{s^H_{m,n}}{r_{loc,R}}\right), & \left(\frac{s^H_{m,n}}{r_{loc,R}}\right) \leq 2, \\ 0, & \text{otherwise}, \end{cases} \tag{2.72}$$

for $n = 1, ..., N_X$ and where $r_{loc,R}$ is another radius parameter (independent of $r_{loc,c}$). Defining a separate importance weight for each component of X_{t_k}, motivates the technicality of using different importance weights in the optimal transport problem for each individual component in (2.66). See Anderson [2012] for details of a similar use of localisation within the EnKF.

Chapter 3

Scoring multilevel Monte Carlo forecasts

This chapter is based upon Gregory and Cotter [2017a]

Chapter synopsis

This chapter concerns the evaluation of forecasts from multilevel Monte Carlo approximations to statistics of discretized random variables, away from the traditional mean square error (MSE) metric. It was discussed in the introduction that ensemble forecasts of probability distributions, associated with random variables, can be verified to evaluate their predictive performance, relative to observations and measurements. Observations were also utilised in the previous chapter to refine forecasts via ensemble data assimilation methods. For standard N-member ensemble forecasts, where Monte Carlo can be used to compute statistics of the random variable in question, their evaluation is a well researched topic in literature [Gneiting

and Raftery, 2007, Ferro et al., 2008, Gneiting et al., 2007]. These verification techniques are frequently referred to as *scoring rules* and try to answer the following question:

How much should one trust forecasts, and is the uncertainty within the forecast representative of the uncertainty in the underlying system that one is trying to forecast?

This chapter investigates how to implement these verification techniques when the ensemble forecast is not given by a single forecast at a certain resolution, but by a hierarchy of ensembles from different resolutions of the random variable, utilised in the multilevel Monte Carlo method. This would enable scoring rules to verify the effectiveness of these types of forecasts, and is the aim of this chapter. The assumption that the random variable of interest is univariate, or is the scalar functional of a multivariate random variable, is made throughout this chapter.

3.1 Scoring rules for ensemble forecasts

Consider the i.i.d. ensemble forecast of the univariate random variable $X \in \mathbb{R}$, with associated distribution π_X and ensemble members $\left\{X^i\right\}_{i=1}^N \sim X$. The empirical cumulative distribution function (CDF) of this forecast is given by the summation

$$F_{\pi_X}^N(x) = \frac{1}{N}\sum_{i=1}^{N} H(x - X^i), \tag{3.1}$$

where H is the Heaviside step function, namely

$$H(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

This empirical estimate approximates the analytical CDF given earlier in Sec. 2.1.2,

$$F_{\pi_X}(x) = \int_{-\infty}^{x} \pi_X(x).$$

One can use this empirical approximation of the forecast CDF to generate scoring rules, that evaluate the effectiveness of the forecast with respect to an observed process, Y, with associated distribution π_Y. In practice one has access to Y by samples / individual measurements $\left\{y_{\text{data}}^j\right\}_{j=1}^{N_{\text{data}}} \sim Y$. A scoring rule is defined below.

Definition 12 (Scoring rule)**.** *A scoring rule is a random variable*

$$S\left(F_{\pi_X}^N, Y\right)$$

representing a scalar score, evaluating different qualities of a forecast, relative to an observed process Y*.*

3.1.1 A probability integral transform histogram

An example of a scoring rule is the probability integral transform. This evaluates the *calibration* of a forecast; calibration is the measure of whether the observations $\left\{y_{\text{data}}^j\right\}_{j=1}^{N_{\text{data}}}$ are statistically indistinguishable from the members of the ensemble forecast $\left\{X^i\right\}_{i=1}^N$ [Carney and Cunningham, 2006]. The probability integral transform is as defined below.

Definition 13 (Probability integral transform)**.** *The probability integral transform of a forecast, with underlying CDF* F_{π_X}*, given an observed process* Y*, is the random variable*

$$P = F_{\pi_X}(Y).$$

A forecast is said to be calibrated if $P \sim U[0,1]$*. This is the case in an identical twin scenario, where the observations are taken from a reference trajectory produced by the same model as the one used to produce the forecast ensemble.*

To compute an estimate of this score for an ensemble forecast, one traditionally calculates a histogram of samples $P_j \in [0,1]$, for $j = 1, ..., N_{\text{data}}$, corresponding to the observations $\left\{y_{\text{data}}^j\right\}_{j=1}^{N_{\text{data}}}$, and tests whether this is a flat, uniform histogram [Gneiting et al., 2007]. This is often referred to as the probability integral transform histogram or rank histogram. In the case that it is uniform, one infers that $P \sim U[0,1]$, and that the forecast is calibrated. If the histogram is convex, it suggests that the ensemble forecast is overdispersed with respect to Y. On the other hand if the histogram is concave, it suggests that the ensemble forecast is underdispersed.

Definition 14 (Probability integral transform histogram)**.** *To construct a probability integral transform (PIT) histogram, start by evaluating the empirical CDF of the ensemble forecast,*

$F^N_{\pi_X}$, with each observation y^j_{data}:

$$P_j = F^N_{\pi_X}(y^j_{data}) = \frac{1}{N}\sum_{i=1}^{N} H(y^j_{data} - X^i). \tag{3.2}$$

A frequency histogram is then computed for $\left\{P_j\right\}_{j=1}^{N_{data}}$.

As an example, consider an ensemble of Brownian motion paths $\left\{W^i_t\right\}_{i=1}^{20}$ at times $t = 1, 2, ..., 100000$. One observation is taken at each time, from the distribution $Y_t \sim N(0, t)$, and used to calculate P_t with the ensemble $\left\{W^i_t\right\}_{i=1}^{20}$. A probability integral transform histogram using the samples P_t, for $t = 1, ..., 100000$, is shown in Figure 9. Taking sampling error into account, this histogram is approximately uniform. This is expected since $P = F_{\pi_{W_t}}(Y_t)$, and the distribution of W_t is $\pi_{W_t} \sim N(0, t)$.

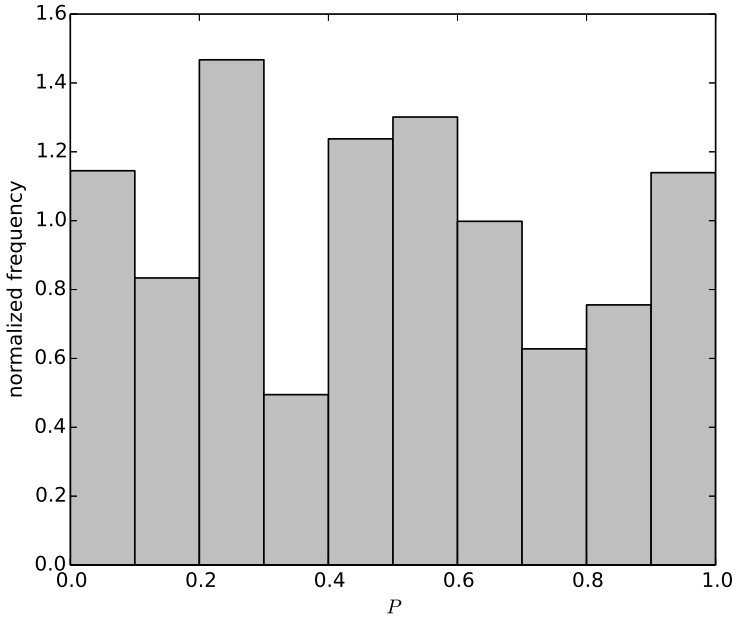


Figure 9: Probability integral transform histogram (with normalized frequency) of a 20-member ensemble forecast for a standard Brownian motion, W_t, with distribution $\pi_{W_t} \sim N(0, t)$. Observations are sampled from $Y_t \sim N(0, t)$, at times $t = 1, 2, 3, ..., 100000$. The histogram is approximately uniform because the ensemble forecast is calibrated with Y_t, i.e. $P = F_{\pi_{W_t}}(Y_t) \sim U$.

3.2 A multilevel Monte Carlo ensemble forecast

Assume $X_t^{h_l}$ is a discretization of the random variable $X_t \in \mathbb{R}$, with associated distribution $\pi_{X_t^{h_l}}$. This discretization uses a time-step h_l, for $l = 0, 1, 2, ...$ with $h_l \propto m^{-l}$ and $m > 1$, as utilised in the multilevel Monte Carlo framework (see Sec. 2.6). Suppose a multilevel Monte Carlo approximation to statistics of $X_t^{h_L}$, for a sufficiently large L, has been computed. The following section proposes a method to generate a single ensemble forecast $\left\{X_t^{F,i}\right\}_{i=1}^{\tilde{N}}$ from the hierarchy ($l = 0, ..., L$) of ensembles,

$$\begin{cases} \left\{X_t^{h_0,i}\right\}_{i=1,...,N_0}, & l = 0 \\ \left\{X_t^{h_l,i}, \hat{X}_t^{h_{l-1},i}\right\}_{i=1,...,N_l}, & l > 0. \end{cases} \tag{3.3}$$

used in the MLMC approximation. Put simply, one can generate a single large ensemble (much larger than the finest ensemble with $l = L$) that represents the entire MLMC approximation to the forecast distribution. This is more useful for the scoring / verification of the hierarchy of ensembles rather than just using standard scoring rules on the finest ensemble in this hierarchy. As aforementioned, the sample sizes for the pairs of ensembles on different levels, N_l, decrease asymptotically. Therefore the finest ensemble is the smallest ensemble in the hierarchy. Using the finest ensemble for the verification of the entire MLMC approximation of the forecast distribution would neglect the majority of samples, on lower levels, from which the approximation was composed.

It is important to note that the ensemble $\left\{X_t^{F,i}\right\}_{i=1}^{\tilde{N}}$ does not contain i.i.d. samples from the finest resolution forecast distribution, $\pi_{X_t^{h_L}}$. Instead they will be approximations to these samples. However, this single ensemble has the properties to form a consistent empirical estimate to the finest resolution forecast distribution and associated distribution functions. This will become clearer later on in the chapter. Assume that values of the MLMC parameters N_l and L have been either set or found, and that the hierarchy of ensembles (3.3) has been generated. Predominantly, this is because the following framework is designed for evaluating any given MLMC approximation. Each approximation has a hierarchy of ensembles that use values of N_l and L that have been optimised around minimising the cost of that particular approximation. This optimisation occurs within an algorithm such as the one in Sec. 2.6.5. Thus, by making the aforementioned assumption, the framework can be kept general to all approximations. In addition to this, it is likely that in real forecasting practice one would

pick the desired finest level L and then set fixed values of N_l based on the maximum computational expense one can use on a particular level. This method of choosing N_l and L is implemented in an example at the end of the chapter.

Inverse transform sampling is used to generate the single ensemble forecast. This technique was previously introduced in Chapter 2. The inverse CDF, F^{-1}, used in inverse transform sampling must typically be empirically estimated. If the true CDF of the forecast distribution is known to be absolutely continuous, as is the case with $X_t \in \mathbb{R}$, and the samples are sorted to form *order statistics*, then some of these estimates have been shown to be consistent approximations to $F^{-1}(u)$ [Ma et al., 2011]. A simple consistent estimate for an evaluation to the quantile function of the distribution π_X, using the (ascending) sorted samples $\left\{X^i\right\}_{i=1}^N \sim X$ is

$$\hat{F}^{-1}_{\pi_X,N}(u) = R(X)^{\lceil N\times u\rceil}, \tag{3.4}$$

where $u \in [0,1]$ and $R(X)^i$ is the i'th order statistic of $\left\{X^i\right\}_{i=1}^N$ such that $R(X)^1 < R(X)^2 < .... < R(X)^N$. Here, the estimate is a consistent one in the sense that it converges in probability to $F^{-1}_{\pi_X}(u)$ as $N \to \infty$. One can also use linear interpolation and extrapolation to smooth this consistent estimate. Other inconsistent techniques include fitting a parametric distribution to the ensemble, such as a Gaussian, and sampling from a closed form quantile function (e.g. Φ for a Gaussian distribution). Inverse transform sampling can be used with these approximations, by sampling $u \sim U[0,1]$ and evaluating them.

The use of inverse transform sampling alongside MLMC was first suggested in Giles [2013]. Here it was proposed to be used to minimise the discrete Wasserstein distance between the two paired ensembles in each difference estimator within (2.20) and thus positively couple them. Instead, the proposed method in this chapter will use inverse transform sampling in the context of a MLMC approximation to the quantile function of the forecast distribution. I.e.

$$\hat{F}^{-1}_{\pi_{X_t^{h_L}},(N_0,...,N_L)}(u) = R(X)_t^{h_0,\lceil N_0\times u\rceil} + \sum_{l=1}^{L}\left(R(X)_t^{h_l,\lceil N_l\times u\rceil} - R(\hat{X})_t^{h_{l-1},\lceil N_l\times u\rceil}\right). \tag{3.5}$$

Note that there is not an exact cancellation in expected values of the terms in (3.5), as was the case in (2.19), because the individual approximations on each level are not unbiased, only consistent in the limit of $N_l \to \infty$. The following algorithm demonstrates how to generate an ensemble $\left\{X_t^{F,i}\right\}_{i=1}^{\tilde{N}}$ of arbitrary size $\tilde{N}$, approximating samples of $X_t^{h_L}$.

(1) For $l = 1, ..., L$ sort $\left\{\hat{X}_t^{h_{l-1},i}\right\}_{i=1}^{N_l}$ and $\left\{X_t^{h_l,i}\right\}_{i=1}^{N_l}$ so that

$$R(\hat{X})_t^{h_{l-1},1} < ... < R(\hat{X})_t^{h_{l-1},N_l},$$

and

$$R(X)_t^{h_l,1} < ... < R(X)_t^{h_l,N_l},$$

in addition to sorting $\left\{X_t^{h_0,i}\right\}_{i=1}^{N_0}$ so that

$$R(X)_t^{h_0,1} < ... < R(X)^{h_0,N_0}.$$

(2) Initialize $X_t^{F,i} = 0$, for $i = 1, ..., \tilde{N}$.

(3) For $l = 0, ..., L$ sample $\left\{u_l^i\right\}_{i=1}^{\tilde{N}} \sim U[0,1]$ and compute

$$X_t^{F,i} += \begin{cases} R(X)^{h_0,\lceil u^i \times N_0 \rceil}, & l = 0 \\ R(X)^{h_l,\lceil u^i \times N_l \rceil} - R(\hat{X})^{h_{l-1},\lceil u^i \times N_l \rceil}, & l > 0. \end{cases}$$

As previously mentioned, the members of the ensemble $\left\{X_t^{F,i}\right\}_{i=1}^{\tilde{N}}$ are not samples from $\pi_{X_t^{h_L}}$, they are only consistent approximations to the evaluations of $F^{-1}_{\pi_{X_t^{h_L}}}(u)$ for a particular u. More specifically, for an uniform random variable $u \sim U[0,1]$, if

$$x = \hat{F}^{-1}_{\pi_{X_t^{h_L}},(N_0,...,N_L)}(u), \tag{3.6}$$

then as $N_l \to \infty$ for all l, it holds that

$$x \xrightarrow{p} F^{-1}_{\pi_{X_t^{h_L}}}(u). \tag{3.7}$$

Then in this limit, x converges in probability to a sample from the forecast distribution on the finest resolution, i.e. $x \sim X_t^{h_L}$. Therefore given the definition of inverse transform sampling in Sec. 2.1.3, the single ensemble $\left\{X_t^{F,i}\right\}_{i=1}^{\tilde{N}}$ can form valid and consistent approximations to statistics of the forecast distribution from the finest resolution, $\pi_{X_t^{h_L}}$. For example, a

consistent approximation to the CDF of $\pi_{X_t^{h_L}}$ can be given by

$$F^{\tilde{N}}_{\pi_{X_t^{h_L}}}(x) = \frac{1}{\tilde{N}} \sum_{i=1}^{\tilde{N}} H(x - X_t^{F,i}). \tag{3.8}$$

This is valid; it is non-decreasing for $X_t^{F,i} \in \mathbb{R}$ and has the support of $[0, 1]$.

One assumes that in practice the computational effort of evaluating the above algorithm a large number of times to generate the ensemble $\left\{X_t^{F,i}\right\}_{i=1}^{\tilde{N}}$ is negligible in comparison to the expense of generating the original samples on all of the different levels of resolution. Thus, the method seems likely to be admissible even when $\tilde{N}$ is much larger than N_0. Having said this, it makes sense here to set $\tilde{N} \propto N_0$ so that both aspects of the approximation (inverse CDF estimator and the ensemble forecast) converge in probability simultaneously with $N_l \to \infty$, for $l = 0, ..., L$ and $\tilde{N} \to \infty$ respectively. Henceforth, assume that $\tilde{N} = \alpha N_0$ with $\alpha \in \mathbb{Z}$ and $\alpha \geq 1$ for simplicity.

The proposed ensemble forecast also preserves the unbiasedness of the approximation to the first moment of $X_t^{h_L}$ from the original MLMC approximation. To show this let $\bar{X}^{h_L}_{\tilde{N}} = \frac{1}{\alpha N_0} \sum_{i=1}^{\alpha N_0} X_t^{F,i}$ be the sample mean of the ensemble forecast from the multilevel hierarchy of ensembles. Then,

$$\begin{aligned}
\bar{X}^{h_L}_{\tilde{N}} &= \frac{1}{\alpha N_0} \sum_{i=1}^{\alpha N_0} X_t^{F,i} \\
&= \left(\frac{1}{\alpha N_0} \sum_{i=1}^{\alpha N_0} \hat{F}^{-1}_{\pi_{X_t^{h_0}, N_0}}(u^i) \right) + \\
&\sum_{l=1}^{L} \left(\left(\frac{1}{\alpha N_0} \sum_{i=1}^{\alpha N_0} \hat{F}^{-1}_{\pi_{X_t^{h_l}, N_l}}(u^i) \right) - \left(\frac{1}{\alpha N_0} \sum_{i=1}^{\alpha N_0} \hat{F}^{-1}_{\pi_{X_t^{h_{l-1}}, N_l}}(u^i) \right) \right), \\
&= \left(\frac{1}{\alpha N_0} \sum_{i=1}^{\alpha N_0} R(X)_t^{h_0, \lceil N_0 \times u^i \rceil} \right) + \\
&\sum_{l=1}^{L} \left(\frac{1}{\alpha N_0} \sum_{i=1}^{\alpha N_0} \left(R(X)_t^{h_l, \lceil N_l \times u^i \rceil} - R(\hat{X})_t^{h_{l-1}, \lceil N_l \times u^i \rceil} \right) \right),
\end{aligned} \tag{3.9}$$

and given that $\{u^i\}_{i=1}^{\alpha N_0} \sim U[0,1]$ are i.i.d. uniform random variables, then

$$\begin{aligned}\mathbb{E}[\bar{X}_{\tilde{N}}^{h_L}] = &\left(\frac{1}{\alpha N_0}\sum_{i=1}^{\alpha N_0}\mathbb{E}[X_t^{h_0}]\right) + \\ &\sum_{l=1}^{L}\left(\frac{1}{\alpha N_0}\sum_{i=1}^{\alpha N_0}\left(\mathbb{E}[X_t^{h_l}] - \mathbb{E}[X_t^{h_{l-1}}]\right)\right) \\ &= \mathbb{E}[X_t^{h_0}] + \sum_{l=1}^{L}\mathbb{E}[X_t^{h_l} - X_t^{h_{l-1}}] = \mathbb{E}[X_t^{h_L}].\end{aligned} \tag{3.10}$$

3.3 Multilevel Monte Carlo scoring rules

Having a method to build the single ensemble forecast representing the multilevel Monte Carlo approximation to $\pi_{X_t^{h_L}}$, leaves us with a simple analogy for a multilevel Monte Carlo scoring rule.

Definition 15. *A multilevel Monte Carlo scoring rule is a random variable*

$$S\left(F^{\tilde{N}}_{\pi_{X_t^{h_L}}}, Y\right)$$

representing a scalar score, evaluating different qualities of a multilevel Monte Carlo forecast, relative to an observed process Y. Here, $F^{\tilde{N}}_{\pi_{X_t^{h_L}}}$ is the empirical CDF using the ensemble forecast $\{X_t^{F,i}\}_{i=1}^{\tilde{N}}$ in (3.8).

3.3.1 A multilevel probability integral transform histogram

Given the definition of a multilevel Monte Carlo scoring rule, one can investigate the calibration of a multilevel Monte Carlo approximation to $\pi_{X_t^{h_L}}$ by using the probability integral transform histogram. This is done in exactly the same way as it was for the standard ensemble forecast earlier on in the chapter, only evaluating the empirical CDF of $\{X_t^{F,i}\}_{i=1}^{\tilde{N}}$ instead. Consider the following example from Sec. 2.4.3. The linear mean reverting OU process, $X_t \in \mathbb{R}$, given by

$$dX_t = \alpha(\mu - X_t)dt + \sigma^2 dW_t, \tag{3.11}$$

over the time interval $t \in [0, T]$, where W_t is a univariate Brownian motion and $T = 40000$, will be used to demonstrate a multilevel Monte Carlo scoring rule. An Euler-Maruyama numerical scheme will be used to discretize X_t with time-step $h_l = 2^{-l-1}$, for $l = 0, ..., L$. In particular, the system with pre-defined parameter sets $S_p = (\alpha, \sigma, \mu)$ for different scenarios of calibration will test PIT histograms from a single ensemble forecast representing a MLMC approximation to $\pi_{X_t^{h_L}}$. Observations are assumed to come from the above model, discretized with time-step $h_4 = 2^{-5}$, with $S_p = (0.1, 0.1, 0)$.

In order to frame this problem in a likely forecasting setting:

(1) First choose a desired fixed finest resolution $L = 4$ and so $l \in [0, 4]$.

(2) Then set a maximum computational expense allowance (the number of floating point operations, in this case) for the propagation of all samples on each level of the ensemble hierarchy, $C_{max} = 1.536 \times 10^7$. The number of floating point operations for each sample in the l'th difference estimator in (2.20) is $\left(Th_l^{-1}(1 + 1/2)\right)$ (as all but the first difference estimator require coarse and fine time-steps of the discretization) and so N_l is given by

$$\begin{aligned} N_l &= \left\lfloor \frac{C_{max}}{T\left(h_l^{-1}\left(1 + 1/2\right)\right)} \right\rfloor \\ &= \left\lfloor \left(\frac{2}{3}\right) C_{max} T^{-1} h_l \right\rfloor. \end{aligned} \tag{3.12}$$

This corresponds to $N_0 = 2^7$.

Pairs of samples from coarse and fine ensembles in each difference estimator in (2.20) are positively coupled by using the same underlying Brownian motion via (2.9), as utilised in Sec. 2.6.7. Observations are collected at times $t_k = k$, $k \in [1, T]$. At each of these times, a single ensemble forecast $\left\{X_{t_k}^{F,i}\right\}_{i=1}^{\tilde{N}}$ is generated from the hierarchy of ensembles that make up the MLMC approximation to the forecast distribution. These are then used to verify the calibration of the approximation via a PIT histogram. In this example, $\tilde{N} = 8N_0 = 2^{10}$ is used. Model parameter sets for four experimental setups are given as follows: $S_p = (0.1, 0.1, 0)$ for the calibrated scenario, $S_p = (0.1, 0.02, 0)$ for the underdispersed scenario, $S_p = (0.1, 0.5, 0)$ for the overdispersed scenario and $S_p = (0.4, 0.1, 0.2)$ for the biased scenario.

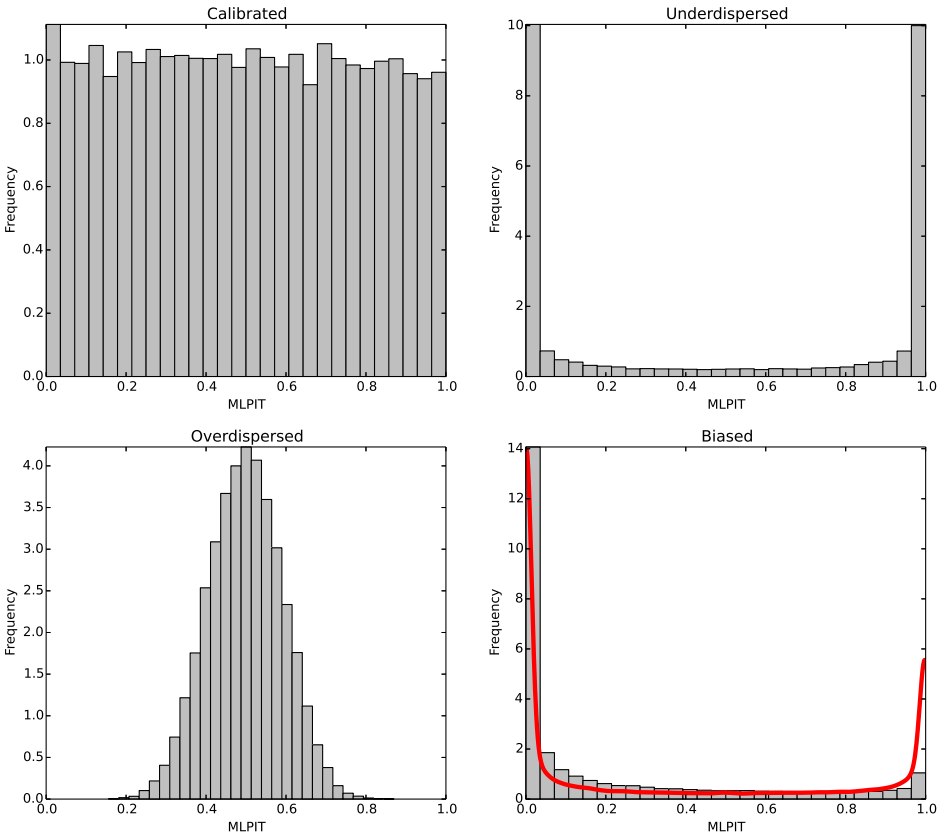


Figure 10: Probability integral transform histograms (normalized frequency), using the multilevel ensemble forecast $\left\{X_{t_k}^{F,i}\right\}_{i=1}^{\tilde{N}}$ of the linear OU process. Each ensemble forecast corresponds to one of four different parameter scenarios: calibrated, overdispersed, underdispersed and biased. The solid line on the biased parameter scenario panel shows a smoothed kernel of the PIT histogram generated from the actual stationary forecast and target distributions.

This setup allows us to establish that the correct calibration behaviour is being shown by the PIT histogram of the multilevel ensemble forecast for each of the S_p scenarios. Figures 10 and 11 show the PIT histograms using the multilevel ensemble forecast and the finest ensemble in (3.3) respectively, for the four scenarios of S_p listed above. Due to the small sample size of the finest ensemble in (3.3), the associated PIT histograms can only represent a very small number of bins of probability. Both sets of PIT histograms show similar behavior for the S_p scenarios above.

