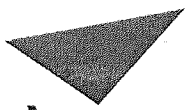
BAYESIAN DATA ASSIMILATION

REMI LOUIS BARILLEC

Doctor Of Philosophy



Aston University

– ASTON UNIVERSITY –

December 2008

ASTON UNIVERSITY

Bayesian Data Assimilation

REMI LOUIS BARILLEC

Doctor Of Philosophy, 2008

Thesis Summary

This thesis addresses data assimilation, which typically refers to the estimation of the state of a physical system given a model and observations, and its application to short-term precipitation forecasting. A general introduction to data assimilation is given, both from a deterministic and stochastic point of view. Data assimilation algorithms are reviewed, in the static case (when no dynamics are involved), then in the dynamic case. A double experiment on two non-linear models, the Lorenz 63 and the Lorenz 96 models, is run and the comparative performance of the methods is discussed in terms of quality of the assimilation, robustness in the non-linear regime and computational time.

Following the general review and analysis, data assimilation is discussed in the particular context of very short-term rainfall forecasting (nowcasting) using radar images. An extended Bayesian precipitation nowcasting model is introduced. The model is stochastic in nature and relies on the spatial decomposition of the rainfall field into rain "cells". Radar observations are assimilated using a Variational Bayesian method in which the true posterior distribution of the parameters is approximated by a more tractable distribution. The motion of the cells is captured by a 2D Gaussian process. The model is tested on two precipitation events, the first dominated by convective showers, the second by precipitation fronts. Several deterministic and probabilistic validation methods are applied and the model is shown to retain reasonable prediction skill at up to 3 hours lead time. Extensions to the model are discussed.

Keywords: Data assimilation, Bayesian, precipitation, nowcasting, validation

Contents

List of Figures

List of Tables

To my parents

Acknowledgements

A PhD thesis is the tale of a journey. I remember setting off a long time ago, with very little knowledge as to where I'd go and how I'd get by. Several years later, looking back, I am amazed I have actually made it. There would have been many opportunities to get lost on the way if it hadn't been for the guidance of numerous people, from those providing occasional directions to those sharing the journey with us. There would have been scope for doubt and possibly failure if it hadn't been for the support of friends, family and colleagues in the difficult times. Many people have contributed, one way or another, to the completion of this work, and having shared the burden, deserve part of the credit.

First and foremost, I would like to thank my supervisor, Dan Cornford, for having been my guide on this journey. Dan lead the way with great enthousiasm and energy, always making time to answer questions and discuss research issues, and was there to keep me going in times of doubt. Working with Dan for four years has been a great and fun experience and I am pleased to say I couldn't have wished for better supervision.

I am grateful to Manfred Opper for sketching the big picture of the mathematical derivations in the rainfall model, to Jort van Mourik for his helpful discussion on the first year preliminary report, to David Saad and David Lowe for insightful comments during departmental seminars, and to everybody at NCRG for making it such a great place to work. Special thanks to Vicky Bond, our blessed department coordinator, for taking care of so many administrative tasks with a smile.

I would also like to thank colleagues from all over the world who have shown interest in my work at conferences and provided interesting suggestions, as well as the anonymous reviewers of the paper on precipitation nowcasting for numerous relevant comments. I am particularly grateful to the British Atmospheric Data Centre for providing the radar data used in this work free of charge and to the School of Engineering and Applied Science, Aston University, for funding the first two years of this PhD. I can also thank in advance the referees for the effort it will take them to read, understand and comment on this thesis.

Being a PhD student has been a great time, and I owe it in particular to my fellow students and friends: Dharmesh Maniyar, Erik Casagrande, Ben Ingram, Mixalis Vrettas, Matthew Williams, Thomas Bermudez, Rajeswari Matamb, Mingmanas Sivaraksa, Diar Nasiev, Jack Raymond, Alexis Boukouvalas, Naoki Yamawaki, Stephen Hall, Richard Hammond and so many others – you know who you are. Thanks in particular to Mixalis and Alexis for proof-reading this thesis.