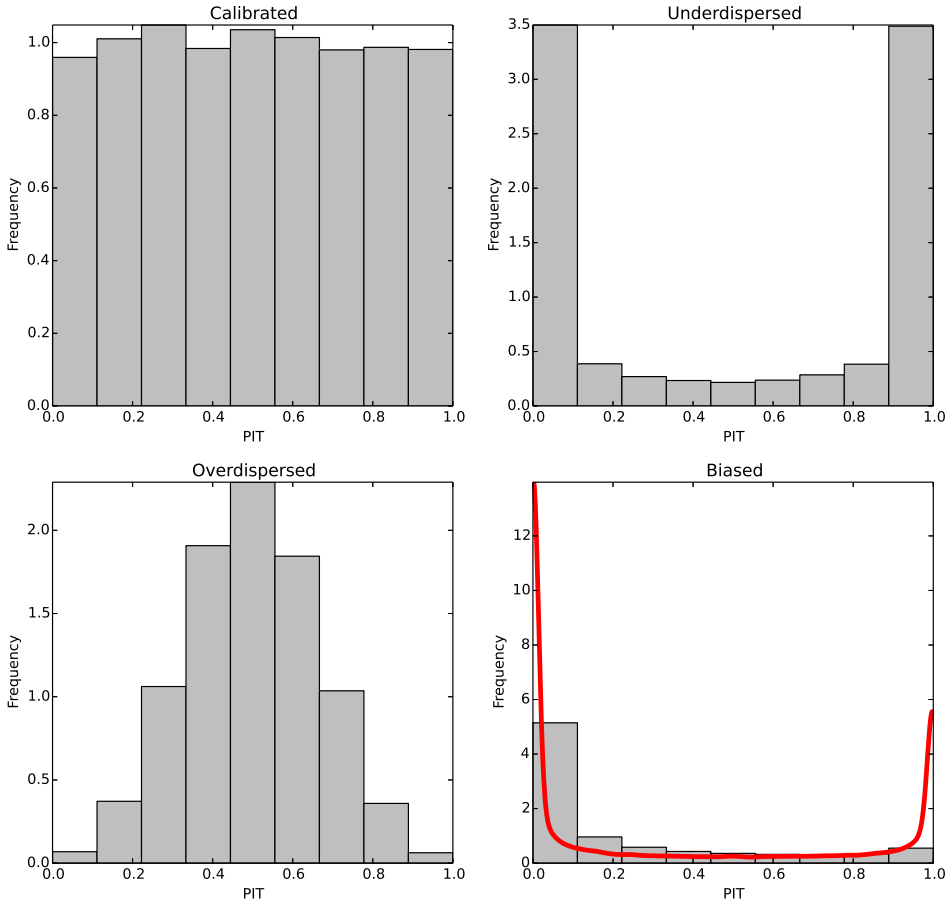


Figure 11: The same as Figure 10 only for the finest ensemble $\left\{X_{t_k}^{h_L,i}\right\}_{i=1}^{N_L}$. The correct type and magnitude of the bias in the biased scenario panel is less clear than in Figure 10.

Recall from Sec. 2.3 that for arbitrary S_p, the stationary distribution of X_t is

$$N\left(\mu, \frac{\sigma^2}{2\alpha}\right).$$

One notes that the stationary forecast distribution using the biased scenario of S_p above is given by $f \sim N\left(0.2, \frac{1}{8}\right)$ and the stationary target distribution is given by $y \sim N\left(0, \frac{1}{2}\right)$. Therefore the actual PIT histogram of $P = F(y)$, where F is the CDF of f, can be generated by taking an arbitrarily large number of samples of $F(y)$. A smoothed density kernel of this histogram is superimposed on the corresponding empirical PIT histograms of the multilevel ensemble forecast and the finest ensemble in (3.3). The empirical histograms approximately

match this, however, due to the lack of samples in the finest ensemble, this particular histogram is not as clear to the type or magnitude of the bias as the one shown in Figure 10. Therefore the calibration and the extent/type of bias in the MLMC approximations to π_{X_t} can be shown with more clarity using the PIT histogram from $\left\{X_{t_k}^{F,i}\right\}_{i=1}^{\tilde{N}}$ than the one from just the finest ensemble in (3.3).

3.4 Conclusions

This chapter proposed a method used to compute scoring rules for multilevel Monte Carlo forecasts, which can estimate statistics via a hierarchy of ensemble forecasts on sequential levels of resolution. Scoring rules allow the verification of ensemble forecasts, away from simple error estimates, relative to a set of observations of a reference trajectory that one wishes to forecast.

One scoring rule in particular is considered in this chapter, the probability integral transform histogram. This allows one to test the calibration of an ensemble forecast; calibration is a measure of whether the observations are statistically indistinguishable from the forecast. Therefore, using the methodology proposed in this chapter, one is able to define what it means for a multilevel Monte Carlo forecast to be calibrated.

Chapter 4

Towards efficient multilevel data assimilation

This chapter is based upon Gregory et al. [2016] and Gregory and Cotter [2017b]

Chapter synopsis

Multilevel Monte Carlo, surveyed in Chapter 2, has recently been posed as an efficient framework for ensemble based data assimilation [Gregory et al., 2016, Gregory and Cotter, 2017b, Jasra et al., 2017a, Hoel et al., 2016]. This is because the filtering estimates discussed in Chapter 2 are a natural extension of the Monte Carlo method. Where the multilevel methods are applied effectively, these so-called 'multilevel filtering estimates' can be significantly more efficient than standard estimates introduced in Chapter 2. This chapter proposes a methodology to apply this theory.

4.1 A multilevel ensemble transform particle filter estimator

A multilevel filtering estimate utilises a hierarchy of models M^{h_l}, for $l = 0, \ldots, L$, denoting discretizations of the prediction model M used in (2.36) (Chapter 2). This was used to transition a forecast random variable X_t between assimilation steps t_{k-1} and t_k, for $k = 1, 2, 3, ...$ with $\Delta t = t_k - t_{k-1}$. Let the associated discretizations of the forecast random variable be $X^{h_l}_{t_k}$. Recall that the parameter h_l represents the resolution of the prediction model M^{h_l} (e.g. the time-step), as in Chapter 2. This parameter is governed by $h_l \propto m^{-l}$ and $m > 1$. It is assumed that the models increase in accuracy as $l \to \infty$, but also become more expensive. Therefore, as evidenced in Chapter 2, the computational cost of evaluating the model M^{h_l} is $\mathcal{O}(h_l^{\gamma})$ with $\gamma > 0$.

Suppose one wishes to approximate statistics of an analysis random variable on the finest level, $\tilde{X}^{h_L}_{t_k}$, with associated distribution $\pi_{\tilde{X}^{h_L}_{t_k}}(x|y_{\text{data},t_1:t_k})$. This represents the posterior of the random variable $X^{h_L}_{t_k}$ given some observations $y_{\text{data},t_1:t_k}$. This can be achieved by the ensemble transform particle filter (ETPF), introduced in Chapter 2. An ETPF estimator for $\mathbb{E}[g(\tilde{X}^{h_L}_{t_k})]$ can be given by

$$\bar{X}^{h_L}_{N,t_k} = \sum_{i=1}^{N} \tilde{w}^{h_L,i}_{t_k} g(X^{h_L,i}_{t_k}), \tag{4.1}$$

where $\left\{X^{h_L,i}_{t_k}\right\}_{i=1}^{N}$ is the forecast ensemble associated with the importance weights $\left\{\tilde{w}^{h_L,i}_{t_k}\right\}_{i=1}^{N}$. This can also be written as

$$\bar{X}^{h_L}_{N,t_k} = \frac{1}{N} \sum_{i=1}^{N} g(\tilde{X}^{h_L,i}_{t_k}), \tag{4.2}$$

where $\left\{\tilde{X}^{h_L,i}_{t_k}\right\}_{i=1}^{N}$ is the analysis ensemble, which has been transformed from the forecast

ensemble. Assume that $g : \mathbb{R}^{N_X} \to \mathbb{R}$ is a scalar function as in Sec. 2.6. Alternatively one can estimate $\mathbb{E}[g(\tilde{X}_{t_k}^{h_L})]$ with a linear combination of ETPF estimators μ_l, with sample sizes N_l, for $l = 0, ..., L$. From Chapter 2, note that

$$\mathbb{E}[g(\tilde{X}_{t_k}^{h_L})] = \mathbb{E}[g(\tilde{X}_{t_k}^{h_0})] + \sum_{l=1}^{L} \mathbb{E}[g(\tilde{X}_{t_k}^{h_l})] - \mathbb{E}[g(\tilde{X}_{t_k}^{h_{l-1}})],$$

and therefore an estimate of $\mathbb{E}[g(\tilde{X}_{t_k}^{h_L})]$ can be written as,

$$\bar{X}_{(N_0,...,N_L),t_k}^{h_L} = \sum_{l=0}^{L} \mu_l, \tag{4.3}$$

where

$$\mu_l = \begin{cases} \sum_{i=1}^{N_0} \tilde{w}_{t_k}^{h_0,i} g(X_{t_k}^{h_0,i}), & l = 0, \\ \sum_{i=1}^{N_l} \tilde{w}_{t_k}^{h_l,i} g(X_{t_k}^{h_l,i}) - \tilde{\tilde{w}}_{t_k}^{h_{l-1},i} g(\hat{X}_{t_k}^{h_{l-1},i}), & l > 0. \end{cases} \tag{4.4}$$

Here, $\left\{X_{t_k}^{h_l,i}\right\}_{i=1}^{N_l} \sim X_{t_k}^{h_l}$ are the forecast ensembles on level l, for $l = 0, ..., L$, with importance weights $\left\{\tilde{w}_{t_k}^{h_l,i}\right\}_{i=1}^{N_l}$ and $\left\{\hat{X}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l} \sim X_{t_k}^{h_{l-1}}$ are the forecast ensembles on level $l-1$, for $l = 1, ..., L$, with importance weights $\left\{\tilde{\tilde{w}}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$. Indeed, the latter set of forecast ensembles are independent of $\left\{X_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_{l-1}}$.

Each forecast ensemble can be transformed to analysis ensembles, as outlined in Chapter 2. The ETPF estimators, μ_l, for $l = 0, ..., L$, can then be written as

$$\mu_l = \begin{cases} \frac{1}{N_0} \sum_{i=1}^{N_0} g(\tilde{X}_{t_k}^{h_0,i}), & l = 0, \\ \frac{1}{N_l} \sum_{i=1}^{N_l} g(\tilde{X}_{t_k}^{h_l,i}) - g(\tilde{\hat{X}}_{t_k}^{h_{l-1},i}), & l > 0. \end{cases} \tag{4.5}$$

Recall from Chapter 2 that the pairs of analysis ensembles in each μ_l, for $l = 1, ..., L$, must be correlated. Between assimilation steps, one can use the same random forcing for each pair of forecast ensembles (if the prediction model M incorporates this, e.g. Brownian paths). Given that the first estimator, μ_0, uses a single independent ensemble $\left\{X_{t_k}^{h_0,i}\right\}_{i=1}^{N_0}$ then this ensemble can be transformed in the standard way for the ETPF framework. However, a challenge is posed to find a transformation strategy, despite the forecast ensembles on different levels being associated with different importance weights, that produces pairs of correlated analysis ensembles $\left\{\tilde{X}_{t_k}^{h_l,i}, \tilde{\hat{X}}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ in each difference estimator, μ_l, for $l = 1, ..., L$. In particular

they must remain correlated so that

$$\mathbb{V}_l = \mathbb{V}\left[g(\tilde{X}_{t_k}^{h_l}) - g(\tilde{X}_{t_k}^{h_{l-1}})\right]$$

decays asymptotically with $h_l \to 0$. This constraint is pivotal for the multilevel estimator in (4.3) to show computational cost reductions (for a fixed order of accuracy) from (4.2). As was the case in Chapter 2, the user is interested in finding an asymptotic rate, $\beta > 0$, such that $\mathbb{V}_l = \mathcal{O}(h_l^\beta)$. In addition to this, recall from Chapter 2 that the greater β is, the greater the computational cost reductions from the multilevel estimator. A methodology to preserve a (albeit low) value of β during the assimilation step in a standard particle filter, with random resampling, was proposed in Jasra et al. [2017a]. The work in Sen et al. [2018] also considered this problem in the general context of positively coupling two particle filters, and investigated using optimal transport for this. The next two sections will address this issue in the context of the ETPF, again drawing on optimal transport.

4.2 Sampling from a coupling between analysis ensembles

4.2.1 Coupling from an assignment problem

This section proposes one possible transformation scheme for the pairs of coarse and fine forecast ensembles $\left\{X_{t_k}^{h_l,i}, \hat{X}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ within each difference estimator μ_l, for $l = 1, ..., L$, whilst preserving positive correlation between the two resulting analysis ensembles. This scheme was proposed in Gregory et al. [2016] and attempts to sample a coupling between both analysis ensembles from a discrete space of couplings. The steps of the scheme are detailed below:

(1) Independently apply an ensemble transform to both the forecast ensembles $\left\{X_{t_k}^{h_l,i}\right\}_{i=1}^{N_l}$ and $\left\{\hat{X}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ with importance weights $\left\{\tilde{w}_{t_k}^{h_l,i}\right\}_{i=1}^{N_l}$ and $\left\{\tilde{\hat{w}}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ following the approach from Sec. 2.7.4. This step results in two independent analysis ensmebles $\left\{\tilde{X}_{t_k}^{h_l,i}\right\}_{i=1}^{N_l}$ and $\left\{\tilde{\hat{X}}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$.

(2) Given both analysis ensembles $\left\{\tilde{X}_{t_k}^{h_l,i}\right\}_{i=1}^{N_l}$ and $\left\{\tilde{\hat{X}}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ associated with equal

weights $1/N_l$ respectively, find the coupling matrix $T^C \in \mathcal{T}^C \in \mathbb{R}^{N_l \times N_l}$ that solves

$$\arg \min_{T^C \in \mathcal{T}^C} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} T^C_{i,j} \left\| \tilde{X}^{h_l,i}_{t_k} - \tilde{\hat{X}}^{h_{l-1},j}_{t_k} \right\|^2, \tag{4.6}$$

where $T^C_{i,j}$ must take non-negative integer values for $i, j = 1, ..., N_l$. In addition to this, it is required to satisfy the constraints

$$\sum_{i=1}^{N_l} T^C_{i,j} = \sum_{j=1}^{N_l} T^C_{i,j} = \frac{1}{N_l}. \tag{4.7}$$

This is called an assignment problem, and can be solved by the Hungarian algorithm [Munkres, 1957], with the same $\mathcal{O}(N^3 \log N)$ complexity as solving the optimal transport problem in multidimensional cases.[1] Solving this results in a matrix T^C with all entries equal to either $1/N_l$ or 0. The analysis ensemble $\left\{\tilde{\hat{X}}^{h_{l-1},i}_{t_k}\right\}_{i=1}^{N_l}$ may then be reordered so that T^C becomes a diagonal matrix.

The filtering estimate in (4.3) with the difference estimators μ_l, for $l = 0, ..., L$, computed by (4.5) and by using the above transformation scheme, is henceforth referred to as the standard multilevel ensemble transform particle filtering estimator [Gregory et al., 2016]. Given that each analysis ensemble in the pair $\left\{\tilde{X}^{h_l,i}_{t_k}, \tilde{\hat{X}}^{h_{l-1},i}_{t_k}\right\}_{i=1}^{N_l}$ is simply rearranged and not modified in Step (2), the actual states of ensembles members will be the same as if they were just transformed in the manner of Sec. 2.7.4. Therefore the equality of (4.4) and (4.5) when g is the identity (mean estimator), is guaranteed via (2.54). The asymptotic consistency of (4.5) with (4.4) when g is arbitrary can also just be shown via (2.58).

4.2.2 A seamless coupling from optimal transport problems

The method proposed in the previous section is inelegant in one sense, as the pairs of forecast ensembles are decorrelated through independent ensemble transformation, before subsequently restoring correlation of the analysis ensembles by using an assignment problem. This assignment problem can be seen as a re-arrangement of the members in both analysis ensembles. Therefore, the coupling T^C of both analysis ensembles is on a discrete space of

[1] The cheaper algorithm described in Reich and Cotter [2015], with complexity $\mathcal{O}(N \log N)$, may be used to solve the assignment problem in one-dimensional cases.

couplings $\mathcal{T}^C$. It is expected that the coupling between the pairs of analysis ensembles will be stronger if the coupling is optimised over a continuous space instead.

A new transformation scheme for both forecast ensembles $\left\{X_{t_k}^{h_l,i}, \hat{X}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ is proposed below to correct this inelegance and optimize over a continuous space of couplings. This is achieved by avoiding the assignment problem and instead seamlessly transforming and coupling the two forecast ensembles simultaneously. To implement this new scheme, the following steps are carried out after updating the importance weights for each pair of forecast ensembles within the difference estimators μ_l, for $l = 1, ..., L$.

(1) Generate a coupling matrix $D \in \mathcal{D} \in \mathbb{R}^{N_l \times N_l}$ that solves

$$\arg\min_{D \in \mathcal{D}} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} D_{i,j} \left\| \hat{X}_{t_k}^{h_{l-1},i} - X_{t_k}^{h_l,j} \right\|^2, \tag{4.8}$$

subject to the marginal constraints

$$\sum_{i=1}^{N_l} D_{i,j} = \tilde{w}_{t_k}^{h_l,j}, \qquad \sum_{j=1}^{N_l} D_{i,j} = \tilde{\hat{w}}_{t_k}^{h_{l-1},i}. \tag{4.9}$$

Define a so-called 'intermediate ensemble' as

$$X_{t_k}^{*,j} = \sum_{i=1}^{N_l} D_{i,j} \tilde{\hat{X}}_{t_k}^{h_{l-1},i} (\tilde{w}_{t_k}^{h_l,j})^{-1}, \tag{4.10}$$

for $j = 1, ..., N_l$, where each member is associated with the importance weight $\tilde{w}_{t_k}^{h_l,j}$. This step transforms the forecast ensemble $\left\{\hat{X}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ with importance weights $\left\{\tilde{\hat{w}}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$, to the intermediate ensemble with the importance weights associated with the forecast ensemble $\left\{X_{t_k}^{h_l,i}\right\}_{i=1}^{N_l}$, i.e. $\left\{\tilde{w}_{t_k}^{h_l,i}\right\}_{i=1}^{N_l}$.

(2) Apply an ensemble transform to the forecast ensemble $\left\{X_{t_k}^{h_l,i}\right\}_{i=1}^{N_l}$ with importance weights $\left\{\tilde{w}_{t_k}^{h_l,i}\right\}_{i=1}^{N_l}$ following the approach within Sec. 2.7.4. This results in the analysis ensemble $\left\{\tilde{X}_{t_k}^{h_l,i}\right\}_{i=1}^{N_l}$.

(3) Find the coupling matrix $T^C \in \mathcal{T}^C \in \mathbb{R}^{N_l \times N_l}$ that solves

$$\arg\min_{T^C \in \mathcal{T}^C} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} T_{i,j}^C \left\| X_{t_k}^{*,i} - \tilde{X}_{t_k}^{h_l,j} \right\|^2, \tag{4.11}$$

subject to the marginal constraints

$$\sum_{i=1}^{N_l} T^C_{i,j} = \frac{1}{N_l}, \qquad \sum_{j=1}^{N_l} T^C_{i,j} = \tilde{w}^{h_l,i}_{t_k}. \tag{4.12}$$

Note that this transformation has the same marginal constraints as the transformation in Step (2), i.e. (2.51) and (2.53), only with importance weights $\{\tilde{w}^{h_l,i}_{t_k}\}_{i=1}^{N_l}$. Importantly, the result is that the space of couplings, $\mathcal{T}^C$, is continuous, unlike in Sec. 4.2.1. Finally the analysis ensemble $\{\tilde{\hat{X}}^{h_{l-1},j}_{t_k}\}_{j=1}^{N_l}$ is given by

$$\tilde{\hat{X}}^{h_{l-1},j}_{t_k} = \sum_{i=1}^{N_l}\sum_{n=1}^{N_l} T^C_{i,j} N_l D_{n,i}\big(\tilde{w}^{h_l,i}_{t_k}\big)^{-1} \hat{X}^{h_{l-1},n}_{t_k}, \tag{4.13}$$

or more simply,

$$\tilde{\hat{X}}^{h_{l-1},j}_{t_k} = \sum_{i=1}^{N_l} T^C_{i,j} N_l X^{*,i}_{t_k}. \tag{4.14}$$

The evenly weighted analysis ensemble $\{\tilde{X}^{h_l,i}_{t_k}\}_{i=1}^{N_l}$ is used in (4.11) to keep itself and the resulting analysis ensemble $\{\tilde{\hat{X}}^{h_{l-1},i}_{t_k}\}_{i=1}^{N_l}$ closely coupled.

The filtering estimate in (4.3) with μ_l, for $l = 0, ..., L$, computed by (4.5) and by using the above transformation scheme, is henceforth referred to as the seamless multilevel ensemble transform particle filtering estimator [Gregory and Cotter, 2017b]. Since the analysis ensemble $\{\tilde{X}^{h_l,i}_{t_k}\}_{i=1}^{N_l}$ is generated from a transform in the manner of Sec. 2.7.4, it can be shown via (2.54) that the mean of it is equal to the weighted mean of the forecast ensemble $\{X^{h_l,i}_{t_k}\}_{i=1}^{N_l}$. However, to fully demonstrate the equality of (4.4) and (4.5) when g is the identity, one must utilise:

$$\begin{aligned}\frac{1}{N_l}\sum_{j=1}^{N_l} \tilde{\hat{X}}^{h_{l-1},j}_{t_k} &= \sum_{j=1}^{N_l}\sum_{i=1}^{N_l}\sum_{n=1}^{N_l} \frac{1}{N_l} T^C_{i,j} N_l D_{n,i}\big(\tilde{w}^{h_l,i}_{t_k}\big)^{-1} \hat{X}^{h_{l-1},n}_{t_k} \\ &= \sum_{i=1}^{N_l}\sum_{n=1}^{N_l} D_{n,i} \hat{X}^{h_{l-1},n}_{t_k} = \sum_{n=1}^{N_l} \tilde{\hat{w}}^{h_{l-1},n}_{t_k} \hat{X}^{h_{l-1},n}_{t_k}.\end{aligned} \tag{4.15}$$

In the same way, the asymptotic consistency of the estimates to higher moments of $\tilde{X}^{h_l}_{t_k}$ is evidenced via (2.58). However, when considering the asymptotic consistency of the estimates to higher moments of $\tilde{X}^{h_{l-1}}_{t_k}$, the transformation of the forecast ensemble $\{\hat{X}^{h_{l-1},i}_{t_k}\}_{i=1}^{N_l}$

should be seen as a combination of two ensemble transforms. By (2.57) [Reich, 2013], the first transform generating the intermediate ensemble satisfies

$$\sum_{i=1}^{N_l} g(X_{t_k}^{*,i})\tilde{w}_{t_k}^{h_l,i} \to \sum_{i=1}^{N_l} g(\hat{X}_{t_k}^{h_{l-1},i})\tilde{\hat{w}}_{t_k}^{h_{l-1},i}, \tag{4.16}$$

as $N_l \to \infty$, and by the same logic

$$\frac{1}{N_l}\sum_{i=1}^{N_l} g(\tilde{\hat{X}}_{t_k}^{h_{l-1},i}) \to \sum_{i=1}^{N_l} g(X_{t_k}^{*,i})\tilde{w}_{t_k}^{h_l,i}, \tag{4.17}$$

also as $N_l \to \infty$. Therefore by combining them and (2.58),

$$\frac{1}{N_l}\sum_{i=1}^{N_l} g(\tilde{\hat{X}}_{t_k}^{h_{l-1},i}) \to \mathbb{E}[g(\tilde{X}_{t_k}^{h_{l-1}})] \tag{4.18}$$

as $N_l \to \infty$. This is demonstrated numerically in a later section.

4.3 Algorithm

The algorithm set out in this section differs from that of the standard MLMC algorithm in Sec. 2.6.5, in that it does not attempt to bound the MSE, or any other error metric, by adaptively fitting the parameters L and N_l, for $l = 0, ..., L$. Instead, for the benefit of simplicity at the assimilation stages, it assumes these parameters are specified already. In practice, and in the numerical demonstrations throughout this chapter, the parameters are designed to give a certain order of accuracy following known values of β and α. In summary, the multilevel ensemble transform particle filter (MLETPF) algorithm is as follows:

(1) For $l = 0, ..., L$, propagate the forecast ensembles

$$\begin{cases} \{X_{t_{k-1}}^{h_0,i}\}_{i=1,...,N_0}, & l = 0, \\ \{X_{t_{k-1}}^{h_l,i}, \hat{X}_{t_{k-1}}^{h_{l-1},i}\}_{i=1,...,N_l}, & l > 0, \end{cases} \tag{4.19}$$

forward to assimilation step t_k using the forward prediction models M^{h_l}.

(2) Compute importance weights for each of the forecast ensemble members using

$$\begin{cases} \tilde{w}_{t_k}^{h_0,i} = \pi_{Y_{t_k}}(y_{\text{data},t_k}|X_{t_k}^{h_0,i}), \quad i = 1,...,N_0, & l = 0, \\ \begin{cases} \tilde{w}_{t_k}^{h_l,i} = \pi_{Y_{t_k}}(y_{\text{data},t_k}|X_{t_k}^{h_l,i}), \\ \tilde{\hat{w}}_{t_k}^{h_{l-1},i} = \pi_{Y_{t_k}}(y_{\text{data},t_k}|\hat{X}_{t_k}^{h_{l-1},i}), \end{cases} & i = 1,...,N_l, \quad l > 0. \end{cases} \tag{4.20}$$

(3) Normalize the weights using

$$\begin{cases} \tilde{w}_{t_k}^{h_0,i} = \tilde{w}_{t_k}^{h_0,i} / \sum_{j=1}^{N_0} \tilde{w}_{t_k}^{h_0,j}, \quad i = 1,...,N_0, & l = 0, \\ \begin{cases} \tilde{w}_{t_k}^{h_l,i} = \tilde{w}_{t_k}^{h_l,i} / \sum_{j=1}^{N_l} \tilde{w}_{t_k}^{h_l,j}, \\ \tilde{\hat{w}}_{t_k}^{h_{l-1},i} = \tilde{\hat{w}}_{t_k}^{h_{l-1},i} / \sum_{j=1}^{N_l} \tilde{\hat{w}}_{t_k}^{h_{l-1},j}, \end{cases} & i = 1,...,N_l, \quad l > 0. \end{cases} \tag{4.21}$$

(4) Transform the weighted forecast ensemble members to the evenly weighted analysis ensembles,

$$\begin{cases} \{\tilde{X}_{t_k}^{h_0,i}\}_{i=1,...,N_0}, & l = 0, \\ \{\tilde{X}_{t_k}^{h_l,i}, \tilde{\hat{X}}_{t_k}^{h_{l-1},i}\}_{i=1,...,N_l}, & l > 0, \end{cases} \tag{4.22}$$

using either one of the schemes in Sec. 4.2.1 or 4.2.2.

(5) Estimate statistics via (4.3) with μ_l, for $l = 0, ..., L$, as in (4.5).

(6) For $l = 0, ..., L$, reset the forecast ensembles

$$\begin{cases} X_{t_k}^{h_0,i} = \tilde{X}_{t_k}^{h_0,i}, \quad i = 1,...,N_0, & l = 0, \\ \begin{cases} X_{t_k}^{h_l,i} = \tilde{X}_{t_k}^{h_l,i}, \\ \hat{X}_{t_k}^{h_{l-1},i} = \tilde{\hat{X}}_{t_k}^{h_{l-1},i}, \end{cases} & i = 1,...,N_l, \quad l > 0, \end{cases} \tag{4.23}$$

and iterate $k = k + 1$.

The computational cost of Step (1) depends on the rate γ introduced in Chapter 2, whilst the cost of Step (4) depends on the optimal transportation cost associated with the transforms in either Sec. 4.2.1 or 4.2.2. Both transform schemes require three optimal transport (or assignment) problems to be solved, and so the order of complexity of the algorithm above is invariant to which scheme is used. The weak convergence of the estimator in (4.3) to

$\mathbb{E}[g(\tilde{X}_{t_k}^{h_L})]$, as $N_l \to \infty$, with $l = 0, ..., L$, for both of the proposed transformation schemes, is evidenced through the each estimator μ_l satisfying

$$\lim_{N_l \to \infty} \mu_l = \begin{cases} \mathbb{E}[g(\tilde{X}_{t_k}^{h_0})], & l = 0 \\ \mathbb{E}[g(\tilde{X}_{t_k}^{h_l})] - \mathbb{E}[g(\tilde{X}_{t_k}^{h_{l-1}})], & l > 0. \end{cases} \tag{4.24}$$

This statement was clarified in the arguments at the end of both Sec. 4.2.1 and 4.2.2. Therefore

$$\bar{X}_{(N_0,...,N_L),t_k}^{h_L} = \sum_{l=0}^{L} \mu_l \to \mathbb{E}[g(\tilde{X}_{t_k}^{h_L})],$$

as $N_l \to \infty$, with $l = 0, ..., L$. It is important to note that an open problem is establishing $\mathbb{L}_2$-convergence rates for this estimator. Indeed, this is a problem inherent of the standard ETPF (and self-normalized sequential Monte Carlo estimators in general) rather than only the multilevel adaptation proposed in this chapter.

4.4 Computational cost considerations

Localisation can be used to reduce the dimensionality of the likelihood computations and cost metrics here (in order to reduce the computational cost of the optimal transport problems) in the same way as done in Sec. 2.7.7. This is done by implementing the algorithm in the previous section for every individual component of $X_{t_k}^{h_l}$, for $l = 0, ..., L$.

When localisation is not utilised, the forward model cost of the standard ETPF (with a fixed order of accuracy) must dominate over the optimal transportation cost of the MLETPF in order for the overall computational cost of the ETPF to be reduced.[2] Even when $\gamma = 1$, if $N_X = 1$ this scenario is likely as the cheap optimal transportation algorithm can be used. However, when $N_X > 1$, optimal transportation becomes significantly more expensive, at least asymptotically. Suppose one desires a MSE of $\mathcal{O}(\epsilon^2)$, then $N_0 = \mathcal{O}(\epsilon^{-2})$ is required for the MLETPF and $N = \mathcal{O}(\epsilon^{-2})$ is required when using the standard ETPF. Also, assume $h_L = \mathcal{O}(\epsilon)$ is required for a first order accurate discretization technique (i.e. where $\alpha = 1$). The result is a forward model cost for the standard ETPF of $C_{FM} = c_m \epsilon^{-2-\gamma}$, as stated in (2.18), where c_m is a constant.

[2] From the arguments in Chapter 2, the MLETPF has reduced forward model cost from that of the standard ETPF.

Assuming that $2N_l = N_{l-1}$ for simplicity, the optimal transportation cost of the MLETPF can then be written as

$$C_{OT} = c_t N_0^3 \sum_{l=0}^{L} (1/2)^{3l} \log\left(N_0 (1/2)^l\right) \leq (8/7) c_t \epsilon^{-6} \log\left(\epsilon^{-2}(1/2)^{1/7}\right), \tag{4.25}$$

using standard (arithmetico) geometric series results [Gregory and Cotter, 2017b]. Here c_t is a constant. In the case where γ is not extremely large, C_{FM} will become less than C_{OT} as $\epsilon \to 0$, and the optimal transportation cost will dominate. As a result, forward model cost reductions from the multilevel framework in the MLETPF would be worthless. However the expectation is that the MLETPF will offer overall cost reductions in a certain ϵ-regime, provided that $c_m/c_t \gg 1$.

It is important to note that the total optimal transport cost in the MLETPF is dominated by the complexity of solving those problems in the coarsest ETPF estimator, where the sample size, N_0, is the greatest. The size of the ϵ-regime, where overall computational cost reductions from the MLETPF will be achieved, is heavily dependent on the number of samples in the coarsest ETPF estimator. Therefore, for an even more efficient implementation and to extend the ϵ-regime, iterative methods such as the Sinkhorn approximation could be considered for the coarser estimators, as described in Sec. 2.7.5. Naturally, significant modifications to the algorithm in Sec. 4.3 (second moment preservation) would need to be applied in this scenario as demonstrated in de Wiljes et al. [2016].

4.5 Numerical demonstrations

4.5.1 Consistency of posterior statistics

During the transformation and coupling scheme in Sec. 4.2.1, both analysis ensembles $\left\{\tilde{X}_{t_k}^{h_l,i}\right\}_{i=1}^{N_l}$ and $\left\{\tilde{\hat{X}}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ are simply rearranged and remain the same as if they were independently transformed (via Sec. 2.7.4). Therefore, (4.24) can be confirmed numerically for this scheme with the evidence in Reich [2013]. The same can be said about the transformation of $\left\{X_{t_k}^{h_l,i}\right\}_{i=1}^{N_l}$ for the scheme in Sec. 4.2.2. However, the forecast ensemble $\left\{\hat{X}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ is transformed to the analysis ensemble $\left\{\tilde{\hat{X}}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N_l}$ via two separate transforms in the second scheme. The numerical confirmation of the asymptotic consistency of this ensemble approximation with $\mathbb{E}[g(\tilde{\hat{X}}_{t_k}^{h_{l-1}})]$ is given below.

Consider the following single Bayesian inference step. Let

$$X_{t_k}^{h_{l-1}} \sim N\big(1,1\big), \qquad X_{t_k}^{h_l} \sim N\big(0.5,1\big).$$

A single observation is sampled from the distribution $N(0.1,2)$. Using Bayes' Theorem, the true posterior distribution, $\pi_{\tilde{X}_{t_k}^{h_{l-1}}}$, given the observation, and the prior distribution $\pi_{X_{t_k}^{h_{l-1}}}$, is $N\big(0.7,2/3\big)$. The ensembles $\big\{\hat{X}_{t_k}^{h_{l-1},i}\big\}_{i=1}^N$ and $\big\{X_{t_k}^{h_l,i}\big\}_{i=1}^N$ of varying size N, are sampled from $X_{t_k}^{h_{l-1}}$ and $X_{t_k}^{h_l}$ respectively, then weighted with respect to the observation. Finally the analysis ensemble $\big\{\tilde{\hat{X}}_{t_k}^{h_{l-1},i}\big\}_{i=1}^N$ is found using the transformation in Sec. 4.2.2. Figure 12 shows the estimated RMSE of the approximations to the mean, variance, third and fourth moments of $\tilde{X}_{t_k}^{h_{l-1}}$ using $\big\{\tilde{\hat{X}}_{t_k}^{h_{l-1},i}\big\}_{i=1}^N$, over $N_{\mathbb{E}} = 10$ independent ensembles.

4.5.2 Stochastic Lorenz 63 equations

Consider the stochastic Lorenz 63 equations for $X_t = (x_t, y_t, z_t)$,

$$\frac{dX_t}{dt} = f(X_t, dW_t) = \begin{cases} (\sigma(y_t - x_t)) + \nu^2 dW_t, \\ (x_t(\rho - z_t) - y_t) + \nu^2 dW_t, \\ (x_t y_t - \beta z_t) + \nu^2 dW_t, \end{cases} \tag{4.26}$$

with $\rho = 28$, $\sigma = 10$, $\beta = 8/3$ and $\nu = 0.1$. The scalar Brownian motion W_t in the system will be the same for each of the three components to maximise the impact of the strong non-linearity in the equations, i.e. to make it a more challenging filtering example. Let X_{t_k} be the solution to the above Lorenz 63 equations at time t_k, and let $X_{t_k}^{h_l}$ be the discretization of X_{t_k} using the Euler-Maruyama numerical scheme with time-step $h_l = m^{-9-l}$, with $m = 2$. This coarsest time-step of $h_0 = 2^{-9}$ is used for stability reasons. Here, $\Delta t = 2^2 h_0$ and take $k \in [1, 1280]$. Observations taken from a reference trajectory of X_{t_k}, are given by a measurement error with $R = 0.25I$, where I is the 3×3 identity matrix. No localisation is used for the importance weighting and the cost metrics in the forecast to analysis transformation, (see Sec. 2.7.7 and 2.7.7 respectively). Between each assimilation step, the same Brownian path is used for each pair of samples in the difference estimators of the MLETPF approximations, μ_l, for $l = 1, ..., L$, via (2.9).

First, this example will be used to demonstrate the effects of the coupling / transformation

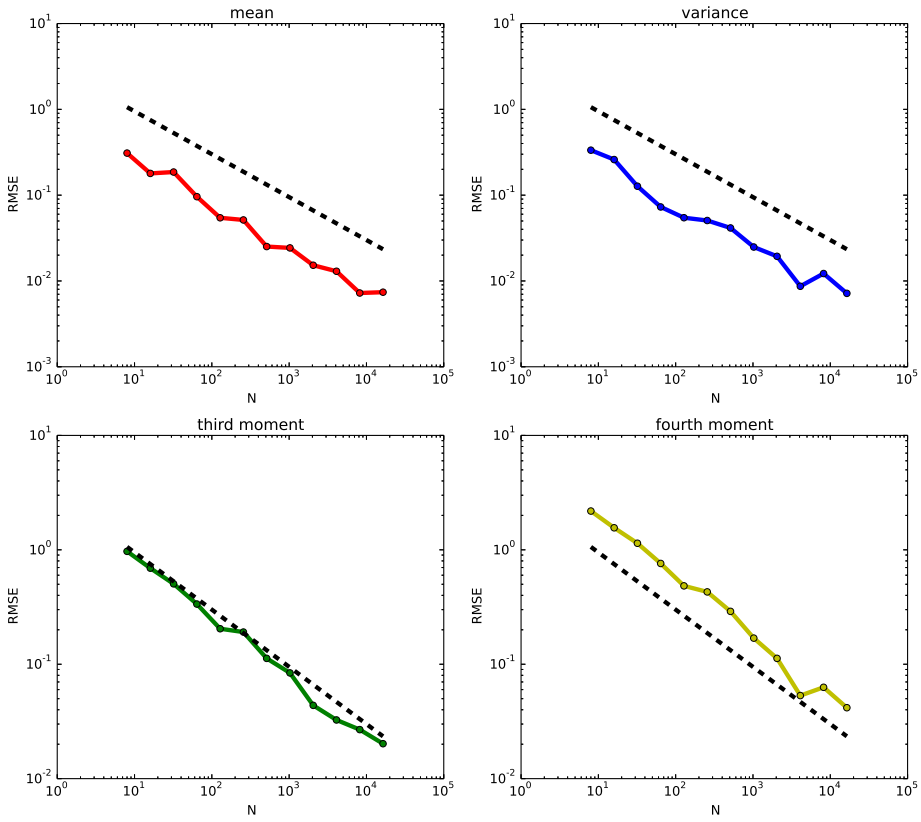


Figure 12: Root mean square errors of the approximations to the mean, variance, third and fourth moments of $\tilde{X}_{t_k}^{h_{l-1}}$ using the analysis ensemble from the seamless coupling scheme, $\left\{\tilde{\hat{X}}_{t_k}^{h_{l-1},i}\right\}_{i=1}^{N}$. Following the asymptotes shown by the black dashed lines, these errors decay at a rate of $\mathcal{O}(N^{1/2})$ showing convergence of the analysis statistical approximations.

schemes in Sec. 4.2.1 and 4.2.2 on the rate of asymptotic decay (β) of

$$\mathbb{V}_l = \begin{cases} Tr\left(\mathrm{Cov}[\tilde{X}_{t_k}^{h_0}]\right), & l = 0 \\ Tr\left(\mathrm{Cov}[\tilde{X}_{t_k}^{h_l} - \tilde{X}_{t_k}^{h_{l-1}}]\right), & l > 0 \end{cases} \tag{4.27}$$

with increasing l. The MLETPF estimator alongside a pre-defined set of sample sizes, $N_l = 2^{8-l}$, for $l \in [0, 6]$, is used to compute an approximation to $\mathbb{E}[\tilde{X}_{t_k}]$, where $\tilde{X}_{t_k}$ is the random variable associated with the posterior $\pi_{\tilde{X}_{t_k}}$, given all of the observations up until time t_k. Figure 13 shows the average rates of asymptotic decay in $\mathbb{V}_l$, using 5 independent implementations of the standard and seamless MLETPF estimators. The seamless estima-

tor produces a variance decay close to $\beta = 2$, whilst $\beta \approx 1$ is obtained for the standard MLETPF estimator. In this case, the coupling produced by the scheme in Sec. 4.2.2 appears to be stronger than the one in Sec. 4.2.1, causing the greater rate of decay in $\mathbb{V}_l$.

Remark 1. *Given the additivity of the noise in this system, and indeed the system in the next section, one can expect an $\mathcal{O}(h_l)$ strong rate of convergence, even for the Euler-Maruyama method. This suggests an optimal value of $\beta = 2$ via arguments in Sec. 2.6.6. The coupling produced by the scheme in Sec. 4.2.2 appears to achieve close to this.*

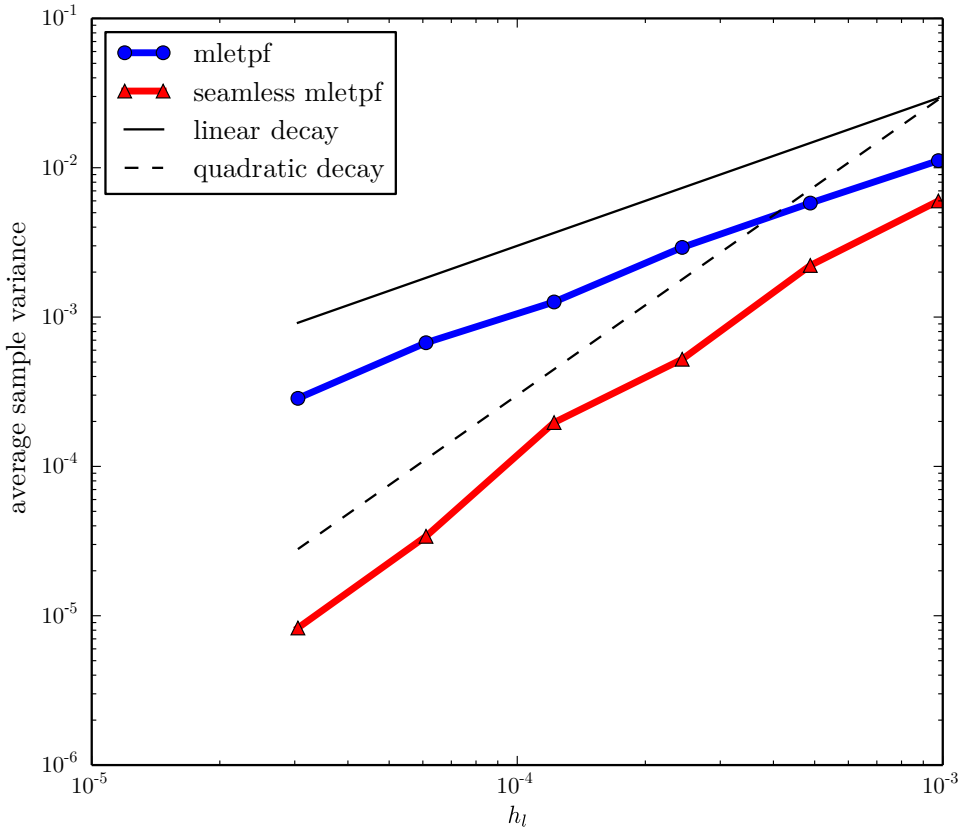


Figure 13: Average value of $\mathbb{V}_l$, for $l \in [1, 6]$, over 5 independent simulations and all assimilation steps $k \in [1, 1280]$ for the stochastic Lorenz 63 equations filtering example. Following the asymptotes shown by the solid and dashed black lines, the quantity $\mathbb{V}_l$ decays at rates of $\mathcal{O}(h_l)$ and $\mathcal{O}(h_l^2)$ for the standard MLETPF and seamless MLETPF implementations respectively.

Second, this example shows the effect that these two values of β, shown in Figure 13, have on the overall forward model cost of the MLETPF estimator and also the forward model

cost reductions from that of the standard ETPF. Since the cost of solving multivariate optimal transport problems dominates over the cheap forward model in this example (associated with $\gamma = 1$), one does not expect the MLETPF to offer overall speed-ups in convergence. However, the hope is that the greater rate of variance decay in the seamless MLETPF estimator will produce greater reductions in the forward model costs than that of the standard MLETPF from the ETPF estimator. This suggests that there would be an overall runtime benefit whenever the forward model cost of the standard ETPF dominates that of solving the multivariate optimal transport problems (see Sec. 4.4).

Define a desired order of time-averaged RMSE, $\mathcal{O}(\epsilon)$, and this can be used to determine the parameters L and N_l. The scaling $L = \mathcal{O}(-log(\epsilon)/log(2))$ is used for both the standard and seamless MLETPF estimators. The scaling $N_l = \mathcal{O}(\epsilon^{-2}2^{-l})$ is used for the standard MLETPF and $N_l = \mathcal{O}(\epsilon^{-2}2^{-(3/2)l})$ is used for the seamless MLETPF. The geometric decay in N_l is used primarily for simplicity and a proof of concept, these are not necessarily optimal, nor are they set adaptively to bound the error exactly as done in Sec. 2.6.7. Instead, the different rates of geometric decay in N_l are designed to exploit the different values of β shown in Figure 13, in order to achieve particular reductions in forward model cost for the MLETPF estimators of a fixed order of estimator variance $\left(\sum_{l=0}^{L} \mathbb{V}_l/N_l\right)$, and thus accuracy. For the standard ETPF, $N = \mathcal{O}(\epsilon^{-2})$ and L is set to be the same as in the MLETPF estimators. The growth rates of forward model cost for the standard ETPF, in addition to both the standard and seamless MLETPF estimators are shown in Figure 14 for decreasing values of ϵ. The rates of $\mathcal{O}\left(\epsilon^{-2}\right)$ for the seamless MLETPF and $\mathcal{O}\left(\epsilon^{-2}\log(\epsilon)^2\right)$ for the standard MLETPF are observed, and are consistent with the analysis in (2.27) for when $\beta > \gamma$ and $\beta = \gamma$ respectively. These are both reductions from the rate of $\mathcal{O}\left(\epsilon^{-3}\right)$ associated with the standard ETPF. Here computational cost is measured by the total number of floating point operations for the forward model computation.

4.5.3 Stochastic Lorenz 96 equations

A slightly different system of arbitrary dimension is now considered: the stochastic Lorenz 96 equations. It is given by the following spatial discretization,

$$\frac{dX(j)}{dt} = -\frac{\big(X(j-1)X(j+1) - X(j-2)X(j-1)\big)}{3\Delta} - X(j) + F + \sigma^2\frac{dW_j}{dt}. \quad (4.28)$$

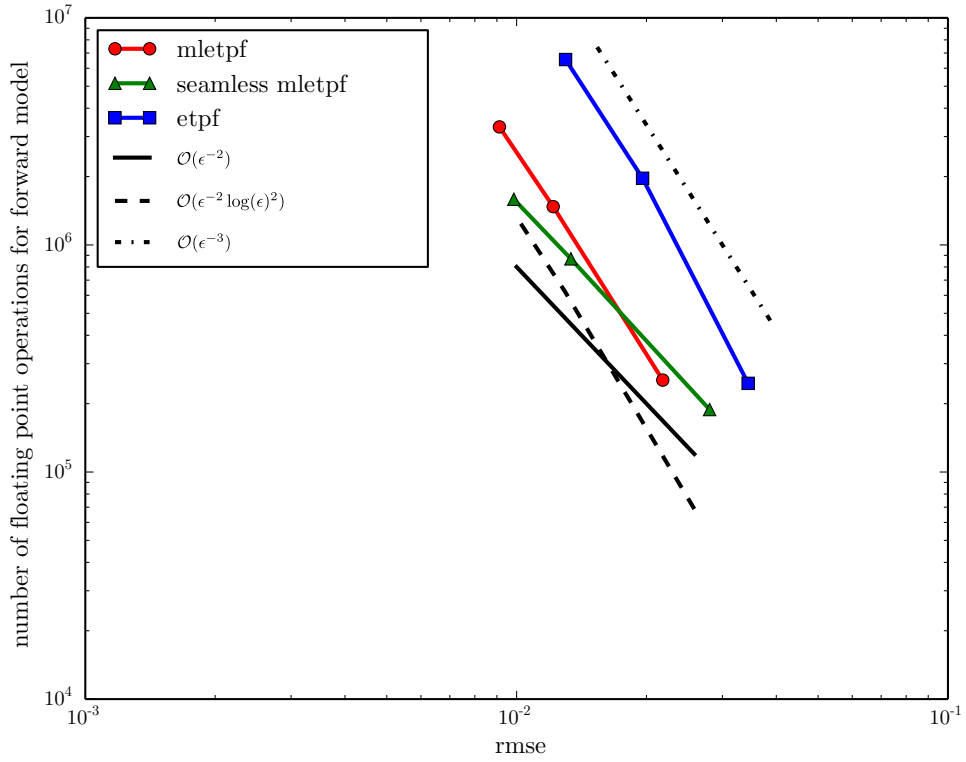


Figure 14: Forward model costs against time-averaged RMSE of the standard ETPF, the standard MLETPF and seamless MLETPF estimators for decreasing ϵ values using the stochastic Lorenz 63 equations filtering example. Following the asymptotes shown by the solid, dashed and dotted black lines, the forward model costs scale as $\mathcal{O}(\epsilon^{-2})$, $\mathcal{O}(\epsilon^{-2}\log(\epsilon)^2)$ and $\mathcal{O}(\epsilon^{-3})$ for the seamless MLETPF, the standard MLETPF and the standard ETPF respectively.

Here, F is a forcing term, typically taken to be $F = 8$ for chaotic behavior, $j \in [1, 40]$, $\Delta = 0.5$, and W_j are independent standard Brownian Motions, with $\sigma^2 = 0.1$. This independence in the stochasticity between components decreases the correlations between them. Therefore localisation, which will be used for this problem, has little effect on the performance of the assimilation. In systems with strong spatial dependence structures, localisation will have more of an effect on the ETPF and the multilevel versions proposed in this chapter [Cheng and Reich, 2013]. The choice of some parameters, such as the frequency of assimilation, is another important aspect to consider when choosing if localisation is suitable for a particular problem. For example, if assimilation is more frequent, spatial dependence will be altered more frequently in the presence of localisation, which in turn could lead to an unstable filter.

Let $X_{t_k}^{h_l}(j)$ represent the j'th component of X at time t_k (with time-index k), discretized

by the Euler-Maruyama numerical scheme with time-step $h_l = m^{-8-l}$, where $m = 2$. The coarsest time-step, $h_0 = 2^{-8}$, is again used for stability purposes. Here, $\Delta t = h_0$, and take $k \in [1, 1280]$. Observations are given at these assimilation steps by using a measurement error variance of $R = 0.25I$, where I is the 40×40 identity matrix. Again, between each assimilation step, the same Brownian path is used for each pair of samples in each difference estimator of the MLETPF approximations, μ_l, for $l = 1, ..., L$, via (2.9).

This system will be used to investigate the overall computational cost reductions of the standard and seamless MLETPF estimators, from that of the standard ETPF, with a pre-defined order of accuracy. Due to the use of localisation for this example (with the settings $r_{loc,c} = r_{loc,R} = 0$), the forward model cost is now expected to dominate over the cost of solving the optimal transport problem in the standard ETPF. It is therefore expected that the MLETPF estimators will return overall computational cost reductions. As stated in Chapter 2, the use of localisation does lead to an inconsistent approximation to statistics of $\tilde{X}_{t_k}^{h_L}$, where $\tilde{X}_{t_k}^{h_L}$ is the random variable associated with the posterior $\pi_{\tilde{X}_{t_k}^{h_L}}$, given all of the observations up until time t_k. Therefore, the time-averaged RMSE is estimated relative to a localised (with the same settings), high accuracy ETPF approximation that the localised ETPF and MLETPF estimates are asymptotically consistent with. The time-step of the discretization used in this approximation is h_{13} and the sample size is $N = 40000$.

It is useful to assume here that one has already chosen to use the localised ETPF for this problem and are using the multilevel framework to improve the efficiency of this. Hence evaluating the convergence of the MLETPF to a localised ETPF approximation rather than the unlocalised truth is sensible in this setting. If this convergence was computed with respect to this truth, the speed-up in convergence from the application of the multilevel framework would plateau at some accuracy due to the localisation bias. This bias is a problem inherent of localisation in general and not specifically the multilevel framework.

As done in the last example, pre-defined values of L and N_l will be used in the standard and seamless MLETPF estimators. The scalings $L = \mathcal{O}(-\log(\epsilon)/\log(2))$ and $N_l = \mathcal{O}(\epsilon^{-2}2^{-(3/2)l})$ are used in both the standard and seamless MLETPF estimators. These sample sizes are chosen in respect of Figure 15. Here the rates of decay (with increasing l) in the average value of $\mathbb{V}_l$ in (4.27) (over all assimilation steps) for the MLETPF estimators are shown. Unlike the example in Sec. 4.5.2, the standard MLETPF estimator achieves the same quadratic rate of decay in $\mathbb{V}_l$ (as the seamless MLETPF) now that localisation (with

$r_{loc,c} = r_{loc,R} = 0$) is used. However the magnitude of variance is still smaller for the seamless MLETPF.

For the standard ETPF estimator the sample sizes are set to scale as $N = \mathcal{O}(\epsilon^{-2})$ and L scales as it did in the last example. The time-averaged RMSEs against overall computational costs for the standard and seamless implementations of the MLETPF estimator to $\mathbb{E}[\tilde{X}_{t_k}]$, in addition to the standard ETPF estimator approximating $\mathbb{E}[\tilde{X}_{t_k}]$, are displayed for various ϵ values in Figure 16. Noting that $\gamma = 1$, the forward model cost and thus the overall computational cost of the standard ETPF estimator follows a $\mathcal{O}(\epsilon^{-3})$ scaling, due to (2.18). Computational cost reductions, down to approximately a $\mathcal{O}(\epsilon^{-2})$ scaling, are present in both the standard and seamless MLETPF estimators. This scaling agrees with (2.27). The lower magnitude of time-averaged RMSEs for the seamless MLETPF estimator are a consequence of the lower variances in Figure 15. Similar rates are shown for the three estimators approximating $\mathbb{E}[(\tilde{X}_{t_k})^2]$. For both sets of estimators, the CAR values, introduced in Chapter 2, for the standard ETPF, standard MLETPF and seamless MLETPF therefore scale as $\mathcal{O}(\epsilon^{-1})$, $\mathcal{O}(1)$ and $\mathcal{O}(1)$ respectively. These match the theoretical rates in (2.28) and (2.29).

4.6 Conclusions

This chapter has presented an application of multilevel Monte Carlo (MLMC), to an ensemble transform method for Bayesian inference, specifically the ensemble transform particle filter (ETPF), to form the multilevel ETPF estimator. This has been rationalized to significantly speed up the propagation of ensembles used within high discretization accuracy estimators to posterior statistics. The application of MLMC requires coupling schemes, in order to maintain positive correlation between pairs of coarse and fine resolution analysis ensembles during the assimilation of observations. Two schemes have been proposed, the second being a seamless combination of transformation and coupling of the weighted coarse and fine resolution forecast ensembles. The stronger the coupling, the less variance the multilevel estimator has, and therefore the more efficient it is. The multilevel estimator, alongside either coupling scheme, achieves speed-ups in forward model complexity for increasing accuracy, over the standard ETPF, in numerical examples.

However, the computational cost of the ETPF can be dominated by large optimal transportation costs for multivariate problems, if the forward model is not expensive to evaluate. In this case the application of MLMC, to reduce the ensemble propagation cost, is negated.