I would also like to thank my parents, for stimulating my thirst for knowledge as a young boy, which eventually led me to embrace an academic career, and for respecting and supporting my choices as an adult. Special thanks to my brother and sister for innumerable hilarious conversations which have been as many breathes of fresh air along the way. I owe you one.

Particular thanks to my flatmates, Anna Topaka and Jorge Christian Guerrero Rodriguez, for coping with me over the last few years, and keeping the house going when I failed to do my share.

Although all these people have played a determining part in the achievement of this thesis, there is one person to whom I am in debt beyond measure: Marianna, my girlfriend, who walked the journey with me, day after day, and never stopped showing care, attention and support. Thank you for having been here, every single day.

Friends, family and especially my girlfriend have all been neglected in the months of writing up. I am in debt for your understanding and support during these last few months.

1 Introduction

CONTENTS

1.1 Data assimilation

1.1.1 Historical context

Records of scientific observations of the physical phenomenons surrounding us go back to the origins of civilisation. Several centuries B.C., Babylonians, Egyptians, Ancient Greeks were all active observers of astronomical phenomena and kept records of the movement of the sun and moon. These were used to design lunar-solar calendars and to predict the occurrence of eclipses (Evans, 1998). The prediction of eclipses is probably one of the earliest examples of the use of observations with a model (the periodicity of eclipses) for prediction purposes.

Data assimilation, the tracking of a physical process using a model and observations, had to wait until the 18th century to be given its first sound theoretical bases. The introduction of the telescope in the 17th century was a major breakthrough in astronomy, and brought a whole new dimension to the observation of planets and asteroids. It allowed for objects never before seen to be discovered and new accuracies of locations to be reached. With improved observations, Kepler was able to formulate his laws on the motion of celestial objects, and Galileo confirmed Copernicus' theoretical assumption that the Earth orbited around the Sun, not the opposite.

In the 18th century, the asteroid Ceres was spotted for the first time, soon before it disappeared in the vicinity of the Sun. Due to the small number of observations collected, the trajectory of the asteroid could not be extrapolated through traditional methods available at the time. However, Gauss was able to predict with astonishing accuracy the position and date of the reapparition of the asteroid, by applying a brand new method now commonly known as least square estimation (Lewis et al., 2006). Based on this concept of least square estimation, a whole field would soon grow to become one of the most widespread areas of science today, with applications in electronics, astronomy, engineering, aeronautics, environmental modelling, the automotive industry, and many more.

1.1.2 General overview of data assimilation

Data assimilation involves two basic components: a *model*, which is responsible for reproducing as well as possible the process of interest, and *observations*, which can be used to estimate the model parameters (the *state*) in space and, often too, in time. Both the model and the observations are known to be imperfect, due to numerical approximations, incomplete formulation of the physical reality, limitations inherent to measurement devices, etc. As a consequence, any estimate obtained through a data assimilation method is bound to be inaccurate and knowledge about how confident in this estimate we are is critical. Data assimilation, at the end of the day, is about finding the optimal trade-off between a model and observations which are both inevitably wrong.

Two major approaches to data assimilation can be found in the literature. The first approach addresses the problem in a *deterministic* fashion, seeking a single, optimal estimate of the true process, while the second seeks a *stochastic* solution which also captures the uncertainty associated with the estimate. Stochastic data assimilation is typically formulated within a Bayesian framework, in which the probability distribution of the estimate is tracked rather than the estimate alone.

Data assimilation is a wide and active research field, and a plethora of methods have been developed in various contexts to address the same problem. The first aim of this thesis is to present an up-to-date review of the most commonly used methods in data assimilation and discuss their respective advantages and drawbacks. The methods are compared on two non-linear toy models widely used in the meteorological community: the Lorenz 63 and the Lorenz 96 models.

1.1.3 Application to precipitation nowcasting

The second aspect of this thesis is the application of data assimilation to the particular problem of very short-term precipitation forecasting (nowcasting). A new advection-based model for precipitation nowcasting is introduced. The spatial model relies on a decomposition of radar rainfall fields into small entities of predefined shape (rain "cells"). The parameters of these cells are estimated in a fully probabilistic manner. The motion (or advection) of the cells is modelled by a two dimensional Gaussian process, with parameters inferred from the cells' displacement. The model is tested on real data and both deterministic and probabilistic forecast validation techniques are used in the analysis of its performance.

1.2 Scientific contribution

This thesis looks at data assimilation and its application to meteorological problems with the particular example of precipitation forecasting.

The contribution of this thesis to the scientific community, and especially to the data assimilation field, can be summarised by the following points:

- The first, relatively minor, contribution of this work is the synthesis of knowledge about data assimilation gathered from various domains and its organisation in a logical manner, so as to provide an up-to-date, extensive overview of the field. Although many textbooks exist on the subject, most of them provide a thorough treatment of one or two data assimilation methods, at the expense of others, which are often simply omitted. We have tried to take

the opposite approach and provide an extensive overview of data assimilation methods, and relate them in a meaningful way.

- Although several of the methods discussed in this work have been separately compared one with another, the lack of a common benchmark in the data assimilation community makes it very difficult to assess the relative strengths and weaknesses of each method. By providing two experiments in which all currently deployed data assimilation methods are compared on two well established non-linear models, we are able to provide a general comparative overview of current data assimilation methods. More importantly perhaps, it is hoped that this piece of research will help the community towards a common benchmark against which future data assimilation methods can be compared to existing ones. To that effect, attention has been paid to providing all necessary parameters for each experiment to be fully reproducible.

- Having covered data assimilation from a general point of view, a new method is introduced for the particular problem of precipitation forecasting at very short lead times (*nowcasting*). The new data assimilation method is tailored for this specific problem but could also be easily generalised to other domains. The method relies on a technique called *variational Bayesian inference* in which the posterior distribution of the state is estimated through minimisation of a cost function: the distance between the exact posterior distribution (as given by Bayes rule) and a user-defined approximating distribution is minimised. It is shown how the model is able to capture the spatial structure of precipitation fields from radar images and estimate its motion in a fully probabilistic manner. The model shows very promising probabilistic forecast skill up to 2h ahead in time, which is good for a nowcasting system.

- A last contribution is the development of a numerical data assimilation framework which has been designed for extendability and ease of use. The framework has been developed in the C++ programming language, which is one of the standard languages used in industrial applications. Support is provided for most data assimilation methods discussed in this work (and a few additional ones) and custom experiments are easily set up. Attention has been paid to keep the framework modular so that users can add their own models and new assimilation methods in a simple manner. Because most data assimilation systems involve large amounts of high-dimensional data, efficiency has been a priority when implementing the framework.

1.3 Outline of the thesis

Chapter 1 is this introduction.

Chapter 2 introduces the general concepts providing the basis for data assimilation. Models and observations are discussed and their mathematical formulation is given, both in a deterministic and stochastic context. The data assimilation problem is then defined.

Chapter 3 discusses data assimilation in the static context, where no dynamics are taken into account. It is shown how data assimilation in this context can be motivated by a natural least squares formulation. An "optimal" solution to the least squares estimation problem is given for the case where the observations are related linearly to the state. The non-linear case is also discussed. An alternative approach based on variational techniques (3D VAR) is reviewed. The discussion is initialy exposited from the deterministic point of view. The (more Bayesian) stochastic viewpoint is subsequently discussed.

Chapter 4 takes data assimilation one step further by adding the system dynamics that were omitted thus far. Changes in the formulation are discussed, here again both from the deterministic and stochastic points of view. The deterministic formulation gives rise to a dynamic least squares algorithm, which can be approximated using variational techniques (4D VAR). The stochastic approach leads to filtering algorithms such as the well-known Kalman filter, which is optimal in the linear case. Several extension of the Kalman filter to address non-linear models are discussed.

Chapter 5 presents two experiments in which the data assimilation methods introduced in the previous chapter are run on two non-linear models using a perfect model setting (the model is assumed known). The quality of the assimilation is evaluated for each method and a comparison of the results is carried out. The software implementation is briefly discussed.