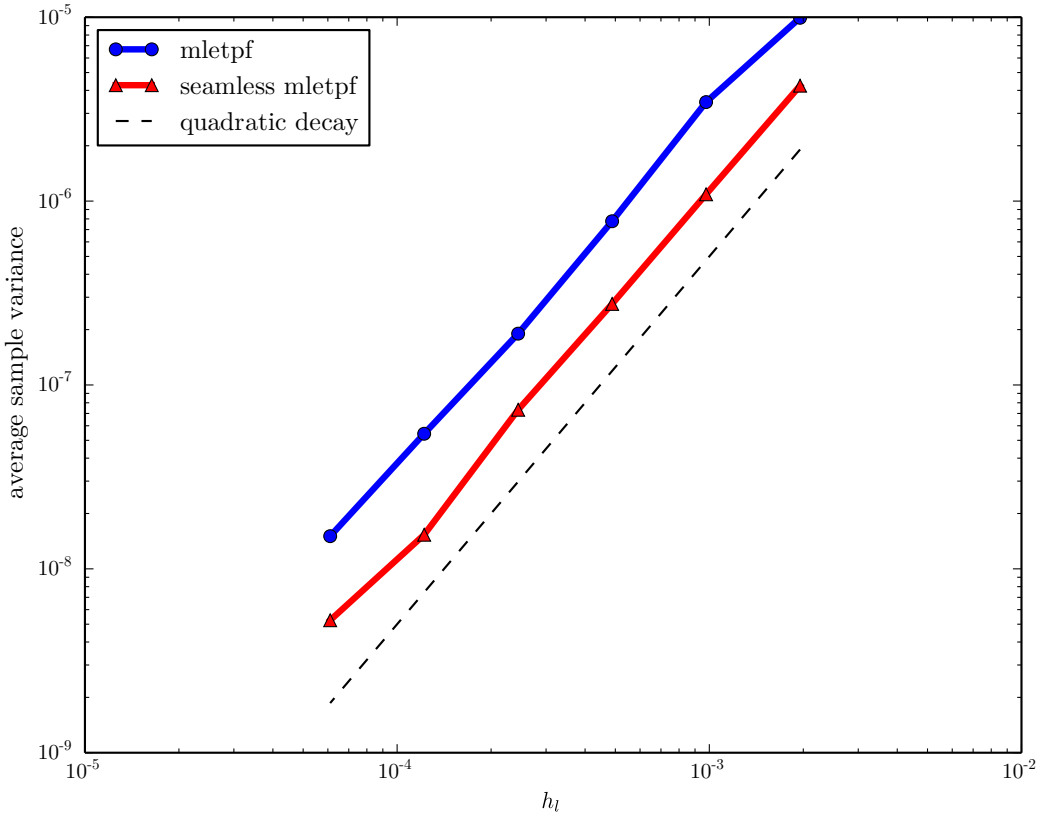


Figure 15: Average values of $\mathbb{V}_l$, for $l \in [1, 6]$, over all assimilation steps $k \in [1, 1280]$ for the stochastic Lorenz 96 equations filtering example. Following the asymptotes shown by the dashed black line, the quantity $\mathbb{V}_l$ decays at a rate of $\mathcal{O}(h_l^2)$ for both the standard and seamless implementations of the MLETPF. However the magnitude of $\mathbb{V}_l$, for $l \in [1, 6]$, is less for the seamless implementation of the MLETPF than it is for the standard one.

Localisation (introduced in Chapter 2) can be used to lower the complexity of the assimilation steps containing the optimal transport problems. When localisation is implemented, numerical examples show that even for a cheap temporal forward model (e.g. $\gamma = 1$), the application of MLMC can achieve overall speed-ups with respect to the ETPF.

In some cases, the forward model will be expensive to evaluate, especially in spatio-temporal models, and dominate over the (multivariate) optimal transportation costs of the ETPF for lower sample sizes / orders of accuracy. In this case, applying MLMC to the ETPF will offer overall runtime benefits. The next two chapters consider this case further.

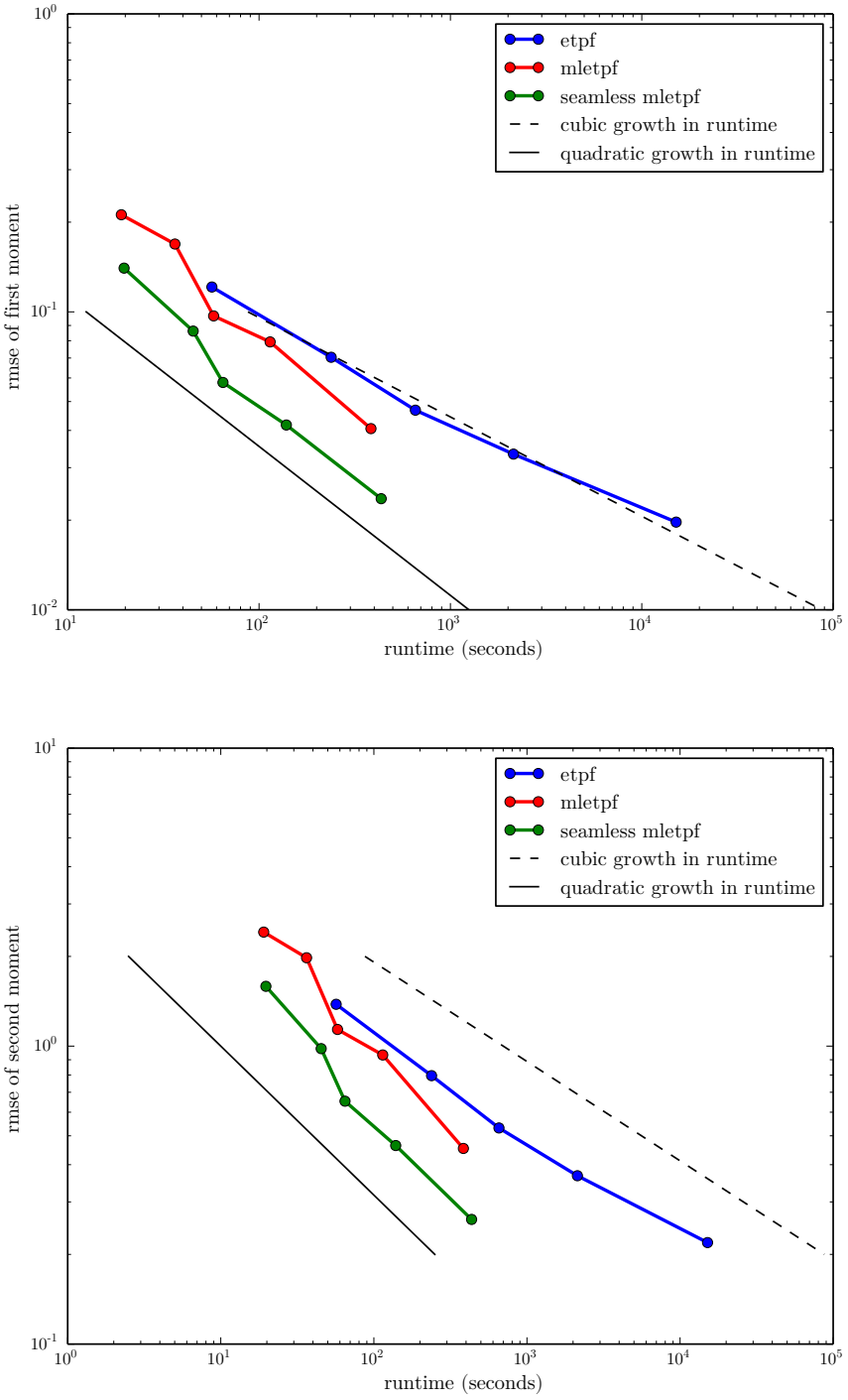


Figure 16: Time-averaged RMSE against runtime (seconds) of the standard ETPF, the standard MLETPF and seamless MLETPF estimators to $\mathbb{E}[\tilde{X}_{t_k}]$ (top panel) and $\mathbb{E}[(\tilde{X}_{t_k})^2]$ (bottom panel) using the stochastic Lorenz 96 equations filtering example. Results are shown for decreasing values of ϵ. Following the asymptotes shown by the solid and dashed black lines, the runtimes scale as $\mathcal{O}(\epsilon^{-2})$ and $\mathcal{O}(\epsilon^{-3})$ for both implementations of the MLETPF and the standard ETPF respectively.

Chapter 5

Spatio-temporal ensemble transform methods

Chapter synopsis

This chapter uses the ensemble transform particle filter (ETPF), introduced in Chapter 2, for data assimilation concerning systems of partial differential equations. This extends the spatial considerations of the ETPF in Sec. 2.7.7 to practical problems in geosciences. To do this, the ETPF is modified to fit random functions of both time and space, $\phi_t(x, \omega)$, with $\omega \in \Omega$, $t \geq 0$ and $x \in \Gamma$. These functions are discretized using finite element methods. It is therefore an extension to the seminal localisation strategy for the ETPF in Cheng and Reich [2013], in that it is designed specifically for finite element approximations to random functions.

5.1 Preliminaries

The function $\phi_t(x, \omega)$, existing on the space V, is assumed to require discretization, as it evolves over time and space according to a random PDE,

$$f\left(t, \phi, \frac{\partial \phi}{\partial t}, ..., \nabla\phi, \nabla\nabla\phi, ...; \omega\right) = 0. \tag{5.1}$$

This PDE can define the propagation of the function $\phi_t(x)$, from time t to time $s > t$, using the forward prediction model in (2.36), i.e.

$$\phi_s(x) = M(t, \phi_t(x), \omega).$$

Let the discretization be denoted by $\phi_t^{(h,\delta x)}(x)$, where the discretization parameter, that has been considered throughout this thesis, is now a vector of temporal (h) and spatial (δx) discretization resolutions: $(h, \delta x)$. Assume this approximation is on the space $V^{\delta x}$.

5.2 Finite element approximations

A finite element approximation of a function, $\phi_t^{(h,\delta x)}(x)$, can be written as

$$\phi_t^{(h,\delta x)}(x) = v^{\delta x}(x) \cdot a^{\delta x},$$

where $v^{\delta x}(x)$ is the vector of nodal basis functions associated with the vector of basis coefficients, $a^{\delta x}$. Denote the mesh by $\mathcal{M}^{\delta x}$. Here, the mesh has cells $c_i^{\delta x}$ that each contain nodes. The basis coefficients represent the value of the function at these nodes. In the next two chapters, motivated by the form of $\phi_t^{(h,\delta x)}(x)$ above, the multivariate random variable $X_t^{(h,\delta x)}(\omega) : \Omega \to \mathbb{R}^{N_X^{\delta x}}$ is considered in the context of data assimilation. This is a vector of all the basis coefficients of $\phi_t^{(h,\delta x)}(x)$, namely

$$X_t^{(h,\delta x)} = \left(a_1^{\delta x}, ..., a_{N_X^{\delta x}}^{\delta x}\right).$$

Note that the number of basis coefficients of $\phi_t^{(h,\delta x)}(x)$, and therefore the dimension of $X_t^{(h,\delta x)}$ is now variable on the spatial resolution δx. Let δx be the resolution of $\mathcal{M}^{\delta x}$, e.g. each cell width/area. Now, assume that in the manner of Chapter 2, the spatial discretization parameter δx is part of a hierarchy governed by $\delta x_l \propto m^{-l}$, for $l = 0, 1, 2, ...$, with $m > 1$, where the associated meshes, $\mathcal{M}^{\delta x_l}$, are nested such that

$$\cdots \subset \mathcal{M}^{\delta x_l} \subset \mathcal{M}^{\delta x_{l-1}} \cdots \subset \mathcal{M}^{\delta x_0}. \tag{5.2}$$

The function spaces existing on these meshes are also part of a hierarchy,

$$\cdots \subset V^{\delta x_l} \subset V^{\delta x_{l-1}} \cdots \subset V^{\delta x_0}. \tag{5.3}$$

The mesh $\mathcal{M}^{\delta x_0}$ is the coarsest mesh, with subsequent meshes $\mathcal{M}^{\delta x_l}$, with $l > 0$, being finer meshes. Due to the meshes existing within a nested hierarchy, it means that a set of cells in a finer mesh $\left\{c_i^{\delta x_l}\right\}_{i=1}^n$ are sub-cells of a cell $c^{\delta x_{l-1}}$ in a coarse mesh. I.e.

$$\bigcup_{i=1,...,n} c_i^{\delta x_l} = c^{\delta x_{l-1}}. \tag{5.4}$$

For the remainder of this chapter, the temporal resolution of $\phi_t^{(h,\delta x_l)}(x)$ is left fixed, unlike in MLMC, see Chapters 2 and 4. The sole reason for the spatial resolution being part of a hierarchy, is for use within a localisation scheme described later in this chapter. This scheme first requires the introduction of projection.

5.3 Projection

In projection one is interested in transferring the functional approximation $\phi_t^{(h,\delta x_l)}(x) \in V^{\delta x_l}$ to another function space $U^{\delta x_l}$. In the case of $\mathbb{L}_2$-projections, this is done by finding the function approximation $f(x) \in U^{\delta x_l}$ that satisfies

$$\int f(x)udx = \int \phi_t^{(h,\delta x_l)}(x)udx, \quad \forall u \in U^{\delta x_l}.$$

Let the operator $\mathbb{I}_{V^{\delta x_l}}^{U^{\delta x_l}}$ represent the $\mathbb{L}_2$-projection of $\phi_t^{(h,\delta x_l)}(x)$ onto the function space $U^{\delta x_l}$. There are two special cases of projections, for when $U^{\delta x_l}$ is part of the hierarchy in (5.3).

5.3.1 Prolongation

Here, one would like to project $\phi_t^{(h,\delta x_l)}(x)$ on to the function space $V^{\delta x_q}$, where $q > l$. For brevity, let the operator $\mathbb{I}_l^q$ represent the prolongation of $\phi_t^{(h,\delta x_l)}(x)$ onto the function space $V^{\delta x_q}$. This is the equivalent to interpolation of the nodal basis coefficients.

5.3.2 Injection

Here, one would like to project $\phi_t^{(h,\delta x_l)}(x)$ on to the function space $V^{\delta x_q}$, where $q < l$. For brevity, let the operator $\mathbb{I}_l^q$ represent the injection of $\phi_t^{(h,\delta x_l)}(x)$ onto the function space $V^{\delta x_q}$. This can be done by $\mathbb{L}_2$-projection or mass lumping for efficiency [Duan et al., 2006].

5.4 Filtering framework

This section will consider a forecast ensemble $\left\{X_t^{(h,\delta x_l),i}\right\}_{i=1}^N \sim X_t^{(h,\delta x_l)}$ in ETPF approximations to the statistics of $X_{t_k}^{(h,\delta x_l)}$ conditioned on observations, $y_{\text{data},t_1:t_k}$. Here, as in Chapter 2, the time indices $k = 1, 2, 3, ...$ define the assimilation steps, and $\Delta t = t_{k+1} - t_k$. The actual structure of the observations, $y_{\text{data},t_1:t_k}$, will be studied in the next section. Let $\tilde{X}_{t_k}^{(h,\delta x_l)}$ represent the analysis random variable, associated with the posterior distribution $\pi_{X_{t_k}^{(h,\delta x_l)}|Y_{t_1:t_k}}$, given the data $y_{\text{data},t_1:t_k}$. Let $\pi_{Y_{t_1:t_k}}$ be the distribution associated with the underlying random variables of the observations $y_{\text{data},t_1:t_k}$ and $\pi_{X_{t_k}^{(h,\delta x_l)}}$ be the distribution associated with the random variable $X_{t_k}^{(h,\delta x_l)}$. Ensemble transform particle filter approximations of $\mathbb{E}[\tilde{X}_{t_k}^{(h,\delta x_l)}]$ for example, provide vectors of basis coefficients for an approximation to $\mathbb{E}[\tilde{\phi}_{t_k}^{(h,\delta x_l)}]$, where

$\tilde{\phi}_{t_k}^{(h,\delta x_l)}$ is a so-called analysis function. These statistical estimates are the aim of the following filtering framework.

Let $\left\{\phi_t^{(h,\delta x_l),i}(x)\right\}_{i=1}^N$ be the discretizations of $\phi_t(x)$ with the basis coefficients making up the random variables $\left\{X_t^{(h,\delta x_l),i}\right\}_{i=1}^N$. The discretized forward prediction model, $M^{(h,\delta x_l)}$, will be used to propagate the ensemble $\left\{\phi_t^{(h,\delta x_l),i}(x)\right\}_{i=1}^N$ from one assimilation step to another.

5.4.1 Observations

The way observations are incorporated into this framework is different than previously done in Chapter 2. Using a reference trajectory, $\phi_t^{(h,\delta x_l),\text{ref}}(x)$, the observation vector $y_{\text{data},t_k} = (y_{\text{data},t_k}^1, ..., y_{\text{data},t_k}^{N_Y}) \in \mathbb{R}^{N_Y}$ at each assimilation step k is given by the following steps:

(1) For each $i = 1, ..., N_Y$, choose coordinates, x_{coord}^i, where observations are available at.

(2) Define the observations as

$$y_{\text{data},t_k}^i = \phi_{t_k}^{(h,\delta x_l),\text{ref}}(x_{\text{coord}}^i) + \xi^i, \tag{5.5}$$

where $\xi^i \sim N(0, R)$, and $R > 0$ is a scalar measurement error variance.

To proceed in the filtering framework, one is required to find the importance weights for the forecast ensemble $\left\{X_t^{(h,\delta x_l),i}\right\}_{i=1}^N$ and transform them into an evenly weighted analysis ensemble $\left\{\tilde{X}_t^{(h,\delta x_l),i}\right\}_{i=1}^N$. To do this, one could simply find weights and transformations for each of the basis coefficients within $X_t^{(h,\delta x_l)}$ individually. However, the function approximations resulting from the analysis ensemble of basis coefficient vectors, $\left\{\tilde{\phi}_{t_k}^{(h,\delta x_l),i}(x)\right\}_{i=1}^N$, would almost certainly lack spatial regularity and the state would be extremely discontinuous. Therefore, in contrast to Sec. 2.7.7 where localisation is used to reduce the dimensionality of the likelihood functions, a localisation scheme is required here to smooth out the expected discontinuities in the analysis function approximations.

5.4.2 Localisation and implementation

This localisation scheme is an extension of the one presented in Sec. 2.7.7 for the ETPF, however instead of using a localisation matrix in the likelihood and transformation computations, a projection operator is used. This is explained in more detail in the following

sections: Sec. 5.4.2 covers the computation of importance weights for the forecast ensemble $\left\{X_{t_k}^{(h,\delta x_l),i}\right\}_{i=1}^N$ and Sec. 5.4.2 covers the construction of a transformation to the analysis ensemble $\left\{\tilde{X}_{t_k}^{(h,\delta x_l),i}\right\}_{i=1}^N$. The experiments used throughout the current and next chapter employ Firedrake, developed by Imperial College London [Rathgeber et al., 2016, Zenodo/Firedrake, 2017]. This allows one to construct mesh and function space hierarchies, projection operators, and simulate the discretized forward prediction model, $M^{(h,\delta x_l)}$. The package FADE (Firedrake and Assimilating Data using Ensembles) is built on top of Firedrake to implement the localisation scheme below, and is available open source from `https://github.com/alsgregory/FADE`, where one can also find documentation and examples.

Importance weights

At each assimilation step, $k = 1, 2, 3, ...$, one is required to update importance weights for each of the forecast ensemble members $\left\{X_{t_k}^{(h,\delta x_l),i}\right\}_{i=1}^N$ given the observations, as done in (2.59). The likelihood of the i'th forecast ensemble member, $\pi_{Y_{t_k}}(y_{\text{data},t_k}|X_{t_k}^{(h,\delta x_l),i})$, needs to be evaluated in order to do this. This likelihood can be approximated as follows.

(1) Start by initialising the piecewise constant functions $D_{t_k}^{(h,\delta x_l),i}(x) \in V^{P0,\delta x_l}$, for $i = 1, ..., N$, where $V^{P0,\delta x_l}$ is a piecewise constant function space, for which there is one basis coefficient per cell, $c^{\delta x_l} \in \mathcal{M}^{\delta x_l}$. Then compute

$$D_{t_k}^{(h,\delta x_l),i}(c^{\delta x_l}) = \sum_{j \in S} \left(y_{\text{data},t_k}^j - \mathbb{I}_{V^{\delta x_l}}^{V^{P0,\delta x_l}} \phi_{t_k}^{(h,\delta x_l),i}(c^{\delta x_l}) \right)^2, \tag{5.6}$$

for $i = 1, ..., N$. Here $S = \left\{ j; \quad x_{\text{coord}}^j \in c^{\delta x_l} \right\}$. This cumulates all squared differences between the value of each forecast ensemble member, projected onto $V^{P0,\delta x_l}$, at a particular cell on the mesh and all observations taken at coordinates within that cell.

(2) Project $D_{t_k}^{(h,\delta x_l),i}(x)$ onto the function space $V^{\delta x_l}$, computing $\mathbb{I}_{V^{P0,\delta x_l}}^{V^{\delta x_l}} D_{t_k}^{(h,\delta x_l),i}(x)$, for each $i = 1, ..., N$.

(3) For a given parameter, $r_{loc} \in [0, l]$, compute the injected function

$$\mathbb{I}_l^{(l-r_{loc})} \mathbb{I}_{V^{P0,\delta x_l}}^{V^{\delta x_l}} D_{t_k}^{(h,\delta x_l),i}(x),$$

for each $i = 1, ..., N$.

(4) Compute the prolonged function

$$\mathbb{I}^{l}_{(l-r_{loc})}\mathbb{I}^{(l-r_{loc})}_{l}\mathbb{I}^{V^{\delta x_l}}_{V^{P0,\delta x_l}}D^{(h,\delta x_l),i}_{t_k}(x),$$

for each $i = 1, ..., N$. Whilst this may seem counterintuitive ($\mathbb{I}^{V^{\delta x_l}}_{V^{P0,\delta x_l}}D^{(h,\delta x_l),i}_{t_k}(x)$ has been injected onto a coarser mesh only to prolong it back to the original mesh), it serves a good purpose. It has cumulated all the squared differences between the forecast ensemble members and observations together from all finer subcells within a coarser cell. It then replaced these individual differences with that cumulated total. Therefore, a level of localisation is achieved. The greater the value of r_{loc}, the greater number of finer subcells are cumulated together. For the benefit of notation, denote the operator $\mathbb{I}^{l}_{(l-r_{loc})}\mathbb{I}^{(l-r_{loc})}_{l}$ by $\mathcal{I}_{(l,r_{loc})}$.

(5) To smooth out discontinuities in $\mathcal{I}_{(l,r_{loc})}\mathbb{I}^{V^{\delta x_l}}_{V^{P0,\delta x_l}}D^{(h,\delta x_l),i}_{t_k}(x)$ at the boundaries of the coarser cells on the mesh $\mathcal{M}^{\delta x_{l-r_{loc}}}$, this function can be projected onto $V^{P0,\delta x_l}$ (to give each cell a single basis coefficient) and then onto a continuous piecewise linear function space, $V^{CG1,\delta x_l}$. This allows the basis coefficients associated with the vertices of each cell to take contributions from the basis coefficients of all the cells that contain that vertex. One needs to then project this function back to onto $V^{P0,\delta x_l}$, and then finally onto the original function space $V^{\delta x_l}$. Denote the operator of this process by $\mathcal{I}_{\text{smooth}}$. The effects of both components within this operator are shown in Figures 17 and 18.

(6) Initialise the functions (importance weights), $\tilde{w}^{(h,\delta x_l),i}_{t_k}(x)$, on the space $V^{\delta x_l}$, for each $i = 1, ..., N$ and set

$$\tilde{w}^{(h,\delta x_l),i}_{t_k}(x) = \frac{\frac{1}{\sqrt{2\pi R}}\exp\left(-\frac{1}{2R}\nu^{*}\mathcal{I}_{\text{smooth}}\mathcal{I}_{(l,r_{loc})}\mathbb{I}^{V^{\delta x_l}}_{V^{P0,\delta x_l}}D^{(h,\delta x_l),i}_{t_k}(x)\right)}{\sum_{j=1}^{N}\frac{1}{\sqrt{2\pi R}}\exp\left(-\frac{1}{2R}\nu^{*}\mathcal{I}_{\text{smooth}}\mathcal{I}_{(l,r_{loc})}\mathbb{I}^{V^{\delta x_l}}_{V^{P0,\delta x_l}}D^{(h,\delta x_l),j}_{t_k}(x)\right)}. \tag{5.7}$$

Here, ν^{*} is a scaling constant, namely the number of finer subcells, $c^{\delta x_l}$, in the cell $c^{\delta x_{l-r_{loc}}} \in \mathcal{M}^{\delta x_{l-r_{loc}}}$ that contains them. This also assumes that $\tilde{w}^{(h,\delta x_l),i}_{t_{k-1}}(x) = 1/N$, for $i = 1, ..., N$, as utilised in the ETPF framework for the case where a transform is implemented on every assimilation step.

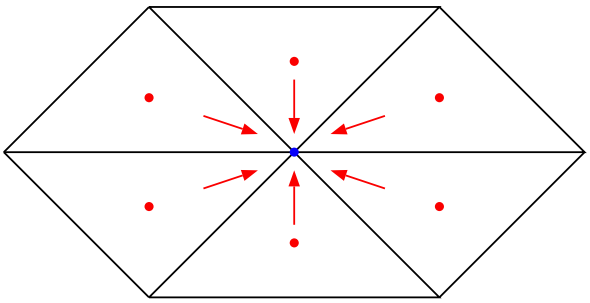


Figure 17: The projection from $V^{P0,\delta x_l}$ to $V^{CG1,\delta x_l}$ assigns a basis coefficient at a given vertex as the sum of all basis coefficients at the centers of the cells containing that vertex.

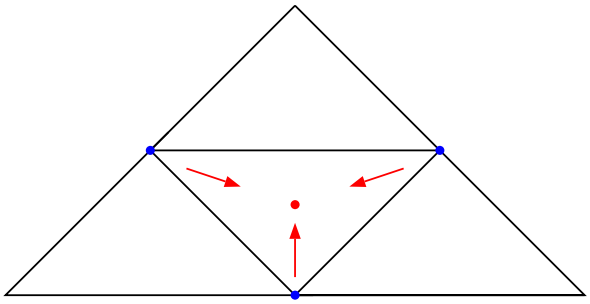


Figure 18: The projection from $V^{CG1,\delta x_l}$ to $V^{P0,\delta x_l}$ assigns a basis coefficient at each cell center as the mean of all basis coefficients at the vertices within the cell.

These are the weighting functions that approximate the likelihood of each ensemble member in $\left\{\phi_{t_k}^{(h,\delta x_l),i}(x)\right\}_{i=1}^{N}$. Finally define the vectors of basis coefficients for each of these weighting functions $\left\{\tilde{w}_{t_k}^{(h,\delta x_l),i}(x)\right\}_{i=1}^{N}$ as

$$W_{t_k}^{(h,\delta x_l),i} = \left(b_1^{\delta x_l,i}, ..., b_{N_X^{\delta x_l}}^{\delta x_l,i}\right), \tag{5.8}$$

for $i = 1, ..., N$. As a demonstration of the localisation technique above, the following functions will be investigated:

$$G_{r_{loc}}(x) = \nu^* \mathcal{I}_{(3,r_{loc})} D(x), \qquad r_{loc} \in [0, 3],$$

where $D \in V^{P0,\delta x_3}$ is a piecewise constant function in one geometric dimension on the mesh $\mathcal{M}^{\delta x_3}$. Here, Γ is the unit interval. The mesh $\mathcal{M}^{\delta x_3}$ belongs on a hierarchy defined by $\delta x_l = 2^{-l}$ being the cell width. Therefore the coarsest mesh in the hierarchy, $\mathcal{M}^{\delta x_0}$, has just one cell. One also notes that $\nu^* = 2^{r_{loc}}$ here. All basis coefficients, apart from the one associated with the fifth cell, are set to 0. The fifth basis coefficient, $a_5^{\delta x_3}$, is set to 2. Figure 19 shows the functions $G_{r_{loc}}(x)$, with $r_{loc} = 0, 1, 2$ and 3 respectively. Now let $F(x) = G_2(x)$. Figure 20 shows $F(x)$ and $\mathcal{I}_{\text{smooth}} F(x)$. The discontinuity that appeared in $G_2(x)$, at $x = 0.5$, has now been partially smoothed by the introduction of Step (5) in the localisation scheme.