Chapter 6 takes us from general data assimilation to the particular problem of precipitation assimilation and forecasting with radar observations. A Bayesian framework is introduced, which relies on a decomposition of the observed precipitation field into Gaussian-shaped rain cells. Dynamics are provided by an advection (or motion) field modelled using a two-dimensional Gaussian process. The novel variational Bayesian data assimilation method is described.

Chapter 7 applies the precipitation model discussed in Chapter 6 to two large scale experiments on real data. The performance of the model is assessed using both deterministic and probabilistic validation methods.

Chapter 8 summarises the work presented in the previous chapters and gives future directions of research.

1.4 Disclaimer

The work presented in this thesis is original and has not been published anywhere else. Parts of the work, however, have been presented in the following conferences and papers:

- Preliminary work on Particle Filters with the Lorenz 96 system was presented at the Annual Meeting of the Royal Meteorological Society, 2005 (oral presentation) (Barillec and Cornford, 2005)
- The comparison of data assimilation methods on the two Lorenz models (Chapter 5) was presented at the European Geosciences Union conference, 2006 (poster presentation) (Barillec and Cornford, 2006)
- The precipitation nowcasting model was presented at the Weather Radar and Hydrology conference, 2008 (oral presentation) (Barillec and Cornford, 2008b)
- A paper on the precipitation nowcasting model has been accepted for publication in Advances in Weather Resources (Barillec and Cornford, 2008a)
- The work in this thesis has been presented and discussed at several departmental seminars in the NCRG, Aston University
- Several experiments using the data assimilation framework (Section 5.2) were provided for Shen et al. (2007)

2 Introduction to Data Assimilation

CONTENTS

2.1 Chapter outline

Given observations of a physical process and a model, what is the "optimal" estimate of the process one can achieve? How is that estimate computed in practice? The aim of this chapter and the following is to discuss the answers to these two questions.

The chapter is organised as follows. Section 2.2 outlines the general concepts and gives the necessary background for the rest of the discussion. Models and observations are discussed, both from a deterministic and stochastic point of view. The data assimilation problem is posed in Section 2.3 and the bases for its treatment in a Bayesian framework are laid down.

There are a large number of books and papers on data assimilation and filtering theory in which the reader will find excellent introductions to the topic. Lewis et al. (2006) provide a particularly thorough coverage of data assimilation methods, in a consistent, extensive, well illustrated manner. Stochastic data assimilation is discussed at great length in Jazwinski (1970), another personal favourite for its rigorous and exhaustive treatment of data assimilation with stochastic models. The reader will find many other introductions in the literature, some of those we found useful include Rhodes (1971); Maybeck (1979); Cohn (1997); Bouttier and Courtier (1999); Holm (2003). For more meteorology-oriented approaches, the reader is referred to Daley (1991); Kalnay (2003) and references in the review paper by Dance (2004).

2.2 Models and observations: general notions

2.2.1 The state-space framework

Dynamical system

There is an infinity of physical processes one might be interested in modelling: the movement of planets, the interaction of atoms, the development of precipitation in the atmosphere, the propagation of some disease in some animal population, etc. Because data assimilation is interested in tracking and predicting the state of processes which in most applications are time-dependent, a model is necessary which expresses the temporal changes of the process. Such a time-dependent model is referred to as a *dynamical system* or *dynamical model*.

According to Weisstein (2002), a dynamical system can be defined as "*a means of describing how one state develops into another state over the course of time.*" A dynamical system is thus a mathematical formulation of the various factors we assume are responsible for the dynamics of the process. When dynamical systems are designed to understand and reproduce the evolution of phenomena observed in the real world, they are also referred to as *dynamical models* or simulators. In this work, the terms "system" and "model" are considered equivalent and used interchangeably.

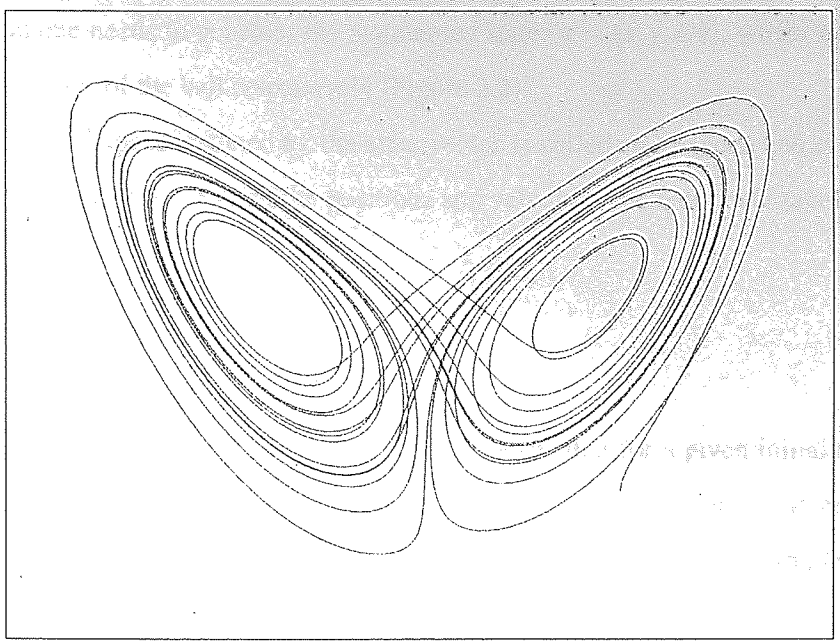


Figure 2.1: An example of deterministic system: the Lorenz 63 dynamical system, system is plotted here in the (x,z) plane. The trajectory of the state orbits around two attractors, giving the well-known "Lorenz butterfly".

The term "process" is used to refer to the "true" physical phenomenon observed. The system is thus a conceptual representation of the process.

State space

In order for the system to be manageable, we need to assume that there exists a finite set of variables sufficient to fully describe its state at any given time. This set of variable is referred to as the *state vector* (or simply *state*) and denoted $\mathbf{x}$ throughout this work.

The state of the system depends very much on the use one wants to make of the system. For example, the state of a ball on an inclined plane can be described by its position in space, identified as a set of coordinates relative to some predefined origin: $\mathbf{x} = (x, y)$. We here assume that the ball is observed in a two-dimensional region orthogonal to the plane it is rolling on, indexed using the usual (x, y) cartesian system.

However, if one is interested in the trajectory of the ball, this set of variables is inadequate. One could imagine two different situations where the ball lies at the same position, but with different velocities. It is easy to see how these two situations would give rise to two different trajectories. As such, the position of the ball is not a sufficient set

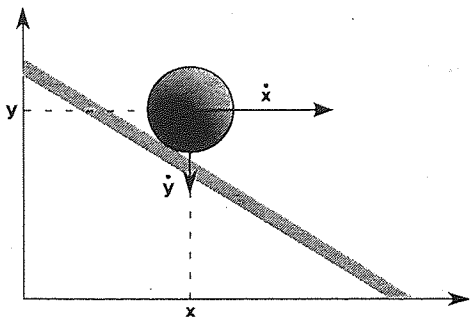


Figure 2.2: A ball on a plane

of variables, and one needs to consider the augmented state: $\mathbf{x} = (x, y, \dot{x}, \dot{y})$, where $\dot{x}$ and $\dot{y}$ denote the x- and y-velocities of the ball respectively (Figure 2.2).

The mathematical space spanned by the state vector is called *state space* and corresponds to all possible values $\mathbf{x}$ can take (e.g. all the positions and velocities the ball could take).

2.2.2 Systems

Deterministic system

A dynamical system is *deterministic* in nature, which means that for a given initial condition $\mathbf{x}_0$, the future evolution of the system is fully determined (as opposed to *stochastic* systems where the same initial condition can lead to different outcomes). Figure 2.1 shows an example of one such deterministic system commonly found in the literature: the Lorenz system (Lorenz, 1963). This system, although deterministic, presents the interesting peculiarity of being extremely sensitive to initial conditions and helped develop the theory of chaotic systems.