Remark 2. *It is important to note that the methodology assumes independence between the measurement error associated with each observation. Therefore for two observations* y_{data}^p, y_{data}^q, *where* $p, q = 1, ..., N_Y$ *and* $p \neq q$, *one notes that* $\text{Cov}[\xi^p, \xi^q] = 0$, *even if*

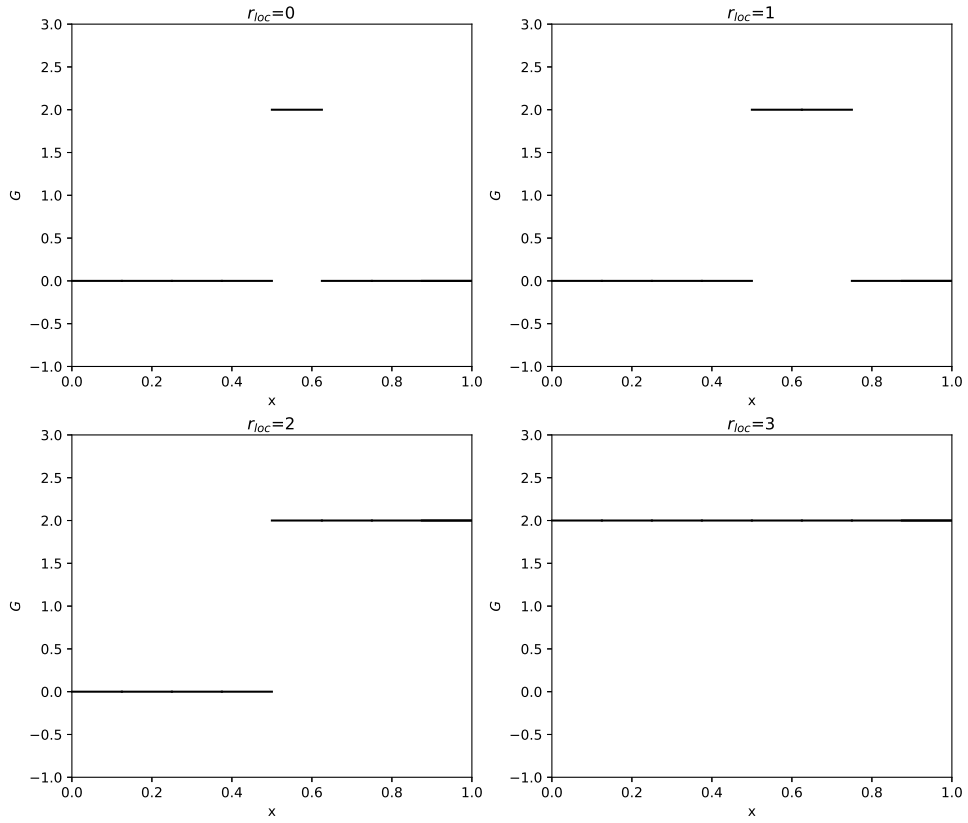


Figure 19: The function $G_{r_{loc}}(x) = \nu^* \mathbb{1}_{(3,r_{loc})} D(x)$ for the values $r_{loc} = 0, 1, 2$ and 3 and $x \in [0, 1]$. Here D is a piecewise constant function, with the fifth out of 8 basis coefficients set to 2 and the rest set to 0. As r_{loc} increases, the coarse cells are assigned the sum of the finer subcells within them.

$\|x^p_{coord} - x^q_{coord}\| < \delta$, *for some small* δ.

Using these importance weight vectors, one can compute an approximation to the posterior mean $\mathbb{E}[\tilde{X}^{(h,\delta x_l)}_{t_k}]$ via

$$\mathbb{E}[\tilde{X}^{(h,\delta x_l)}_{t_k}] \approx \sum_{i=1}^{N} \left(X^{(h,\delta x_l),i}_{t_k} \odot W^{(h,\delta x_l),i}_{t_k} \right), \tag{5.9}$$

where $\odot$ is the Hadamard product. As detailed in Chapter 2, the ETPF now proceeds by replacing this approximation with another one that uses an analysis ensemble $\left\{ \tilde{X}^{(h,\delta x_l),i}_{t_k} \right\}_{i=1}^{N}$ constructed from a transformation to the weighted forecast ensemble. The next section shows

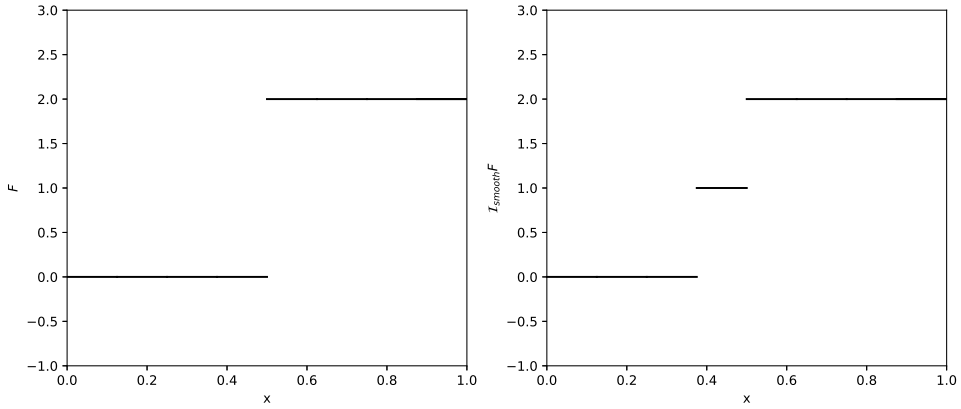


Figure 20: The functions $F(x) = G_2(x)$ and $\mathcal{I}_{\text{smooth}}F(x)$. The function $G_2(x)$ is defined as it was in Figure 19. With the application of $\mathcal{I}_{\text{smooth}}$, the discontinuity in $F(x)$ between the two cells on the coarsest mesh is smoothed out.

how this can be done in a localised manner for the spatio-temporal case considered in this chapter.

A transformation from forecast to analysis ensembles

Let the components of the analysis ensemble members in $\left\{\tilde{X}_{t_k}^{(h,\delta x_l),i}\right\}_{i=1}^{N}$ be the basis coefficients $\left\{\tilde{a}_j^{\delta x_l,i}\right\}_{j=1}^{N_X^{\delta x_l}}$, for $i = 1, ..., N$, of a member in an analysis ensemble of function approximations $\left\{\tilde{\phi}_{t_k}^{(h,\delta x_l),i}(x)\right\}_{i=1}^{N}$. I.e.

$$\tilde{\phi}_{t_k}^{(h,\delta x_l),i}(x) = v^{\delta x_l}(x) \cdot \tilde{a}^{\delta x_l,i},$$

where

$$\tilde{X}_{t_k}^{(h,\delta x_l),i} = \left(\tilde{a}_1^{\delta x_l,i}, ..., \tilde{a}_{N_X^{\delta x_l}}^{\delta x_l,i}\right). \tag{5.10}$$

The transformation is given in the form

$$\tilde{a}_n^{\delta x_l,j} = \sum_{i=1}^{N} N T_{i,j}^n a_n^{\delta x_l,i}, \tag{5.11}$$

for each basis coefficient, $n = 1, ..., N_X^{\delta x_l}$, and for $j = 1, ..., N$. Here, $T^n \in \mathcal{T}^n \in \mathbb{R}^{N\times N}$ is an individual coupling matrix for each basis coefficient. The transformation matrices satisfy the following marginal constraints using the components of the importance weight vector,

$W_{t_k}^{(h,\delta x_l),i}$:

$$\sum_{i=1}^{N} T_{i,j}^{n} = \frac{1}{N}, \qquad \sum_{j=1}^{N} T_{i,j}^{n} = b_n^{\delta x_l, i}.$$

The matrices are found by solving

$$\arg\min_{T^n \in \mathcal{T}^n} \sum_{i=1}^{N}\sum_{j=1}^{N} T_{i,j}^{n} e_n^{(i,j)}.$$

Here, $e_n^{(i,j)}$ is a cost metric, defined by the n'th basis coefficient of the following function $J^{(i,j)}(x) \in V^{\delta x_l}$:

$$J^{(i,j)}(x) = \mathcal{I}_{\text{smooth}}\mathcal{I}_{(l,r_{loc})}\left\{\left(\phi_{t_k}^{(h,\delta x_l),i}(x) - \phi_{t_k}^{(h,\delta x_l),j}(x)\right)^2\right\}. \tag{5.12}$$

As done for the computation of the importance weights, this function cumulates the squared differences between the i'th and j'th forecast ensemble members for all local finer subcells, on $\mathcal{M}^{\delta x_l}$, within a coarse cell, on $\mathcal{M}^{\delta x_{l-r_{loc}}}$. It also utilises the smoothing operator introduced earlier to smooth discontinuities between coarse cell boundaries. Note that, as was the case within the ETPF framework for multivariate random variables, the parameter r_{loc} used in (5.12) can be independent of that used during the importance weight computation.

This transformation allows us to re-write the approximation for the expected value of $\tilde{X}_{t_k}^{(h,\delta x_l)}$ in (5.9) as

$$\mathbb{E}[\tilde{X}_{t_k}^{(h,\delta x_l)}] \approx \frac{1}{N}\sum_{i=1}^{N} \tilde{X}_{t_k}^{(h,\delta x_l),i}. \tag{5.13}$$

These estimates can be re-written in terms of the analysis ensemble of function approximations $\left\{\tilde{\phi}_{t_k}^{(h,\delta x_l),i}(x)\right\}_{i=1}^{N}$ whose basis coefficients are the components of the random variables $\left\{\tilde{X}_{t_k}^{(h,\delta x_l),i}\right\}_{i=1}^{N}$. Importantly, define an estimate to $\mathbb{E}[\tilde{\phi}_{t_k}^{(h,\delta x_l)}(x)]$ as

$$\bar{\phi}_{N,t_k}^{(h,\delta x_l)}(x) = \frac{1}{N}\sum_{i=1}^{N} \tilde{\phi}_{t_k}^{(h,\delta x_l),i}(x). \tag{5.14}$$

This chapter will now proceed by introducing an example of a system of equations that can be approximated via finite element methods, the stochastic quasi-geostrophic equations. In Sec. 5.5.1, it is then used in combination with the ETPF localisation framework, that has been proposed throughout this section, for data assimilation.

5.5 The stochastic quasi-geostrophic equations

The stochastic quasi-geostrophic system of equations is given by,

$$dq_t(x,y) + \nabla \cdot \left(q_t(x,y) \left(\nabla^{\perp}\psi_t(x,y)dt + \sum_{i=1}^{\mathcal{N}} \nabla^{\perp}\zeta_i(x,y) \circ dW_t \right) \right) = 0, \tag{5.15}$$

where $\psi_t(x,y)$ is the streamfunction and the potential vorticity is

$$q_t(x,y) = \nabla^2\psi_t(x,y) + \beta y + f\psi_t(x,y). \tag{5.16}$$

Here $f = 1$ and the β-plane constant takes the value of 0.5. The equation (5.16) can be written as an elliptic function of $\psi_t(x,y)$,

$$q_t(x,y) = \mathcal{L}\left(\psi_t(x,y)\right),$$

and therefore,

$$\psi_t(x,y) = \mathcal{L}^{-1}\left(q_t(x,y)\right).$$

This model was introduced in Holm [2015]. For the remainder of this thesis, the Kahunen-Loeve (KL) expansion in (5.15) truncates at $\mathcal{N} = 4$ and

$$\zeta_i(x,y) = \begin{cases} \nu \sin(\pi x)\sin(\pi y), & i = 1 \\ \nu \sin(2\pi x)\sin(\pi y), & i = 2 \\ \nu \sin(\pi x)\sin(2\pi y), & i = 3 \\ \nu \sin(2\pi x)\sin(2\pi y), & i = 4. \end{cases} \tag{5.17}$$

The parameter ν is the variance of the stochasticity in the system. The domain is set to $(x,y) \in [0,1]^2 = \Gamma$ and the initial condition takes the value of

$$q_0(x,y) = \sum_{i=1}^{3} (-1)^{i+1} \left(p(x,y,z_i,0.7) - p(x,y,z_i,0.3) \right), \tag{5.18}$$

where

$$z = (0.25, 0.5, 0.75) \quad \text{and} \quad p(x,y,x_0,y_0) = \exp\left(-\frac{(x-x_0)^2}{0.02} - \frac{(y-y_0)^2}{0.02} \right).$$

The initial condition for $\psi_t(x,y)$ is then given by

$$\psi_0(x,y) = \mathcal{L}^{-1}\left(q_0(x,y)\right).$$

The equations will be spatially and temporally discretized with a finite element method [Bernsen et al., 2006, Zenodo/Firedrake, 2017] and a SSP Runge-Kutta three-step time-stepping scheme. More detail will be given on the time-stepping scheme later in the chapter. Denote these discretizations of $q_t(x,y)$ and $\psi_t(x,y)$ by $q_t^{(h,\delta x_l)}(x,y)$ and $\psi_t^{(h,\delta x_l)}(x,y)$ respectively. These are discontinuous piecewise linear, $q_t^{(h,\delta x_l)}(x,y) \in V^{DG1,\delta x_l}$, and continuous piecewise linear, $\psi_t^{(h,\delta x_l)}(x,y) \in V^{CG1,\delta x_l}$, functions respectively. The spatial resolution, $\delta x_l = 2^{-l-1}$, for $l = 5$, is the width *and* height of each cell in a regular, triangular mesh, $\mathcal{M}^{\delta x_l}$. Therefore one must use a r_{loc} value between 0 and 5 for this problem. Note that the time-step used in the Runge-Kutta scheme remains fixed and not part of a hierarchy - it is set to $h = 0.005$.

For brevity, the following spatial discretization of (5.15) adopts the short-hand notation ψ_t and q_t for $\psi_t^{(h,\delta x_l)}(x,y)$ and $q_t^{(h,\delta x_l)}(x,y)$ respectively,

$$d\int_{c^{\delta x_l}} pq_t dxdy - \int_{c^{\delta x_l}} \nabla p \cdot (q_t d\underline{\mathrm{u}})\, dxdy + \int_{\partial c^{\delta x_l}} p(\underline{\mathrm{n}} \cdot d\underline{\mathrm{u}})\hat{f} dS = 0, \tag{5.19}$$

with

$$d\underline{\mathrm{u}} = \nabla^{\perp}\psi_t dt + \sum_{i=1}^{\mathcal{N}} \nabla^{\perp}\zeta_i \circ dW_t,$$

where $p(x,y) \in V^{DG1,\delta x_l}$ is a test function, $\underline{\mathrm{n}}$ is the unit normal, and $\partial c^{\delta x_l}$ represents the boundary of the cell $c^{\delta x_l}$. Here, $\hat{f}$, is an upwind numerical flux [Cockburn and Shu, 2001], where it takes the value of q_t inside the cell if $(\underline{\mathrm{n}} \cdot d\underline{\mathrm{u}}) \geq 0$ and outside the cell if $(\underline{\mathrm{n}} \cdot d\underline{\mathrm{u}}) < 0$.

Let the random variable $X_t^{(h,\delta x_l),q}$ be the vector of the basis coefficients for the approximation $q_t^{(h,\delta x_l)}(x,y)$. In the same way, let the random variable $X_t^{(h,\delta x_l),\psi}$ be the vector of basis coefficients for the approximation $\psi_t^{(h,\delta x_l)}(x,y)$. One can interpret the temporal discretization of (5.15) as simply a Stratonovich SDE with respect to the basis coefficients of $q_t^{(h,\delta x_l)}(x,y)$:

$$dX_t^{(h,\delta x_l),q} = \mu\left(X_t^{(h,\delta x_l),q}\right) dt + \sigma\left(X_t^{(h,\delta x_l),q}\right) \circ dW_t, \tag{5.20}$$

where the functions μ and σ relate to the spatial discretization above. The following three-step SSP Runge-Kutta scheme [Gottlieb et al., 2001] is used for the temporal discretization

of this SDE, with $\mathcal{O}(h)$ weak convergence and $\mathcal{O}(h^{1/2})$ strong convergence. It is given by

$$k_1 = X_t^{(h,\delta x_l),q} + \mu\left(X_t^{(h,\delta x_l),q}\right) h + \sigma\left(X_t^{(h,\delta x_l),q}\right) \delta W_t, \tag{5.21}$$

$$k_2 = \frac{3}{4} X_t^{(h,\delta x_l),q} + \frac{1}{4}\left(k_1 + \mu\left(k_1\right) h + \sigma\left(k_1\right) \delta W_t\right), \tag{5.22}$$

$$X_{(t+h)}^{(h,\delta x_l),q} = \frac{1}{3} X_t^{(h,\delta x_l),q} + \frac{2}{3}\left(k_2 + \mu\left(k_2\right) h + \sigma\left(k_2\right) \delta W_t\right), \tag{5.23}$$

where $\delta W_t \sim N(0,h)$. The stability of this scheme for the case where $\left\{\zeta_i\right\}_{i=1}^{\mathcal{N}} = 0$, used in combination with the spatial discretization considered in this chapter, with details about the associated critical Courant numbers, is discussed in Cockburn and Shu [2001]. Finally then, to propagate the system from time t to $t+h$, one implements the following algorithm:

(1) Find k_1 using (5.21).

(2) Find k_2 using (5.22).

(3) Find $q_{(t+h)}^{(h,\delta x_l)}(x,y)$ using (5.23) and solve $\psi_{(t+h)}^{(h,\delta x_l)}(x,y) = \mathcal{L}^{-1}\left(q_{(t+h)}^{(h,\delta x_l)}(x,y)\right)$.

5.5.1 An ensemble transform particle filter approximation

Using the stochastic quasi-geostrophic equations in combination with the filtering framework presented in Sec. 5.4, the aim of this section is to approximate statistics of the analysis functions $\tilde{\psi}_{t_k}^{(h,\delta x_l)}(x,y)$ and $\tilde{q}_{t_k}^{(h,\delta x_l)}(x,y)$, for $k = 1,2,3,...$, given observations $y_{\text{data},t_1:t_k}$. These observations are in the form of (5.5), taken at coordinates $\left\{x_{\text{coord}}^j\right\}_{j=1}^{N_Y}$. Only a reference trajectory of the streamfunction $\psi_{t_k}^{(h,\delta x_l)}(x,y)$ is observed for this example. To estimate the statistics of both $\tilde{\psi}_{t_k}^{(h,\delta x_l)}(x,y)$ and $\tilde{q}_{t_k}^{(h,\delta x_l)}(x,y)$, the ETPF approximation in (5.14) is used in combination with the filtering and localisation framework proposed in Sec. 5.4.

For the forecast ensemble, pairs of function approximations $\left\{q_{t_k}^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^{N}$ and $\left\{\psi_{t_k}^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^{N}$ are used. Recall the definitions of the two random variables $X_{t_k}^{(h,\delta x_l),q}$ and $X_{t_k}^{(h,\delta x_l),\psi}$ from the last section. Ensembles of these random variables will be transformed to analysis ensembles, using (5.11). The importance weights for each forecast ensemble pair will only be dependent on the observed streamfunction. Those importance weight functions will then be projected to the spaces $V^{CG1,\delta x_l}$ and $V^{DG1,\delta x_l}$ for the transformations of $\left\{\psi_{t_k}^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^{N}$ and $\left\{q_{t_k}^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^{N}$ respectively. These procedures are explained in more depth in the algorithm below.

(1) Whilst $t < t_k$, set $h = \min(0.005, t_k - t)$ and propagate $\left\{\psi_t^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^N$ and $\left\{q_t^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^N$ to time t_k.

(2) Generate observations $\left\{y^j_{\text{data},t_k}\right\}_{j=1}^{N_Y}$ using a reference trajectory of the streamfunction via

$$y^j_{\text{data},t_k} = \psi_{t_k}^{(h,\delta x_l),\text{ref}}(x^j_{\text{coord}}, y^j_{\text{coord}}) + \xi^j, \tag{5.24}$$

given coordinates $(x^j_{\text{coord}}, y^j_{\text{coord}})$, for $j = 1, ..., N_Y$.

(3) Define the forecast ensemble of random variables $\left\{X_{t_k}^{(h,\delta x_l),\psi,i}\right\}_{i=1}^N$ and $\left\{X_{t_k}^{(h,\delta x_l),q,i}\right\}_{i=1}^N$ whose components are the basis coefficients of $\left\{\psi_{t_k}^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^N$ and $\left\{q_{t_k}^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^N$ respectively.

(4) From Sec. 5.4.2, compute the following piecewise constant functions using

$$D_{t_k}^{(h,\delta x_l),i}(c^{\delta x_l}) = \sum_{j\in S}\left(y^j_{\text{data},t_k} - \mathbb{I}_{V^{CG1,\delta x_l}}^{V^{P0,\delta x_l}}\psi_{t_k}^{(h,\delta x_l),i}(c^{\delta x_l})\right)^2,$$

for $i = 1, ..., N$ and each cell $c^{\delta x_l} \in \mathcal{M}^{\delta x_l}$, where $S = \left\{j; \quad (x^j_{\text{coord}}, y^j_{\text{coord}}) \in c^{\delta x_l}\right\}$. Then the importance weight vectors are defined as the basis coefficients of the following functions:

$$\tilde{w}_{t_k}^{(h,\delta x_l),\psi,i}(x) = \frac{\frac{1}{\sqrt{2\pi R}}\exp\left(-\frac{1}{2R}\nu^*\mathcal{I}_{\text{smooth}}\mathcal{I}_{(l,r_{loc})}\mathbb{I}_{V^{P0,\delta x_l}}^{V^{CG1,\delta x_l}}D_{t_k}^{(h,\delta x_l),i}(x)\right)}{\sum_{j=1}^N \frac{1}{\sqrt{2\pi R}}\exp\left(-\frac{1}{2R}\nu^*\mathcal{I}_{\text{smooth}}\mathcal{I}_{(l,r_{loc})}\mathbb{I}_{V^{P0,\delta x_l}}^{V^{CG1,\delta x_l}}D_{t_k}^{(h,\delta x_l),j}(x)\right)}$$

and

$$\tilde{w}_{t_k}^{(h,\delta x_l),q,i}(x) = \frac{\frac{1}{\sqrt{2\pi R}}\exp\left(-\frac{1}{2R}\nu^*\mathcal{I}_{\text{smooth}}\mathcal{I}_{(l,r_{loc})}\mathbb{I}_{V^{P0,\delta x_l}}^{V^{DG1,\delta x_l}}D_{t_k}^{(h,\delta x_l),i}(x)\right)}{\sum_{j=1}^N \frac{1}{\sqrt{2\pi R}}\exp\left(-\frac{1}{2R}\nu^*\mathcal{I}_{\text{smooth}}\mathcal{I}_{(l,r_{loc})}\mathbb{I}_{V^{P0,\delta x_l}}^{V^{DG1,\delta x_l}}D_{t_k}^{(h,\delta x_l),j}(x)\right)}$$

on the spaces $V^{CG1,\delta x_l}$ and $V^{DG1,\delta x_l}$ respectively, for $i = 1, ..., N$.

(5) Using Sec. 5.4.2, transform the weighted forecast ensembles of random variables

$$\left\{X_{t_k}^{(h,\delta x_l),\psi,i}\right\}_{i=1}^N \text{ and } \left\{X_{t_k}^{(h,\delta x_l),q,i}\right\}_{i=1}^N,$$

to analysis ensembles of random variables

$$\left\{\tilde{X}_{t_k}^{(h,\delta x_l),\psi,i}\right\}_{i=1}^{N} \text{ and } \left\{\tilde{X}_{t_k}^{(h,\delta x_l),q,i}\right\}_{i=1}^{N},$$

whose components are the basis coefficients for analysis ensembles of functions

$$\left\{\tilde{\psi}_{t_k}^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^{N} \text{ and } \left\{\tilde{q}_{t_k}^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^{N},$$

respectively. The importance weight vectors $\left\{W_{t_k}^{(h,\delta x_l),\psi,i}\right\}_{i=1}^{N}$ and $\left\{W_{t_k}^{(h,\delta x_l),q,i}\right\}_{i=1}^{N}$ are used for each of these transformations respectively.

(6) Estimate $\mathbb{E}[\tilde{\psi}_{t_k}^{(h,\delta x_l)}(x,y)]$ using

$$\bar{\psi}_{N,t_k}^{(h,\delta x_l)}(x,y) = \frac{1}{N}\sum_{i=1}^{N}\tilde{\psi}_{t_k}^{(h,\delta x_l),i}(x,y), \tag{5.25}$$

and $\mathbb{E}[\tilde{q}_{t_k}^{(h,\delta x_l)}(x,y)]$ using

$$\bar{q}_{N,t_k}^{(h,\delta x_l)}(x,y) = \frac{1}{N}\sum_{i=1}^{N}\tilde{q}_{t_k}^{(h,\delta x_l),i}(x,y). \tag{5.26}$$

(7) Reset the forecast ensembles $\psi_t^{(h,\delta x_l),i}(x,y) = \tilde{\psi}_{t_k}^{(h,\delta x_l),i}(x,y)$ and $q_t^{(h,\delta x_l),i}(x,y) = \tilde{q}_{t_k}^{(h,\delta x_l),i}(x,y)$, for $i = 1, ..., N$, and iterate $k = k + 1$.

It is important to remember that this framework is an example of an identical twin experiment. This creates an idealised environment to test a data assimilation algorithm given that there is no model error, only approximation error.

5.6 Numerical demonstrations

The algorithm in Sec. 5.5.1 is now demonstrated. For the following section, the random component of $\psi_t^{(h,\delta x_l)}(x,y)$, is removed until $t = 10$. One can do this by setting $\left\{\zeta_i\right\}_{i=1}^{\mathcal{N}} = 0$. This is because the state at $t = 10$ is a more complex and interesting initial condition than the state at $t = 0$. It speeds up the implementation of the algorithm in Sec. 5.5.1 considerably by only simulating one deterministic realisation of $\psi_t^{(h,\delta x_l)}(x,y)$ over the interval $t \in [0, 10]$

and then starting all random ensembles from this point. The ensembles are then simulated over the time interval $t \in [10, 80]$.

The assimilation in the following numerical demonstrations use $\Delta t = 0.5$. For simplicity, the coordinates of the observations are given on the grid specified by

$$x^j_{\text{coord}} = \frac{\lfloor (j-1)/32 \rfloor + 1/2}{32}, \qquad y^j_{\text{coord}} = \frac{((j-1) \mod 32) + 1/2}{32},$$

for $j = 1, ..., 1024$. Two scenarios will be considered, one where assimilation is started right from the time $t = 10$, and thus $k \in [1, 140]$, and one where assimilation starts from a delayed point in time, $t = 30$, and thus $k \in [1, 100]$. The latter of these scenarios should be more challenging, testing the filter in honing down on a reference trajectory from an ensemble at $t = 30$ with large variance. The observations are defined by (5.24), where $R = 2 \times 10^{-9}$. The following demonstrations will use $\nu = 3.125 \times 10^{-4}$. The localisation setting of $r_{loc} = 2$ is used. Finally, $N = 30$ ensemble members will be used in both experiments.