The evolution of a deterministic system can be expressed as a *rule* relating the state of the system at a given instant to its state at a later time. This rule is generally expressed in terms of a differential equation:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{dt}} = m(\mathbf{x}, \mathrm{t}), \tag{2.1}$$

where m is the *system operator* responsible for propagating the state forward in time.

Stochastic system

Because deterministic systems obey a fixed rule, if one knew the operator m precisely and the initial state $\mathbf{x}_0$, one would be able to predict the state of the system at any future time. In practice, however, any physical phenomenon we observe is the result of so many causes that only the most important of these can be identified.

For instance, the ball from our previous example slides down the plane with increasing velocity because of the gravitational force. A simple model can be devised which relates the position and velocity of the ball to that force through the application of Newton's laws of motion. Although such a model would provide a reasonable representation of the dynamics responsible for the ball's motion, there are many minor factors it does not take into account, such as the fact that the plane is not a perfect mathematical plane but presents some irregularities, that it is not perfectly smooth either and the ball is subject to a friction force, the fact that the room where the experiment takes place has open windows and air currents affect (albeit to a very small extent) the ball's velocity...

We just illustrated the fact that, despite our best efforts, any phenomenon we observe can only be approximated by an incomplete system. The imperfection of the system needs to be included

in its formulation. To that effect, a stochastic term can be appended to Equation (2.1) to account for the discrepancy between the system and the true process:

$$\frac{d\mathbf{x}}{dt} = m(\mathbf{x},t) + \eta(\mathbf{x},t). \tag{2.2}$$

$\eta(\mathbf{x},t)$ represents the influences of all the missing factors in the model and is usually referred to as *system noise* or *model error*. The resulting model is no more deterministic, due to the stochastic nature of the model error term.

Typically, system noise is a consequence of one or more of the following factors:

- *Formalisation errors*, resulting from an invalid or incomplete physical understanding of the true process.
- *Discretisation errors*: most physical processes we observe are continuous in space and time. However, computer models and data storage require the infinite continuous domain to be projected onto a finite discrete domain.
- *Numerical errors*: round-off errors, approximations in numerical solvers, linearisation...

2.2.3 Observations

Deterministic observation

An immediate consequence of the system's imperfection is that our ability to deterministically predict the evolution of the true process is limited. Fairly soon, the factors that have not been correctly accounted for will make the model diverge from the process. How far ahead that limit lies depends essentially on how close our system is to the true underlying process intially, and how sensitive to small changes the process in question is.

In order to overcome the limitation of our models, information about the true state of the process is needed to correct the state of the system before the system diverges. However, in most cases, the state cannot be observed directly. Instead, some other quantity $\mathbf{y}$ is measured and related to the state through some function h:

$$\mathbf{y}(t) = h(\mathbf{x},t). \tag{2.3}$$

h is called the *observation operator* and is a mapping from *state space* to *observation space*. $\mathbf{y}$ is simply called the *observation*. Depending on the problem, the relation between the measurements and the state can be linear (including direct, i.e. $h(\mathbf{x}) = \mathbf{x}$) or non-linear (satellite radiance, radar reflectivity...).

Figure 2.3 illustrates the concepts of model divergence and estimation of the state in the case of direct, noise-free observations. In that case, h is assumed to be the identity, so that $\mathbf{y}(t) = \mathbf{x}(t)$.

The evolution of the true process is represented by a dashed line, with time flowing from left to right. The evolution of the state through the model is represented by a solid line, which quickly diverges from the process. A perfect observation of the process, denoted by a star, is available at time t and used to update the estimate of the state.