5.6.1 Assimilation from a known state

For the first scenario, the algorithm in Sec. 5.5.1 is used to assimilate over the time interval $t \in [10, 80]$, and so assimilation starts from the time when all forecast members start from a known state at time $t = 10$. The analysis ensemble $\{\tilde{\psi}^{(h,\delta x_l),i}_{t_k}(x, y)\}^N_{i=1}$ at each assimilation step is recorded and the timeseries of these ensemble members evaluated at the coordinates

$$(x_r, y_r) = \{(0.3, 0.3), (0.7, 0.3), (0.3, 0.7), (0.7, 0.7)\},$$

are shown in Figure 21 in the dark grey dots. Together they form a highlighted region honed down on the reference trajectory, $\psi^{(h,\delta x_l),\text{ref}}_t(x, y)$, shown in red. The spread of this analysis ensemble over time is significantly less than the spread of 30 simulations of $\psi^{(h,\delta x_l)}_t(x, y)$, denoted by $\{\hat{\psi}^{(h,\delta x_l),i}_t(x, y)\}^N_{i=1}$, without the use of filtering. This ensemble is shown by the light grey lines. These simulations give an indication of the variance within the system in comparison to the variance of the filtered analysis distribution.

The state of the streamfunction estimator, $\bar{\psi}^{(h,\delta x_l)}_{N,t_k}(x, y)$, and the associated reference trajectory, $\psi^{(h,\delta x_l),\text{ref}}_{t_k}(x, y)$, at times $t = 42.5$ and $t = 75$ are shown in Figures 22 and 23 respectively. In the same manner, the state of the potential vorticity estimator, $\bar{q}^{(h,\delta x_l)}_{N,t_k}(x, y)$, and the trajectory, $q^{(h,\delta x_l),\text{ref}}_{t_k}(x, y)$, associated with the reference streamfunction through (5.16), at

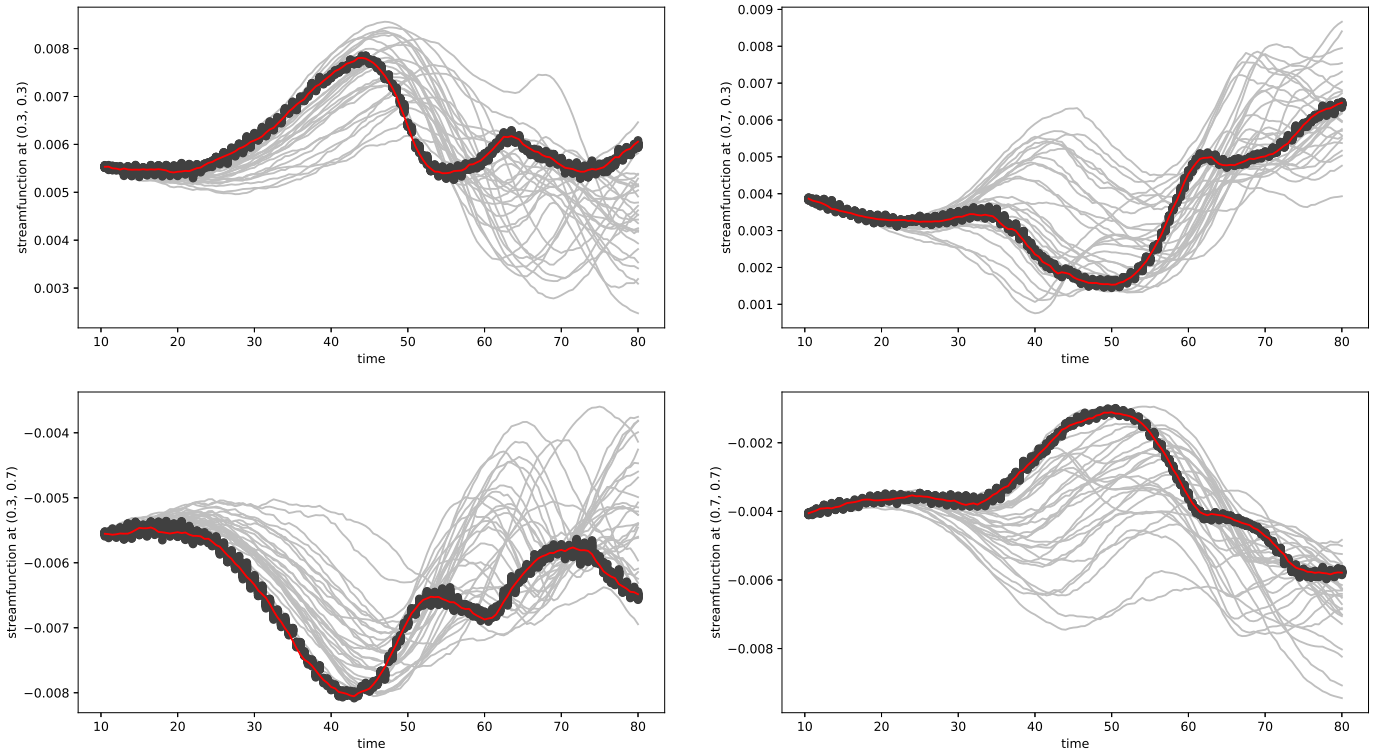


Figure 21: The values of $\left\{\tilde{\psi}_{t_k}^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^{N}$ evaluated at the coordinate set (x_r, y_r) and $k \in [1, 140]$, used within the ETPF approximation to the filtered streamfunction, and are shown by the dark grey area of dots. The reference trajectory, from which the observations are taken from during assimilation, is shown evaluated at (x_r, y_r) by the red line. The variance of the analysis ensemble members around the reference trajectory is significantly less than that of N random trajectories of the system without data assimilation, shown by the light grey lines.

times $t = 42.5$ and $t = 75$ are shown in Figures 24 and 25 respectively. Despite the potential vorticity not being observed, the ETPF estimator tracks the state of this reference field well.

Localisation has been used in this problem, with the parameter $r_{loc} = 2$, and so the optimal transport problems within the transformation scheme (see Sec. 5.4.2) still have to be solved via an expensive algorithm (e.g. the problems are multivariate). Despite this, it is interesting to note that the computational cost of the forward prediction model in this problem is still marginally superior to that of the total optimal transportation cost: $\sim$75 minutes to $\sim$50 minutes. This superiority will be accentuated in the next chapter when the multilevel ETPF estimator is introduced.

5.6.2 Delayed assimilation

For the second scenario, the forecast ensemble is simulated from $t = 10$, however the start of assimilation is delayed by 20 time units from $t = 10$ to $t = 30$. The algorithm in Sec. 5.5.1 is then used to assimilate over the time interval $t \in [30, 80]$. This allows the forecast ensemble to spread out significantly and test the assimilation strategy in its attempts to lower the variance of this ensemble. As with the previous section, the dark grey dots in Figure 26 show the timeseries of each analysis ensemble member in $\left\{\tilde{\psi}_{t_k}^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^{N}$ evaluated at (x_r, y_r). The reference trajectory and an ensemble of N random simulations of $\psi_t^{(h,\delta x_l)}(x,y)$ without the use of filtering, all evaluated at (x_r, y_r), are also shown.

It is also interesting to study the error of the approximation to the analysis mean in (5.25) for this delayed assimilation scenario. A suitable error metric is the $\mathbb{L}_2$-norm. The benefit of this integral quantity is that it is normalized over the number of degrees of freedom ($N_X^{\delta x_l}$) used in the approximations, and therefore will be useful in the next chapter when comparing estimators from varying resolutions. The $\mathbb{L}_2$-norm of $\bar{\psi}_{N,t_k}^{(h,\delta x_l)}$, with respect to the reference trajectory, is given by

$$\begin{aligned}&\left\|\bar{\psi}_{N,t_k}^{(h,\delta x_l)}(x,y) - \psi_{t_k}^{(h,\delta x_l),\text{ref}}(x,y)\right\|_{\mathbb{L}_2} \\ &= \sqrt{\int_0^1 \int_0^1 \left(\bar{\psi}_{N,t_k}^{(h,\delta x_l)}(x,y) - \psi_{t_k}^{(h,\delta x_l),\text{ref}}(x,y)\right)^2 dxdy}.\end{aligned} \tag{5.27}$$

Figure 27 shows this error stabilising from the time when assimilation begins, at $t = 30$.

5.6.3 Probability integral transform histogram

As an extension to the data assimilation problems considered in the previous two experiments, this section draws on evaluating the calibration of an ensemble forecast using the probability integral transform (PIT) histogram introduced in Chapter 3. A PIT histogram can be generated from the data assimilation simulation in Sec. 5.6.1. More specifically the observations used to construct the filtering distribution can now be used to evaluate the calibration of the forecast ensemble at every assimilation step. To do this, the states of the forecast ensemble of streamfunctions $\left\{\psi_{t_k}^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^{N}$ at each assimilation step $k = 1, ..., 140$, are evaluated at (0.3,0.3). These evaluations form a scalar ensemble forecast at each assimilation step. One would like to evaluate the calibration of these using a PIT histogram. The observations used in the evaluation of the forecast ensemble CDF at every assimilation step, are not perturbed as they are in (5.24), instead they are equal to the reference trajectory evaluated at (0.3,0.3). Note that this is because the observations in Chapter 3 are unperturbed.

Recall from Chapter 3 that an empirical CDF for $\left\{\psi_{t_k}^{(h,\delta x_l),i}(0.3,0.3)\right\}_{i=1}^{N}$ can be defined as

$$F_{\pi_{t_k}}^{N}(x) = \frac{1}{N}\sum_{i=1}^{N} H\left(x - \psi_{t_k}^{(h,\delta x_l),i}(0.3,0.3)\right),$$

where H is the Heaviside function, and π_{t_k} is the distribution associated with the random variable $\psi_{t_k}^{(h,\delta x_l)}(0.3,0.3)$. One can then use the reference trajectory observations, at every assimilation step $k = 1, ..., 140$, alongside (3.2) to generate a frequency histogram of the PIT samples. The resultant histogram is shown in Figure 28. It shows approximately uniform frequency suggesting a calibrated forecast ensemble, at this spatial coordinate, throughout the filtering. Indeed, this further supports the performance of the spatio-temporal ETPF framework proposed in this chapter.

5.7 Conclusions

This chapter has applied the ensemble transform particle filter (ETPF), to finite element discretizations of random functions, via a projection-based localisation technique. The localisation technique is a variant of the one proposed in Cheng and Reich [2013] for the ETPF. This allows data assimilation to be carried out on spatio-temporal systems with random components, using observations taken from a reference trajectory evaluated at coordinates on the

domain. This technique is utilised in both the importance weight updates and the ensemble transform stages of the ETPF algorithm in Sec. 2.7.4.

A stochastic version of the quasi-geostrophic equations [Holm, 2015] is used to demonstrate the effectiveness of this localisation technique for the ETPF. This was done in a standard (from a known state) and a delayed assimilation setting. An effective proof of concept has been presented, with the ETPF approximation to the filtering distribution tracking a reference trajectory of both the potential vorticity and the associated streamfunction. The variation in the forecast system, without data assimilation, appears relatively large, even when only using four Sine modes in the truncated KL expansion forming the random forcing function in the system.

For the simulations carried out in this chapter, the computational cost of the algorithm appears to be dominated by the forward prediction model rather than that of solving the optimal transport problems in the assimilation steps. Therefore the multilevel ETPF, proposed in Chapter 4, has the potential to be beneficial at reducing the overall computational cost of generating these spatio-temporal filtering estimators, even if only for a range of accuracies. This is considered in the following chapter.

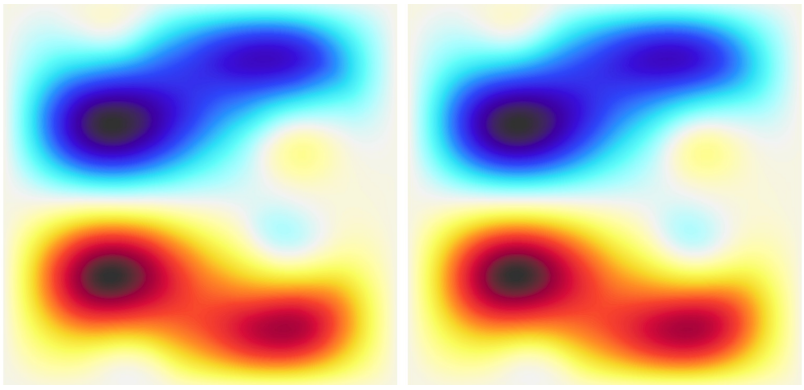
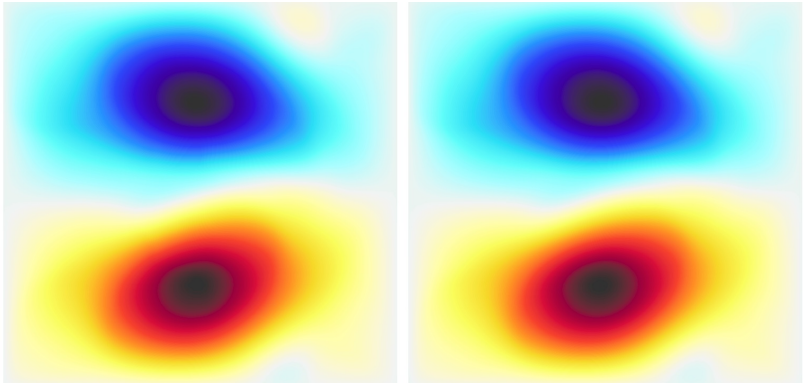


Figure 22: The state of the streamfunction estimator, $\bar{\psi}^{(h,\delta x_l)}_{N,42.5}(x,y)$ (left panel), and the reference trajectory, $\psi^{(h,\delta x_l),\mathrm{ref}}_{42.5}(x,y)$ (right panel), at time 42.5.



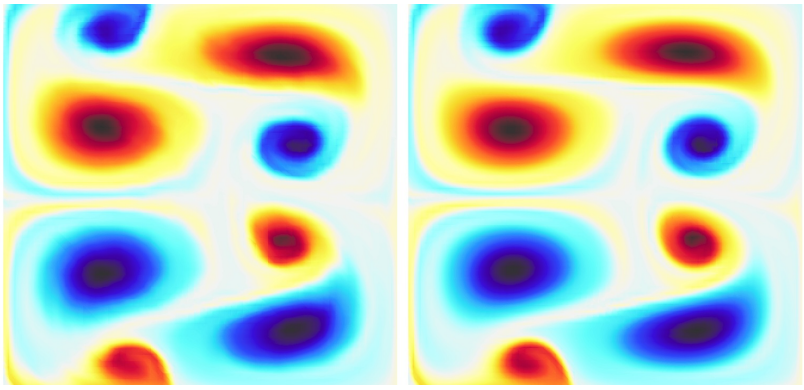
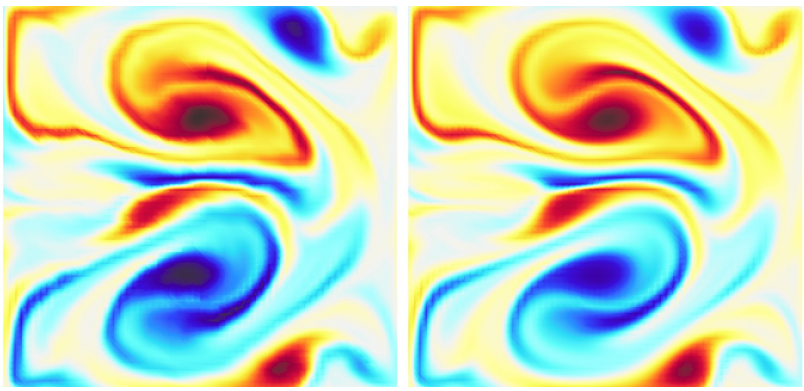


Figure 24: The state of the potential vorticity estimator, $\bar{q}^{(h,\delta x_l)}_{N,42.5}(x,y)$ (left panel), and the reference trajectory, $q^{(h,\delta x_l),\text{ref}}_{42.5}(x,y)$ (right panel), at time 42.5.



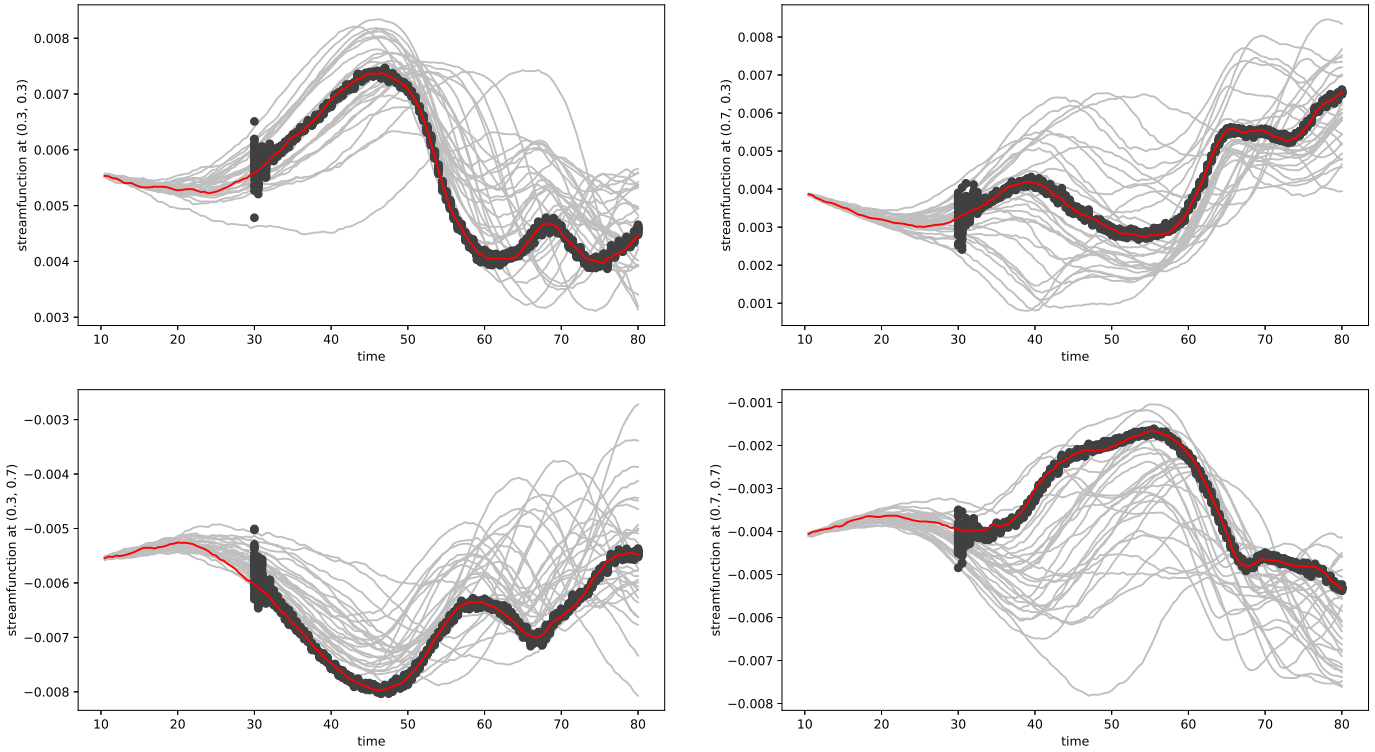


Figure 26: The same as Figure 21, only for the scenario where assimilation is delayed by 20 time units. Once again, from the time that data assimilation starts at, the variance of the analysis ensemble members around the reference trajectory is significantly less than that of N random trajectories of the system without data assimilation, shown by the light grey lines.

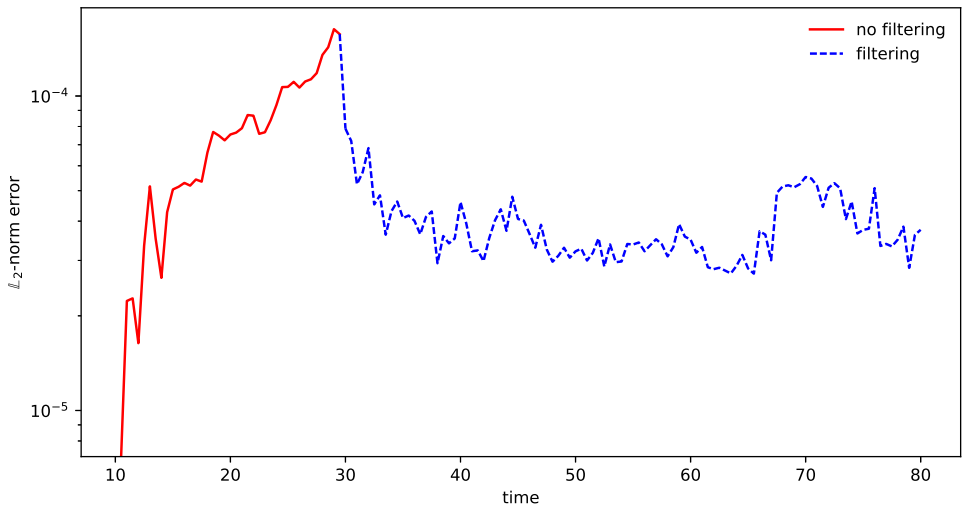


Figure 27: The $\mathbb{L}_2$-norm error of the ETPF approximation to the analysis mean of the filtered streamfunction, with respect to the reference trajectory, in the case where assimilation is delayed by 20 time units. The error during this delay period increases exponentially until assimilation starts, where it then begins to stabilise.

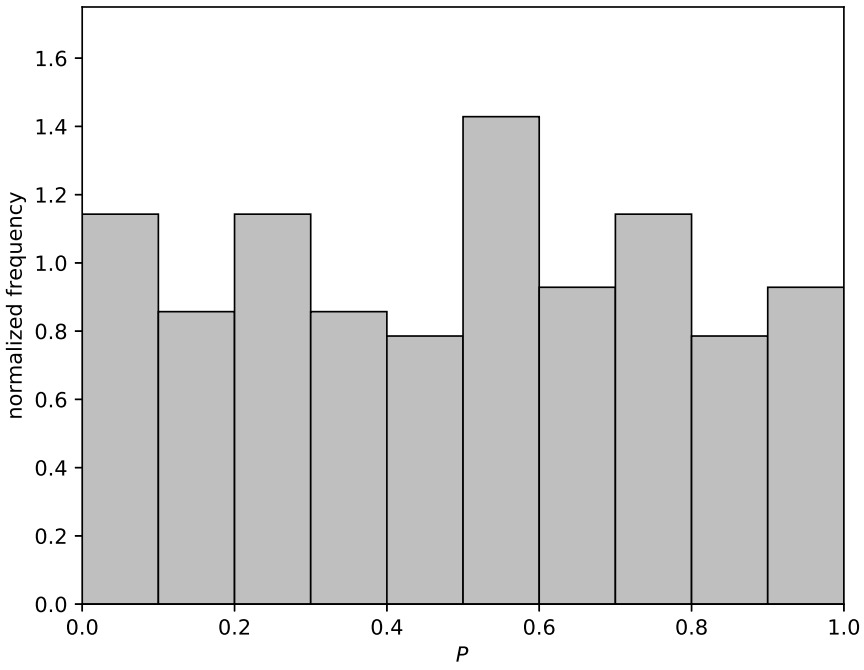


Figure 28: Probability integral transform histogram (with normalized frequency) of the forecast ensemble $\left\{\psi_{t_k}^{(h,\delta x_l),i}(0.3,0.3)\right\}_{i=1}^{N}$ used for the ETPF approximation to the analysis mean of the filtered streamfunction. Observations are given at every assimilation step by the reference trajectory evaluated at $(0.3,0.3)$. The histogram is approximately uniform suggesting that the forecast from the ETPF is calibrated with the reference trajectory.

Chapter 6

Multilevel spatio-temporal ensemble transform methods

Chapter synopsis

This chapter proposes a multilevel extension to the framework presented in the previous chapter, applying the ensemble transform particle filter (ETPF) to systems of partial differential equations. The high computational cost associated with propagating realisations of random fields within the ETPF (apparent in the previous chapter) is the main motivation for this application. The efficiency of the multilevel ETPF (MLETPF) presented in Chapter 4,

relative to its standard counterpart, can be beneficial here by alleviating this high computational cost.

6.1 A multilevel ensemble transform particle filter estimator

In the previous chapter, the extension of the ETPF to spatio-temporal systems primarily concerned the estimator for $\mathbb{E}[\tilde{\phi}_{t_k}^{(h,\delta x_L)}(x)]$, where $\tilde{\phi}_{t_k}^{(h,\delta x_L)}(x) \in V^{\delta x_L}$ is a finite element approximation of the function $\phi_{t_k}(x)$. It is conditioned on partial observations of the form (5.5) from all assimilation steps up to and including time t_k. Unlike in the previous chapter, the time-step h used for each approximation will now also be refined over l (in addition to the spatial resolution), as was the case in Chapter 4. Denote the new approximation by $\tilde{\phi}_{t_k}^{(h_L,\delta x_L)}(x)$. Here L denotes a sufficiently high level of temporal and spatial resolution. In practice the time-step for these approximations can be adaptively set given δx_l, typically based on stability criteria or Courant numbers, but for the scope of this chapter, the assumption that $h_l \propto \delta x_l \propto m^{-l}$, with $m > 1$, is made.

The estimator for $\mathbb{E}[\tilde{\phi}_{t_k}^{(h_L,\delta x_L)}(x)]$ in the previous chapter was given by

$$\bar{\phi}_{N,t_k}^{(h_L,\delta x_L)}(x) = \frac{1}{N}\sum_{i=1}^{N} \tilde{\phi}_{t_k}^{(h_L,\delta x_L),i}(x), \tag{6.1}$$

where the so-called analysis ensemble $\left\{\tilde{\phi}_{t_k}^{(h_L,\delta x_L),i}(x)\right\}_{i=1}^{N}$ was generated by transforming a forecast ensemble of function approximations $\left\{\phi_{t_k}^{(h_L,\delta x_L),i}(x)\right\}_{i=1}^{N}$, associated with importance weight functions $\left\{\tilde{w}_{t_k}^{(h_L,\delta x_L),i}(x)\right\}_{i=1}^{N}$ given by (5.7). The hierarchy of meshes and spaces in (5.2) and (5.3) that these function approximations exist on is now used not only in the localisation scheme proposed in the previous chapter, but also to construct a multilevel version of (6.1). The injection and prolongation operators are also used in this respect. In the style of Chapter 4, one can write $\mathbb{E}[\tilde{\phi}_{t_k}^{(h_L,\delta x_L)}(x)]$ as

$$\mathbb{E}[\tilde{\phi}_{t_k}^{(h_L,\delta x_L)}(x)] = \mathbb{E}[\tilde{\phi}_{t_k}^{(h_{L^0},\delta x_{L^0})}(x)] + \sum_{l=L^0+1}^{L} \mathbb{E}[\tilde{\phi}_{t_k}^{(h_l,\delta x_l)}(x)] - \mathbb{E}[\tilde{\phi}_{t_k}^{(h_{l-1},\delta x_{l-1})}(x)], \tag{6.2}$$

where $0 \leq L^0 < L$ and therefore an alternative, multilevel estimator for (6.2) can be given

by

$$\bar{\phi}^{(h_L,\delta x_L)}_{(N_{L^0},\ldots,N_L),t_k}(x) = \sum_{l=L^0}^{L} \mu_l. \tag{6.3}$$

The difference between this spatio-temporal multilevel estimator and the one given in (4.3) lies in the difference estimators,

$$\mu_l = \begin{cases} \frac{1}{N_{L^0}} \sum_{i=1}^{N_{L^0}} \mathbb{I}^L_{L^0} \tilde{\phi}^{(h_{L^0},\delta x_{L^0}),i}_{t_k}(x), & l = L^0 \\ \frac{1}{N_l} \sum_{i=1}^{N_l} \mathbb{I}^L_l \tilde{\phi}^{(h_l,\delta x_l),i}_{t_k}(x) - \mathbb{I}^L_{l-1} \tilde{\hat{\phi}}^{(h_{l-1},\delta x_{l-1}),i}_{t_k}(x), & l > L^0. \end{cases} \tag{6.4}$$

The multilevel estimator starts at level L^0 of resolution so that one is still able to use localisation from the hierarchy of meshes in (5.2) for the approximation on the coarsest level of resolution. In (6.4), one is required to prolong the analysis ensembles to the finest function space, $V^{\delta x_L}$. This allows one to gain a statistical estimator comparable to a standard ETPF estimator using samples $\left\{\tilde{\phi}^{(h_L,\delta x_L),i}_{t_k}(x)\right\}_{i=1}^{N}$ all on the finest function space.

The analysis ensembles $\left\{\tilde{\phi}^{(h_l,\delta x_l),i}_{t_k}(x)\right\}_{i=1}^{N_l}$ and $\left\{\tilde{\hat{\phi}}^{(h_{l-1},\delta x_{l-1}),i}_{t_k}(x)\right\}_{i=1}^{N_l}$ for $l > L^0$, where $\left\{\tilde{\hat{\phi}}^{(h_{l-1},\delta x_{l-1}),i}_{t_k}(x)\right\}_{i=1}^{N_l} \sim \tilde{\phi}^{(h_{l-1},\delta x_{l-1})}_{t_k}(x)$, have been transformed in some manner from forecast ensembles $\left\{\phi^{(h_l,\delta x_l),i}_{t_k}(x)\right\}_{i=1}^{N_l}$ and $\left\{\hat{\phi}^{(h_{l-1},\delta x_{l-1}),i}_{t_k}(x)\right\}_{i=1}^{N_l}$ with importance weight functions $\left\{\tilde{w}^{(h_l,\delta x_l),i}_{t_k}(x)\right\}_{i=1}^{N_l}$ and $\left\{\tilde{\hat{w}}^{(h_{l-1},\delta x_{l-1}),i}_{t_k}(x)\right\}_{i=1}^{N_l}$. The importance weight functions for each forecast ensemble are computed following the procedure in Sec. 5.4.2. For the transformation used in μ_{L^0}, one can follow the same methodology that was presented in Sec. 5.4.2. The form of the transformations used in μ_l, for $l = L^0 + 1, ..., L$, will follow the seamless scheme presented in Sec. 4.2.2, and is further explained in the next section. Again, as utilised in Chapter 4, the same random components of the spatio-temporal system are required to be used between assimilation steps for each pair of samples $\left\{\phi^{(h_l,\delta x_l),i}_{t_k}(x)\right\}_{i=1}^{N_l}$ and $\left\{\hat{\phi}^{(h_{l-1},\delta x_{l-1}),i}_{t_k}(x)\right\}_{i=1}^{N_l}$.

The approximations on each level of resolution, l, will use a localisation parameter, $r_{loc,l}$, in the computation of the importance weights within (5.7). Note that the constraint $r_{loc,l} \in [0, l]$ from the previous chapter still holds. In order for (6.3) to be consistent, the parameter $r_{loc,l}$ used within the computation of μ_l has to be the same as the parameter $r_{loc,l}$ used within the computation of μ_{l+1}. A suitable choice is that $r_{loc,l} = \mathcal{O}(1)$ and doesn't depend on l, i.e. the physical area of localisation will decrease as l increases.

6.2 A seamless coupling for spatio-temporal systems

The aim of this section is to describe an extension of Sec. 4.2.2 to the spatio-temporal case considered in this chapter, to use in combination with the MLETPF estimator in (6.3). This replaces the forecast ensembles of functions $\left\{\phi_{t_k}^{(h_l,\delta x_l),i}(x)\right\}_{i=1}^{N_l}$ and $\left\{\hat{\phi}_{t_k}^{(h_{l-1},\delta x_{l-1}),i}(x)\right\}_{i=1}^{N_l}$ alongside importance weight functions $\left\{\tilde{w}_{t_k}^{(h_l,\delta x_l),i}(x)\right\}_{i=1}^{N_l}$ and $\left\{\tilde{\hat{w}}_{t_k}^{(h_{l-1},\delta x_{l-1}),i}(x)\right\}_{i=1}^{N_l}$ with the evenly weighted analysis ensembles used in (6.4). However, it does this whilst preserving the coupling between them, such that

$$\mathbb{V}_l = \begin{cases} \int \mathbb{V}\left[\mathbb{I}_{L^0}^{L}\tilde{\phi}_{t_k}^{(h_{L^0},\delta x_{L^0})}(x)\right] dx, & l = L^0 \\ \int \mathbb{V}\left[\mathbb{I}_{l}^{L}\tilde{\phi}_{t_k}^{(h_l,\delta x_l)}(x) - \mathbb{I}_{l-1}^{L}\tilde{\phi}_{t_k}^{(h_{l-1},\delta x_{l-1})}(x)\right] dx, & l > L^0, \end{cases} \tag{6.5}$$

decays at an asymptotic rate for increasing l, i.e. $\mathbb{V}_l = \mathcal{O}(\delta x_l^{\beta}) = \mathcal{O}(h_l^{\beta})$. For $l > L^0$, the three steps of the seamless coupling algorithm in Sec. 4.2.2 need to be modified for the spatio-temporal case as follows:

(1) The first step transforms the forecast ensemble $\left\{\hat{\phi}_{t_k}^{(h_{l-1},\delta x_{l-1}),i}(x)\right\}_{i=1}^{N_l}$ alongside associated importance weight functions $\left\{\tilde{\hat{w}}_{t_k}^{(h_{l-1},\delta x_{l-1}),i}(x)\right\}_{i=1}^{N_l}$, into an intermediate ensemble of functions $\left\{\phi_{t_k}^{*,i}(x)\right\}_{i=1}^{N_l}$, on the space $V^{\delta x_{l-1}}$, alongside importance weight functions $\left\{\mathbb{I}_l^{l-1}\tilde{w}_{t_k}^{(h_l,\delta x_l),i}(x)\right\}_{i=1}^{N_l}$. Note these weight functions are the injected importance weight functions on the finer resolution level.[1] Let the $n = 1, ..., N_X^{\delta x_{l-1}}$ basis coefficients of these intermediate ensemble members be $\left\{a_n^{*,j}\right\}_{j=1}^{N_l}$. They can be found by using the coupling matrices $D^n \in \mathcal{D}^n \in \mathbb{R}^{N_l \times N_l}$, for $n = 1, ..., N_X^{\delta x_{l-1}}$, alongside

$$a_n^{*,j} = \sum_{i=1}^{N_l} D_{i,j}^n (b_n^{\delta x_{l-1},j})^{-1} \hat{a}_n^{\delta x_{l-1},i},$$

where $\hat{a}_n^{\delta x_{l-1},i}$ is the n'th basis coefficient of $\hat{\phi}_{t_k}^{(h_{l-1},\delta x_{l-1}),i}(x)$ and $b_n^{\delta x_{l-1},j}$ is the n'th basis coefficient of $\mathbb{I}_l^{l-1}\tilde{w}_{t_k}^{(h_l,\delta x_l),j}(x)$ respectively for $i, j = 1, ..., N_l$. The matrices are required to satisfy the marginal constraints

$$\sum_{i=1}^{N} D_{i,j}^n = b_n^{\delta x_{l-1},j}, \qquad \sum_{j=1}^{N} D_{i,j}^n = \hat{b}_n^{\delta x_{l-1},i},$$

[1] In practice, one might be required to renormalize these importance weight functions once injected such that $\sum_{i=1}^{N_l} \mathbb{I}_l^{l-1}\tilde{w}_{t_k}^{(h_l,\delta x_l),i}(x) = 1, \forall x \in \Gamma$.

where $\hat{b}_n^{\delta x_{l-1},i}$ is the n'th basis coefficient of $\tilde{\hat{w}}_{t_k}^{(h_{l-1},\delta x_{l-1}),i}(x)$ for $i = 1, ..., N_l$. The matrices are then found by solving the optimal transport problems,

$$\arg \min_{D^n \in \mathcal{D}^n} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} D_{i,j}^n e_n^{(i,j)}.$$

Here, $e_n^{(i,j)}$ is defined by the n'th basis coefficient of the function $J^{(i,j)}(x) \in V^{\delta x_{l-1}}$, given by

$$J^{(i,j)}(x) = \mathcal{I}_{\text{smooth}} \mathcal{I}_{(l-1,r_{loc,l-1})} \left\{ \left(\hat{\phi}_{t_k}^{(h_{l-1},\delta x_{l-1}),i}(x) - \mathbb{I}_l^{l-1} \phi_{t_k}^{(h_l,\delta x_l),j}(x) \right)^2 \right\}. \quad (6.6)$$

Note that the localisation parameter $r_{loc,l-1} \in [0, l-1]$ is used in this step.

(2) The second step can be modified by simply implementing the transform of the forecast ensemble $\left\{ \phi_{t_k}^{(h_l,\delta x_l),i}(x) \right\}_{i=1}^{N_l}$ with the importance weight functions $\left\{ \tilde{w}_{t_k}^{(h_l,\delta x_l),i}(x) \right\}_{i=1}^{N_l}$, to the evenly weighted analysis ensemble $\left\{ \tilde{\phi}_{t_k}^{(h_l,\delta x_l),i}(x) \right\}_{i=1}^{N_l}$, explained in Sec. 5.4.2. The localisation parameter $r_{loc,l} \in [0, l]$ is used in this step.

(3) The last step transforms the intermediate ensemble of functions $\left\{ \phi_{t_k}^{*,i}(x) \right\}_{i=1}^{N_l}$ with the importance weight functions $\left\{ \mathbb{I}_l^{l-1} \tilde{w}_{t_k}^{(h_l,\delta x_l),i}(x) \right\}_{i=1}^{N_l}$, to an evenly weighted analysis ensemble $\left\{ \tilde{\hat{\phi}}_{t_k}^{(h_{l-1},\delta x_{l-1}),i}(x) \right\}_{i=1}^{N_l}$. Let $\tilde{\hat{a}}_n^{\delta x_{l-1},i}$ and $\tilde{a}_n^{\delta x_{l-1},i}$ be the basis coefficients of $\tilde{\hat{\phi}}_{t_k}^{(h_{l-1},\delta x_{l-1}),i}(x)$ and $\mathbb{I}_l^{l-1} \tilde{\phi}_{t_k}^{(h_l,\delta x_l),i}(x)$ (injections of the analysis ensemble members on the finer resolution level) respectively, for $i = 1, ..., N_l$. The former of these can be found by using the coupling matrices $T^n \in \mathcal{T}^n \in \mathbb{R}^{N_l \times N_l}$, for $n = 1, ..., N_X^{\delta x_{l-1}}$, alongside

$$\tilde{\hat{a}}_n^{\delta x_{l-1},j} = \sum_{i=1}^{N_l} N T_{i,j}^n a_n^{*,i},$$

for $j = 1, ..., N_l$. The matrices are required to satisfy the marginal constraints

$$\sum_{i=1}^{N} T_{i,j}^n = \frac{1}{N_l}, \qquad \sum_{j=1}^{N} T_{i,j}^n = b_n^{\delta x_{l-1},i},$$

where $\left\{ b_n^{\delta x_{l-1},i} \right\}_{i=1}^{N_l}$ are as in step (1) and are then found by solving the optimal trans-

port problems,

$$\arg \min_{T^n \in \mathcal{T}^n} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} T^n_{i,j} e^{(i,j)}_n.$$

Here, $e^{(i,j)}_n$ is defined by the n'th basis coefficient of the function $J^{(i,j)}(x) \in V^{\delta x_{l-1}}$:

$$J^{(i,j)}(x) = \mathcal{I}_{\text{smooth}} \mathcal{I}_{(l-1, r_{loc,l-1})} \left\{ \left(\phi^{*,i}_{t_k}(x) - \mathbb{I}^{l-1}_l \tilde{\phi}^{(h_l, \delta x_l), j}_{t_k}(x) \right)^2 \right\}. \tag{6.7}$$

Note that the localisation parameter $r_{loc,l-1} \in [0, l-1]$ is also used in this step.

The estimator in (6.3), used in combination with the seamless coupling scheme above, will be henceforth known as the spatio-temporal MLETPF estimator. As a demonstration of the modified transformation/coupling scheme presented above, consider the following 'coarse' and 'fine' forecast ensembles (ignore temporal resolution here):

$$\hat{\phi}^{(\delta x_0), i}(x) \in V^{P0, \delta x_0}, \quad \phi^{(\delta x_1), i}(x) \in V^{P0, \delta x_1}, \qquad i = 1, ..., 50,$$

Here, each ensemble member is a piecewise constant function, and is an interpolation of $\eta \sin(2\pi x)$, with $\eta \sim N(0, 0.25)$, on the unit interval mesh, with grid cell lengths $\delta x_1 = 0.1$ and $\delta x_0 = 0.2$. A reference function is sampled on the finer mesh, and 10 observations are taken from this at coordinates

$$x^j_{\text{coord}} = (j-1)/9, \qquad j = 1, ..., 10,$$

using (5.5) and $R = 0.1$. Using the parameters $r_{loc,0} = r_{loc,1} = 0$, the weight functions for the coarse and fine forecast ensembles are constructed via the methodology in Sec. 5.4.2. Then both forecast ensembles with associated weight functions are transformed into analysis ensembles,

$$\tilde{\hat{\phi}}^{(\delta x_0), i}(x) \in V^{P0, \delta x_0}, \quad \tilde{\phi}^{(\delta x_1), i}(x) \in V^{P0, \delta x_1}, \qquad i = 1, ..., 50,$$

via the transformation/coupling scheme presented earlier in this section. Both of these ensembles and the forecast ensembles, evaluated at $x = 0.5$, are shown in Figure 29. One notes that the analysis ensembles are positively correlated and coupled.

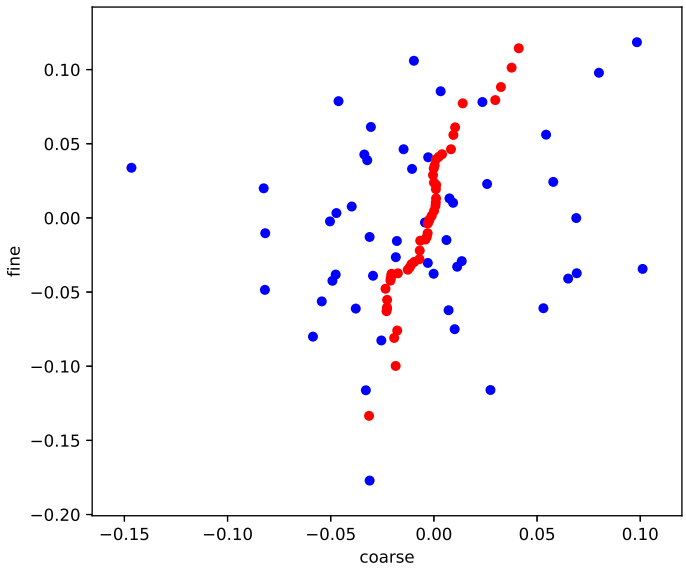


Figure 29: The coarse and fine forecast ensembles $\left\{\hat{\phi}^{(\delta x_0),i}(x)\right\}_{i=1}^{50}$ and $\left\{\phi^{(\delta x_1),i}(x)\right\}_{i=1}^{50}$, and their respective analysis ensembles constructed from the coupling/transformation scheme in Sec. 6.2. The analysis ensembles are positively correlated due to the optimal transport problems within the scheme.

6.3 Numerical demonstrations

The stochastic quasi-geostrophic equations are now used to present benchmark results for the MLETPF applied to spatio-temporal cases. These equations were introduced in the previous chapter. Our aim is to reduce the overall computational cost of the ETPF estimators attained in Sec. 5.6.1 and Sec. 5.6.2. These sections considered the estimation of the analysis mean for the streamfunction, $\mathbb{E}[\tilde{\psi}_{t_k}^{(h_L,\delta x_L)}(x,y)]$, on a sufficiently fine resolution level L, conditioned on partial observations taken from a reference trajectory.

As done in Chapter 4, such increases in efficiency will be demonstrated by constructing MLETPF and ETPF estimators with pre-defined orders of accuracy. The order of accuracy, $\mathcal{O}(\epsilon^2)$, will dictate the parameters L and N_l, for $l = L^0, ..., L$, used within the MLETPF estimator as well as N and L used within the standard ETPF estimator. In order to evaluate the accuracy, an error metric needs to be defined for the spatio-temporal case. This error metric is defined with respect to an accurate ETPF approximation of $\mathbb{E}[\tilde{\psi}_{t_k}(x,y)]$, the true analysis mean of the streamfunction. In practice this approximation of $\mathbb{E}[\tilde{\psi}_{t_k}(x,y)]$ uses one level of spatial resolution greater than any used in the experiments, and is taken over

four times as many samples as used in any of the experiments. The time-averaged squared $\mathbb{L}_2$-norm for the MLETPF estimator is then given by

$$\begin{aligned}&\frac{1}{N_t}\sum_{k=1}^{N_t}\left\|\bar{\psi}^{(h_L,\delta x_L)}_{(N_{L^0},\ldots,N_L),t_k}(x,y)-\mathbb{E}[\tilde{\psi}_{t_k}(x,y)]\right\|^2_{\mathbb{L}_2}\\&=\frac{1}{N_t}\sum_{k=1}^{N_t}\int_0^1\int_0^1\left(\bar{\psi}^{(h_L,\delta x_L)}_{(N_{L^0},\ldots,N_L),t_k}(x,y)-\mathbb{E}[\tilde{\psi}_{t_k}(x,y)]\right)^2 dxdy,\end{aligned} \tag{6.8}$$

where N_t is the number of assimilation steps. This will compare the MLETPF and ETPF estimators, both localised in a particular manner, with respect to their convergence to a localised, higher discretization accuracy approximation, as done with the stochastic Lorenz 96 equations in Chapter 4. With this in mind, the standard ETPF estimator will use the same localisation parameter as the approximation at the finest resolution in the MLETPF approximations, $r_{loc,L}$. In addition to this, the ETPF approximation used as a proxy for $\mathbb{E}[\tilde{\psi}_{t_k}(x,y)]$ in the error metric above utilises a localisation parameter of $(r_{loc,L}+n_l)$. Here n_l is the difference between L and the level of resolution that the approximation is on. This is so that the approximation uses the same physical area of localisation as the standard ETPF estimator and the finest approximation in the MLETPF estimators. Recall that in the previous chapter, the forward model cost for this system was superior to that of solving the optimal transport problems in the ETPF assimilation stages for a certain order of accuracy. Therefore one expects that overall computational cost reductions can be obtained, at least for a regime of ϵ.

As stated earlier, the assumption that $h_l \propto \delta x_l$ is made throughout this chapter. For the case of the quasi-geostrophic equations, the analysis in Cockburn and Shu [2001] allows one to adaptively set h_l given δx_l, a Courant constant and the velocity of the system. For the simplicity of the framework proposed in this chapter, a constant time-step is preferred. The time-step hierarchy used in the following simulations will be given by $h_l = 0.4\delta x_l$. This is based on the Courant constant associated with the linear degree of the function approximations and the type of Runge-Kutta scheme used. It also assumes an $\mathcal{O}(1)$ velocity [Cockburn and Shu, 2001]. The spatial resolution follows the same hierarchy as it did in the previous chapter, i.e. $\delta x_l = 2^{-l-1}$.

6.3.1 Coupling performance

As was the case in Chapter 4, the sample statistics, μ_l and $\mathbb{V}_l$, can be computed for a simulation of the MLETPF for the filtering of the stochastic quasi-geostrophic equations. For this particular case, the statistics μ_l and $\mathbb{V}_l$ are given by (6.4) and (6.5) respectively with $\phi(x) = \psi(x, y)$. In particular, the asymptotic decay of $\mathbb{V}_l$, as l increases, can demonstrate the effectiveness of the coupling scheme in Sec. 6.2.

This simulation uses most of the major parameter values that were used in the assimilation experiment in Sec. 5.6.1. The time interval over which the assimilation takes place is kept as $t = [10, 80]$, and the assimilation frequency is set to $\Delta t = 0.5$.[2] For this particular task of computing sample statistics, it suffices to use a fixed number of samples $N_l = 70$ to compute each of the difference estimators, μ_l, for $l = L^0, ..., L$, in (6.4). Here, the finest level of the mesh hierarchy is assumed to be $L = 7$. The minimum level of spatial / temporal resolution used in the MLETPF estimator is $L^0 = 3$. This is to allow the coarsest simulation to still utilise localisation, i.e. $r_{loc,L^0} \in [0, 3]$.

The average integrated (over the domain $\Gamma \in [0, 1]^2$) values of $|\mu_l|$ and values of $\mathbb{V}_l$, for $l = L^0, ..., L$, over all assimilation steps are shown in Figure 30. They both decay at a rate of $\mathcal{O}(\delta x_l^2) = \mathcal{O}(h_l^2)$. This corresponds to $\alpha = \beta = 2$ and therefore second order weak convergence, as supposed to the first order weak convergence that was examined in temporally discretized problems throughout Chapter 4. A similar rate of variance decay for quantities of interest in two-dimensional systems of partial differential equations is found in Cliffe et al. [2011] for a standard multilevel Monte Carlo estimator (without assimilation). These rates will be important in the following sections.

6.3.2 Computational cost considerations

It was straightforward to pre-define an order of magnitude for the sample sizes within the MLETPF in Chapter 4, given that the rate of computational cost growth associated with the increase in resolution, γ, was known. Here, the value of γ will depend on both the spatial and temporal resolution of the problem. The runtime of simulating the stochastic quasi-geostrophic equations over the time interval $t \in [0, 2]$ is shown in Figure 31. This simulation uses the implementation with temporal and spatial resolutions both being refined together,

[2] As utilised in the previous chapter, a deterministic system is used up until time $t = 10$, with all filtering simulations starting from this point in time.

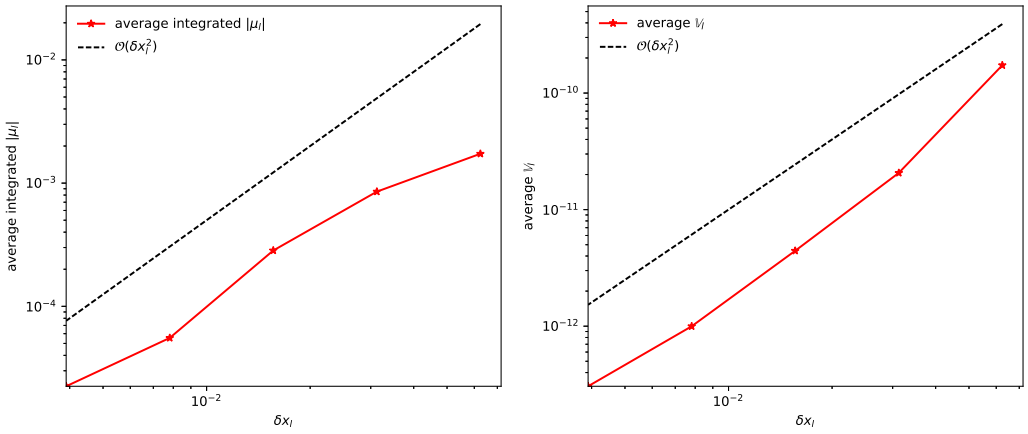


Figure 30: Average values of $\int_0^1 \int_0^1 |\mu_l| dxdy$ and $\mathbb{V}_l$, for $l \in [L^0, 7]$, over all assimilation steps $k \in [1, 140]$ for the MLETPF approximation to the analysis mean of the filtered streamfunction. Following the asymptotes shown by the dashed black lines, both of these quantities decay at a rate of $\mathcal{O}(\delta x_l^2)$.

explained in Sec. 6.3, and the figure suggests a value of $\gamma = 4$.

This runtime can be decomposed into the time-stepping of (5.15) and solving the elliptic problem in (5.16). The first part has a computational cost that scales at $\mathcal{O}(h_l^{-1}) = \mathcal{O}(\delta x_l^{-1})$. The second part, which therefore has a computational cost that scales as $\mathcal{O}(\delta x_l^{-3})$, can be further decomposed into the number of iterations used in the elliptic solver from Firedrake, and the time of each iteration. Figure 31 also shows the growth rates, with increasing l, of each of these parts. The runtime of each iteration used scales as $\mathcal{O}(\delta x_l^{-2})$, and therefore the number of iterations scales as $\mathcal{O}(\delta x_l^{-1})$. Whilst parallelisation could decrease the growth rate of the former, multi-grid methods can be employed to make the number of iterations independent of δx_l.

6.3.3 Assimilation from a known state

The experiment from Sec. 5.6.1 will now be repeated with the MLETPF to compare the efficiency of it against that of the ETPF. Recall that in the previous chapter the spatio-temporal forward model cost was superior to that of solving the optimal transport problems in the ETPF algorithm, at least only for a given accuracy. This suggests that one can expect the MLETPF to achieve overall computational cost reductions over the standard ETPF, but only for a regime of accuracy given that $\gamma = 4$, as stated in Sec. 4.4. This size of this regime

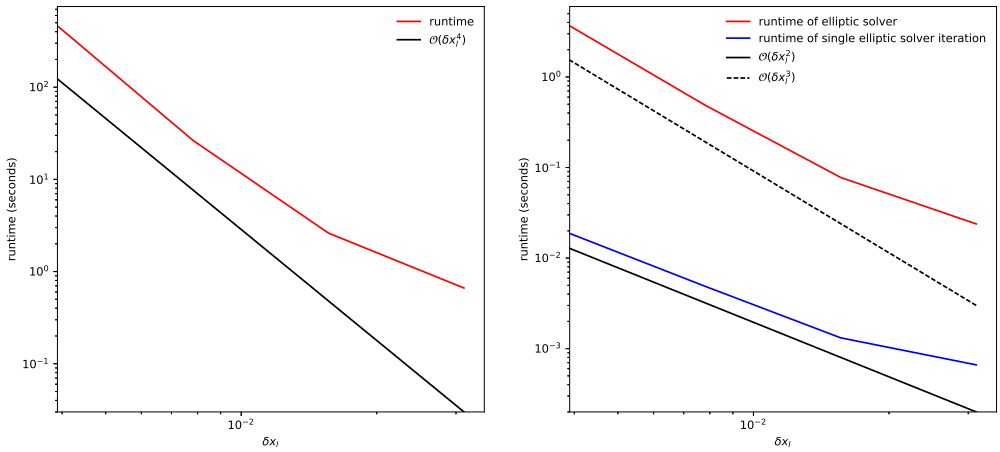


Figure 31: The runtimes (seconds) of simulating the stochastic quasi-geostrophic equations (left panel), solving (5.16) and an individual iteration of solving (5.16) (right panel), on varying levels of temporal and spatial resolution. Following the asymptotes shown by the black lines, the total runtime of the simulation grows at a rate of $\mathcal{O}(\delta x_l^{-4})$, whereas the total and individual iteration runtimes of solving (5.16) grow at rates of $\mathcal{O}(\delta x_l^{-3})$ and $\mathcal{O}(\delta x_l^{-2})$ respectively.

depends on the scalar constants associated with the forward model cost, $\mathcal{O}(\delta x_l^{-4})$, and the cost of solving the optimal transport problems within the scheme outlined in Sec. 6.2.

The assimilation parameters used in the experiment in Sec. 5.6.1 remain the same, and the coarsest resolution used in the MLETPF is set to $L^0 = 3$. The localisation parameters for each level of resolution are set to $r_{loc,l} = L^0$. The reference trajectory, that the observations are taken from in (5.5), is given by a single simulation of the stochastic quasi-geostrophic equations with temporal and spatial resolution h_L and δx_L, for a given ϵ, i.e. $\psi_{t_k}^{(h_L,\delta x_L),\text{ref}}(x,y)$.

Following the generalised version of (2.30) in Cliffe et al. [2011], the sample sizes of each estimator μ_l are set to scale as $N_l = \mathcal{O}(\delta x_l^{\beta}\delta x_l^{\gamma})^{1/2}$. Given that $\alpha = 2$, $\beta = 2$ and $\gamma = 4$, therefore $N_l = 2^{-3}N_{l-1}$. Therefore the coarsest estimator will have many more samples than the estimators on higher levels. This reflects the high rate of growth for the forward model computational cost, as l increases. The sample size on the coarsest level will scale as $N_0 = \mathcal{O}(\epsilon^{-2})$, as in all experiments in Chapter 4. The finest level of resolution used in the MLETPF and standard ETPF estimators scales at $L = \mathcal{O}(-\log(\epsilon^{1/2})/\log(m))$. This is because of the second order weak convergence ($\alpha = 2$), and means that $\delta x_L = \mathcal{O}(\epsilon^{1/2})$ is required.

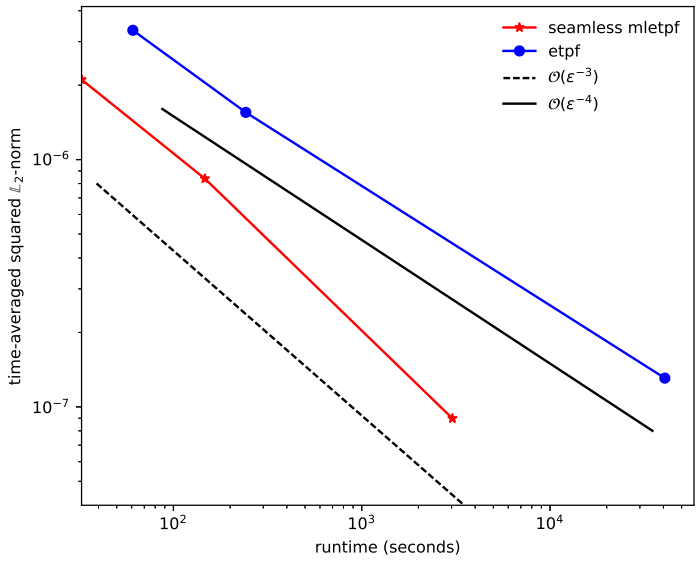


Figure 32: Time-averaged squared $\mathbb{L}_2$-norm against runtime (seconds) of the standard ETPF and MLETPF estimators to $\mathbb{E}[\tilde{\psi}_{t_k}(x, y)]$ for the stochastic quasi-geostrophic equations. Results are shown for decreasing values of ϵ. Following the asymptotes shown by the dashed and solid black lines, the runtimes grow at rates of $\mathcal{O}(\epsilon^{-3})$ and $\mathcal{O}(\epsilon^{-4})$ for the MLETPF and standard ETPF estimators respectively.

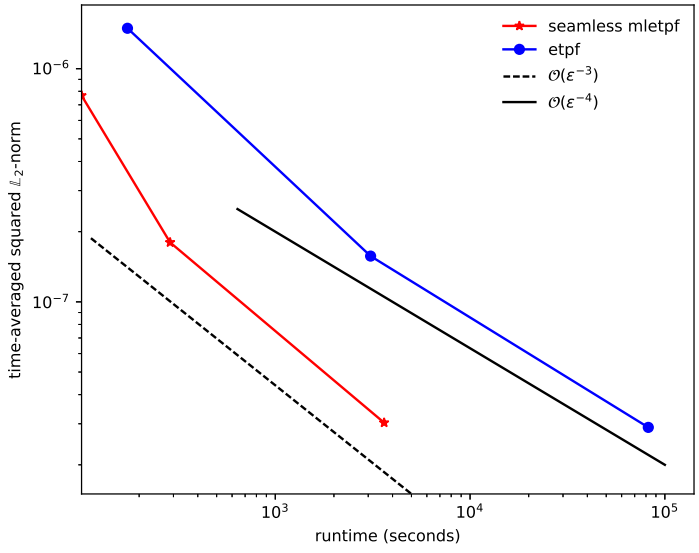


Figure 33: The same as Figure 32 only for the case where assimilation is delayed for 20 time units.

The MLETPF and standard ETPF estimators are constructed for various values of ϵ and their computational cost against time-averaged squared $\mathbb{L}_2$-norms are shown in Figure 32. Given the values of α, β and γ noted above, it is apparent that the rate of computational cost growth for the MLETPF and standard ETPF estimators, $\mathcal{O}(\epsilon^{-2-(\gamma-\beta)/\alpha}) = \mathcal{O}(\epsilon^{-3})$ and $\mathcal{O}(\epsilon^{-2-\gamma/\alpha}) = \mathcal{O}(\epsilon^{-4})$ respectively, are as they are in (2.27). These reductions from the MLETPF suggest that the forward model cost in these simulations is superior to the cost of solving the optimal transport problems within the MLETPF and ETPF estimators, at least for the orders of accuracy considered here. In fact this claim is supported in Figure 34 where the forward model cost and the cost of solving the optimal transport problems in an ETPF simulation over $t \in [10, 12]$ is shown for various values of ϵ. The computational cost of solving the optimal transport problems increases with ϵ at a greater rate than the forward model cost, as one would expect given a value of $\gamma = 4$. Therefore in support of Sec. 4.4, the MLETPF would not offer overall computational cost reductions for all values of ϵ, but the forward model cost appears to be large enough to give a regime of ϵ values for which overall cost reductions can be achieved for. Note that the MLETPF estimator and standard ETPF estimator for the smallest value of ϵ both use a finest mesh $M^{\delta x_L}$, with $L = 6$, which has 128×128 cells. The rates of CAR values for the MLETPF and standard ETPF estimators are then $\mathcal{O}(\epsilon^{-1})$ and $\mathcal{O}(\epsilon^{-2})$ respectively. Therefore the MLETPF delivers overall computational cost reductions, over the standard ETPF, for the assimilation of the stochastic quasi-geostrophic equations.

6.3.4 Delayed assimilation

It is also interesting to compare the MLETPF and standard ETPF estimators for the delayed assimilation case in Sec. 5.6.2. This is when assimilation does not start until $t = 30$, and then proceeds as it did in the last example until the end of the time interval at $t = 80$. All of the parameters used in the previous section for the MLETPF and standard ETPF estimators remain the same. Figure 33 shows the same as Figure 32 only for the delayed assimilation case considered in this section. These results broadly follow those shown in the previous section, with a $\mathcal{O}(\epsilon^{-1})$ speed-up in computational cost being achieved by the MLETPF estimators.

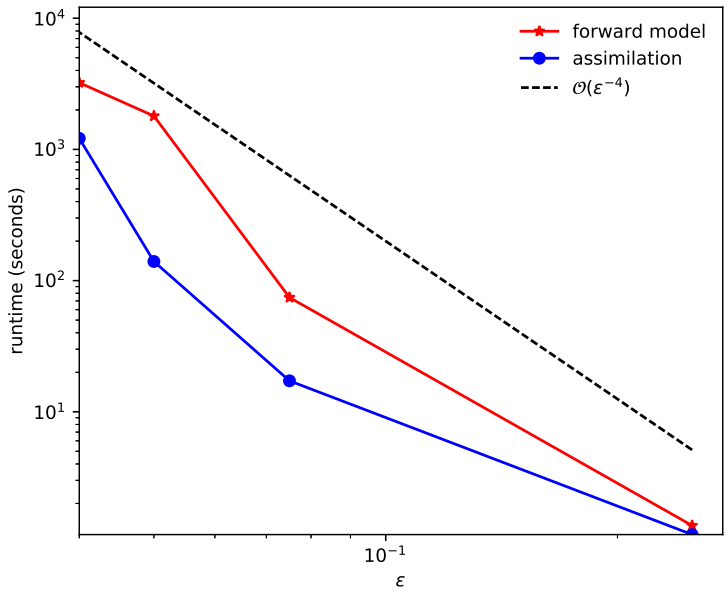


Figure 34: The forward model computational cost, and the cost of solving the optimal transport problems during assimilation in an ETPF simulation for the stochastic quasi-geostrophic equations with various values of ϵ. Following the asymptote shown by the dashed black line, the forward model cost scales as $\mathcal{O}(\delta x_l^{-4})$, however is not growing as fast as the cost associated with solving the optimal transport problems.

6.3.5 Probability integral transform histogram

In Sec. 5.6.3 the calibration of the forecast ensemble of streamfunction approximations $\left\{\psi_{t_k}^{(h,\delta x_l),i}(x,y)\right\}_{i=1}^{N}$ in a data assimilation example was considered. To do this, observations taken from the reference trajectory, $\psi_{t_k}^{(h,\delta x_l),\text{ref}}(x,y)$, evaluated at a coordinate on the domain, were used to generate a probability integral transform histogram via the methodology in Sec. 3.1.1. This is now done for the hierarchy of forecast ensembles, evaluated at (0.3, 0.3),

$$\begin{cases} \left\{\psi_{t_k}^{(h_{L^0},\delta x_{L^0}),i}(0.3,0.3)\right\}_{i=1,\ldots,N_{L^0}}, & l = L^0, \\ \left\{\psi_{t_k}^{(h_l,\delta x_l),i}(0.3,0.3), \hat{\psi}_{t_k}^{(h_{l-1},\delta x_{l-1}),i}(0.3,0.3)\right\}_{i=1,\ldots,N_l}, & l > L^0, \end{cases} \tag{6.9}$$

that were used in the data assimilation example in Sec. 6.3.3. The hierarchy above corresponds to that produced in the MLETPF simulation associated with the smallest ϵ value used in Sec. 6.3.3. At each assimilation step, a single ensemble forecast of size N_0 is generated from the hierarchy above via the methodology in Sec. 3.2, and the probability integral trans-

form histogram is then constructed in the same way as in Sec. 5.6.3 using this ensemble forecast. The histogram is shown in Figure 35. This approximately uniform histogram, very much like the one shown in Sec. 5.6.3, suggests that the MLETPF approximation to the forecast distribution at each assimilation step is calibrated with the reference trajectory, at the spatial coordinate considered.

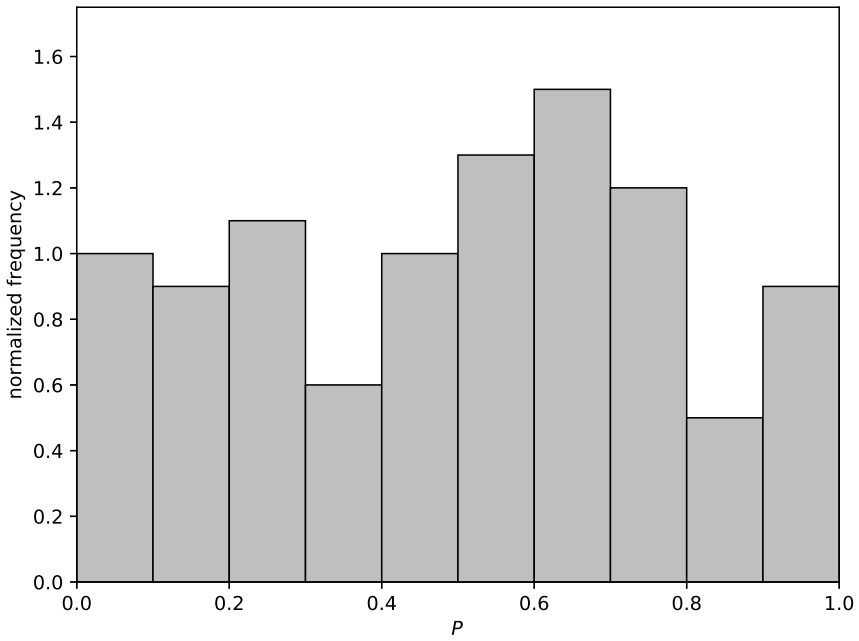


Figure 35: Probability integral transform histogram (with normalized frequency) of the multilevel ensemble forecast generated from the hierarchy of forecast ensembles in (6.9), used for the MLETPF approximation to the analysis mean of the filtered streamfunction. Observations are given at every assimilation step by the reference trajectory evaluated at $(0.3, 0.3)$. The histogram is approximately uniform suggesting that the forecast from the MLETPF is calibrated with the reference trajectory.

6.4 Conclusions

This chapter has extended the multilevel ensemble transform particle filtering (MLETPF) framework, and the seamless transformation / coupling scheme in Sec. 4.2.2, to spatio-temporal cases. More specifically, these cases consider filtering partial observations into forecasts of finite element discretizations of random functions. The observations come from

evaluating a reference trajectory at coordinates on the domain. A proof of concept is presented with the stochastic quasi-geostrophic equations, introduced in the previous chapter.

This extension of the MLETPF creates an estimator that is found to be significantly more efficient than the standard ETPF estimators for this problem, proposed in the previous chapter. This overall speed-up is a result of the forward prediction model cost dominating that of solving the optimal transport problems during assimilation. The efficiency gains are products of a seamless coupling scheme that successfully transforms weighted forecast ensembles of finite element discretizations of random functions, with fine and coarse resolutions, to analysis ensembles that are positively correlated with one another. This coupling holds in two cases, where assimilation starts from a known state and where assimilation is delayed until later in the simulation.

Due to the spatial discretization of the problems in this chapter, the sample sizes of the difference estimators in the multilevel framework should be set to decay more rapidly than in only temporally discretized problems. Therefore approximations on a coarse level of resolution will be the bottle-neck to the computational cost of solving the optimal transport problems required at assimilation stages in the MLETPF. Although it is not investigated here, due to effective results being established regardless, it is a likely future research direction to implement the iterative optimal transport algorithms [Cuturi, 2013] for these coarser approximations, possibly by using a second-order accurate framework [de Wiljes et al., 2016]. This would extend the regime of ϵ values that the MLETPF achieves speed-ups over the standard ETPF for, see Sec. 4.4.

Chapter 7

Conclusions

This thesis has presented a way of using multilevel Monte Carlo (MLMC), a novel variance reduction technique for statistical estimators, to increase the efficiency of ensemble data assimilation. Multilevel Monte Carlo [Giles, 2008, Cliffe et al., 2011] is a generic framework applicable to many areas within computational statistics, including ensemble data assimilation. The technique can significantly reduce the computational expense of estimating statistics to discretized random variables. A number of unique challenges arise when applying MLMC to ensemble data assimilation, and these have been carefully addressed throughout this thesis by utilising a wide range of existing techniques, such as optimal transportation [Villani, 2008].

Ensemble data assimilation, in particular filtering, is a natural application for MLMC, due to the large computational cost associated with propagating ensembles of model simulations forward in time and space, especially in high-dimensional cases. These ensembles make up approximations to potentially nonlinear forecast distributions of states from random systems, conditioned on partial observations. Ensemble transform methods for data assimilation, such as the ensemble transform particle filter (ETPF) [Reich, 2013], are effective algorithms to generate these ensembles. As a benchmark for applying MLMC to ensemble data assimilation, the ETPF has been considered in this thesis. A particular localisation scheme can be used in combination with this algorithm to make it applicable to high-dimensional cases [Cheng and Reich, 2013]. The resultant methodology of this application is referred to as 'the multilevel ETPF' (MLETPF) throughout.

Multilevel Monte Carlo generates statistical estimates via a linear combination of a sequence of independent Monte Carlo estimators, each approximating the difference between

two adjacent levels of discretization for the same random variable. It is important that the coarse and fine ensembles in each of these sub-estimators are positively correlated. This allows for a 'trade off' between bias, from the levels of discretization, and sampling variance, in each sub-estimator. In turn this leads to computational cost reductions relative to a standard Monte Carlo equivalent with the same order of accuracy. Maintaining such a coupling between coarse and fine ensembles during resampling in nonlinear filtering algorithms is the challenge unique to applying MLMC to this research area.

Additionally, this thesis extends the scope of evaluation for MLMC approximations past that of just error analysis. A framework for applying scoring rules for ensemble forecasts to MLMC approximations is proposed. In standard ensemble forecasting literature, scoring rules use observations from a process that one is trying to forecast in order to evaluate properties of the forecast. Indeed, data assimilation can be utilised in combination with the observations used in this context. The proposed framework allows the hierarchy of ensembles, that form the difference estimators in the MLMC method, to be transformed into a single ensemble where it is possible to apply the aforementioned scoring rules.

Unlike the standard MLMC framework proposed by Giles [2008], where the computational cost of propagating the ensembles forward in time and space (the so-called forward model cost) made up the entire cost of the estimators, multilevel ensemble data assimilation algorithms have an additional cost of the assimilation stages. Therefore, for the MLMC application to have a significant impact on the overall computational cost of producing filtering estimators, the forward model cost must dominate the cost of the assimilation stages. This dominating cost has been found to occur in two different ways:

- Firstly, localisation that alleviates likelihood degeneracy from high-dimensional states, can also be used to reduce the computational cost of the assimilation stages in the ETPF algorithm that require optimal transport problems to be solved. 'Full localisation' is used in order for this to occur, meaning all correlations amongst state components are assumed to be negligible during assimilation. Like the localisation used in the ensemble Kalman filter, this does lead to inconsistency. However the resulting filters are still found to be stable in models such as the Lorenz 96 equations [Gregory and Cotter, 2017b].

- Secondly, for high-dimensional systems, the forward model cost of the ETPF can be large enough, for some orders of accuracy, to dominate the assimilation cost of the

MLETPF. Therefore MLETPF estimators can be computed with overall computational cost reductions relative to that of the standard ETPF. However as the desired order of accuracy of these estimators increases, the assimilation cost will likely dominate that of the forward model, hindering the cost reductions that the MLETPF achieves.

In order to investigate the latter point, this thesis has explored an extension to the localisation technique for the ETPF presented in Cheng and Reich [2013], specifically for finite element approximations. The extension uses $\mathbb{L}_2$-projection on a hierarchy of meshes to localise likelihoods and distance metrics associated with the optimal transport problems solved during assimilation stages. This framework enables observations, taken from a reference field, to be filtered into an ensemble of finite element discretizations of a random field. An example of this is illustrated by the stochastic quasi-geostrophic equations, where stochasticity in the streamfunction is modelled via Stratanovich noise. The ETPF filtering approximation, applied in combination with the proposed localisation technique, can successfully track the reference field, even when assimilation is delayed for a period of time at the start of simulation.

The MLETPF framework using this localisation technique is also applied to the stochastic quasi-geostrophic equations. Here, filtering estimators have been computed for a significantly reduced computational cost relative to that of the standard ETPF method. It is because of the large, dominating computational cost of the forward model in this case that these overall cost reductions are achieved, even with solving only partially, and not fully, localised optimal transport problems[1] at each assimilation stage. This demonstrates how the MLETPF can achieve computational cost reductions relative to the ETPF in partially localised as well as fully localised cases.

A future direction of this research is to apply iterative schemes for solving the optimal transport problems that appear within the assimilation stages of the MLETPF framework. Used in combination with a second order correction, as done in de Wiljes et al. [2016], this could reduce the computational cost of these assimilation stages. Given that the majority of this computational cost is from the coarsest sub-estimators in the MLETPF estimator, using iterative schemes for solving the optimal transport problems within just these sub-estimators could extend the regime of accuracies that the MLETPF achieves cost reductions relative to the ETPF for.

[1] Without full localisation, the expensive $\mathcal{O}(N^3 \log N)$ algorithm has to be used.

The MLETPF framework offers an efficient, powerful tool for ensemble data assimilation. This field of research is being utilised in an increasing number of applications, including incorporating in-situ observations into weather forecasts. As such it is important that research into both high-dimensional ensemble nonlinear filtering, and the associated efficiency of the process is progressed as a priority. This thesis attempts to increase understanding of both of these aspects.

References

J. L. Anderson. Localization and sampling error correction in ensemble Kalman filter data assimilation. *Monthly Weather Review*, 140(7):2359–2371, 2012.

K. Bergemann and S. Reich. A localization technique for ensemble Kalman filters. *Quarterly Journal of the Royal Meteorological Society*, 136(648):701–707, 2010.

E. Bernsen, O. Bokhove, and J. J. van der Vegt. A (dis) continuous finite element model for generalized 2D vorticity dynamics. *Journal of Computational Physics*, 211(2):719–747, 2006.

A. Beskos, D. Crisan, A. Jasra, K. Kamatani, and Y. Zhou. A stable particle filter in high-dimensions. *arXiv preprint arXiv:1412.3501*, 2014.

A. Beskos, A. Jasra, K. Law, R. Tempone, and Y. Zhou. Multilevel sequential Monte Carlo samplers. *Stochastic Processes and their Applications*, 127(5):1417–1440, 2017.

O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

M. Carney and P. Cunningham. Evaluating density forecasting models. Technical report, Trinity College Dublin, Department of Computer Science, 2006.

Y. Cheng and S. Reich. A McKean optimal transportation perspective on Feynman-Kac formulae with application to data assimilation. *arXiv preprint arXiv:1311.6300*, 2013.

A. Chernov, H. Hoel, K. Law, F. Nobile, and R. Tempone. Multilevel ensemble Kalman filtering for spatially extended models. *arXiv:1608.08558*, 2016.

N. Chopin. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411, 2004.

N. Chustagulprom, S. Reich, and M. Reinhardt. A hybrid ensemble transform particle filter for nonlinear and spatially extended dynamical systems. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):592–608, 2016.

K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14(1):3–15, 2011.

B. Cockburn and C.-W. Shu. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of scientific computing*, 16(3):173–261, 2001.

S. E. Cook. *Multi Level Monte Carlo Methods for Atmospheric Dispersion Modelling*. PhD thesis, University of Bath, 2013.

M. Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.

R. Daley. Atmospheric data assimilation. *Journal of the Meteorological Society of Japan. Ser. II*, 75(1B):319–329, 1997.

J. de Wiljes, W. Acevedo, and S. Reich. A second-order accurate ensemble transform particle filter. *arXiv preprint arXiv:1608.08179*, 2016.

R. Douc, O. Cappé, and E. Moulines. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69, 2005.

A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. In *Oxford Handbook of Nonlinear Filtering*, 2011.

H. Duan, P. Lin, P. Saikrishnan, and R. C. Tan. L2-projected least-squares finite element methods for the Stokes equations. *SIAM journal on numerical analysis*, 44(2):732–752, 2006.

C. A. T. Ferro, D. S. Richardson, and A. P. Weigel. On the effect of ensemble size on the discrete and continuous ranked probability scores. 2008.

M. Giles. Multilevel Monte Carlo methods. In *Monte Carlo and Quasi-Monte Carlo Methods*, volume 65, pages 83–103. Springer, 2013.

M. B. Giles. Multilevel Monte Carlo path simulation. *Oper. Res.*, 56(3):607–617, 2008.

M. B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015.

M. B. Giles and B. J. Waterhouse. Multilevel quasi-Monte Carlo path simulation. *Advanced Financial Modelling, Radon Ser. Computat. and Appl. Math.*, 8:165–181, 2009.

M. B. Giles, T. Nagapetyan, and K. Ritter. Multilevel Monte Carlo approximation of distribution functions and densities. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1): 267–295, 2015.

W. R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

T. Gneiting, F. Balabdaoui, and A. E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2): 243–268, 2007.

S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM review*, 43(1):89–112, 2001.

A. Gregory and C. J. Cotter. On the calibration of multilevel Monte Carlo ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 143:1929–1935, 2017a.

A. Gregory and C. J. Cotter. A seamless multilevel ensemble transform particle filter. *SIAM Journal on Scientific Computing*, 39:A2684–A2701, 2017b.

A. Gregory, C. J. Cotter, and S. Reich. Multilevel ensemble transform particle filtering. *SIAM Journal on Scientific Computing*, 38(3):A1317–A1338, 2016.

A. Haji-Ali, F. Nobile, and R. Tempone. Multi-index Monte Carlo: when sparsity meets sampling. *Numerische Mathematik*, 132(4):767–806, 2016.

D. J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, 43(3):525–546, 2001.

H. Hoel, E. Von Schwerin, A. Szepessy, and R. Tempone. Adaptive multilevel Monte Carlo simulation. In *Numerical Analysis of Multiscale Computations*, pages 217–234. Springer, 2012.

H. Hoel, K. J. H. Law, and R. Tempone. Multilevel ensemble Kalman filtering. *SIAM Journal on Numerical Analysis*, 54(3):1813–1839, 2016.

J. D. Hol, T. B. Schon, and F. Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 79–82. IEEE, 2006.

D. D. Holm. Variational principles for stochastic fluid dynamics. In *Proc. R. Soc. A*, volume 471, page 20140963. The Royal Society, 2015.

A. Jasra, K. Kamatani, K. J. H. Law, and Y. Zhou. Multilevel particle filter. *SIAM Journal on Numerical Analysis*, 55(6):3068–3096, 2017a.

A. Jasra, K. J. H. Law, and C. Suciu. Advanced multilevel Monte Carlo methods. *arXiv preprint arXiv:1704.07272*, 2017b.

G. Katsiolides, E. H. Müller, R. Scheichl, T. Shardlow, M. B. Giles, and D. J. Thomson. Multilevel Monte Carlo and improved timestepping methods in atmospheric dispersion modelling. *Journal of Computational Physics*, 354:320–343, 2018.

C. Ketelsen, R. Scheichl, and A. L. Teckentrup. A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow. *Center for Applied Scientific Computing*, 2013.

K. J. Law, H. Tembine, and R. Tempone. Deterministic mean-field ensemble Kalman filtering. *SIAM Journal on Scientific Computing*, 38(3):A1251–A1279, 2016.

Y. Ma, M. G. Genton, and E. Parzen. Asymptotic properties of sample quantiles of discrete distributions. *Annals of the Institute of Statistical Mathematics*, 63(2):227–243, 2011.

P. D. Moral, A. Jasra, K. J. H. Law, and Y. Zhou. Multilevel sequential Monte Carlo samplers for normalizing constants. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 27(3):20, 2017.

J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

J. Olsson and S. Bizjajeva. Antithetic sampling for sequential Monte Carlo methods with application to state-space models. *Annals of the Institute of Statistical Mathematics*, 68 (5):1025–1053, 2016.

S. Pauli and P. Arbenz. Determining optimal multilevel Monte Carlo parameters with application to fault tolerance. *Computers & Mathematics with Applications*, 70(11):2638–2651, 2015.

O. Pele and M. Werman. Fast and robust Earth Mover's Distances. In *Computer Vision, IEEE 12th International Conference*, pages 460–467, 2009.

J. Poterjoy. A localized particle filter for high-dimensional nonlinear systems. *Monthly Weather Review*, 144(1):59–76, 2016.

F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G. Bercea, G. R. Markall, and P. H. J. Kelly. Firedrake: automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software (TOMS)*, 43(3): 24, 2016.

P. Rebeschini and R. Van Handel. Can local particle filters beat the curse of dimensionality? *arXiv preprint arXiv:1301.6585*, 2013.

S. Reich. A nonparametric ensemble transform method for Bayesian inference. *SIAM J. Sci. Comput.*, 35(4):A2013–A2024, 2013.

S. Reich and C. J. Cotter. Ensemble filter techniques for intermittent data assimilation. *Large Scale Inverse Problems. Computational Methods and Applications in the Earth Sciences*, 13:91–134, 2013.

S. Reich and C. J. Cotter. *Probabilistic Forecasting and Bayesian Data Assimilation*. Cambridge Univ. Press, 2015.

C. A. Rohde. *Introductory Statistical Inference with the Likelihood Function*. Springer, Switzerland, 2014.

R. Schefzik, T. L. Thorarinsdottir, and T. Gneiting. Uncertainty quantification in complex simulation models using ensemble copula coupling. *Statist. Sci.*, 28(4):616–640, 2013.

D. Sen, A. Thiery, and A. Jasra. On coupling particle filter trajectories. *Statistics and Computing*, 28(2):461–475, 2018.

C. Villani. *Optimal transport: Old and New*, volume 338. Springer Science & Business Media, 2008.

Zenodo/Firedrake. Firedrake: an automated finite element system, Oct. 2017. URL `https://doi.org/10.5281/zenodo.1039613`.