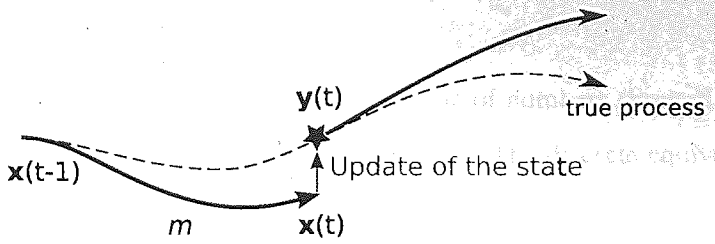


Figure 2.3: Schematic illustration of dynamic data assimilation (basic). The model is corrected using noise-free, direct observations (denoted by a star)

Stochastic observation

We mentioned how our models inevitably suffer from being inaccurate. A similar issue occurs with observations. Observations are obtained from measurement devices which suffer from their own limitations. Similarly to model error, observation error needs to be accounted for in Equation (2.3):

$$\mathbf{y}(t) = h(\mathbf{x}, t) + \varepsilon(\mathbf{x}, t), \tag{2.4}$$

where $\varepsilon(\mathbf{x}, t)$ represents the discrepancy from a perfect observation (as would be provided by a flawless measurement device). The stochastic term ε is called *observation error* or *measurement error*.

Observation error arises for the following reasons:

- *Measurement devices* are imperfect, they have limited accuracy, need to be calibrated by error-prone experts and rely on the exploitation of physical properties which are not always completely understood.
- *Post-processing* such as interpolation (to increase the resolution) and smoothing (to get rid of noise) can degrade the quality of the observation.
- *Projection*: often, measurements are converted into quantities that are more tractable. The relation between raw measurements and the projected quantities is a model in itself and suffers from the errors associated with models.
- *Relation to the state*: the operator h which relates the state to the observation is also a model subject to error.

- *Numerical errors* found in models also apply to observations (discretisation, approximations, etc.)

2.2.4 Further considerations

Discrete formulation

Because computers can only handle a finite representation of numbers (limited to a fixed number of bits), it is convenient to work in discrete time and space. The discrete equivalent of Equations (2.2) and (2.4) is:

$$\mathbf{x}_t = m_t(\mathbf{x}_{t-1}) + \eta_t, \tag{2.5}$$

$$\mathbf{y}_t = h_t(\mathbf{x}_t) + \varepsilon_t. \tag{2.6}$$

where m_t is the (integral) operator that maps the state from time $t-1$ to time t.

Error assumptions

We have made the assumption, in Equations (2.5) and (2.6), that the stochastic terms η_t and ε_t do not depend on the state. Although common in the data assimilation literature, such an assumption is not always justified as often the errors in the model vary in different regions of the state space. For instance, a model estimating the speed and location of an object is likely to be more accurate a lower speeds than at higher speeds. Similarly, observation error is often related to the state. For example, when measuring a signal's intensity, it is likely that noise will vary with changes in the signal properties (amplitude, wavelength). When such cases are identified, it is important to model the error as an additional process itself.

Furthermore, it is very often assumed that the errors are uncorrelated in time and cancel out when averaged (white noise). In mathematical terms, this means that the errors are drawn from a zero-mean distribution with diagonal covariance matrix. The realism of such an assumption will depend heavily on the causes for model/observation error listed above. A strong motivation for the use of white noise (in particular Gaussian white noise) is the simplicity of the associated computations.

2.3 Formulation of the data assimilation problem

2.3.1 Deterministic formulation

Sections 2.2.2 and 2.2.3 gave a mathematical formulation of the two components of data assimilation: *models* and *observations*. Figure 2.4 summarises the elements involved in data assimilation:

- Prediction: a model m_t propagates the state forward in time, from an estimate $\mathbf{x}^a_{t-1}$ to a predicted estimate $\mathbf{x}^f_t$
- Assimilation: observations $\mathbf{y}_t$ of the true process are confronted against the predicted estimate in order to generate an updated estimate $\mathbf{x}^a_t$ closer to the true, unknown process (dashed circle)
- The observation operator h_t is a mapping from state space to observation space.
- Both the model and the observation operator are subject to errors.

The question in data assimilation is then the following: *how do we use imperfect observations to correct an imperfect model, so that the model remains consistent with the true process?* Answers to that question are the object of Chapters 3 and 4.

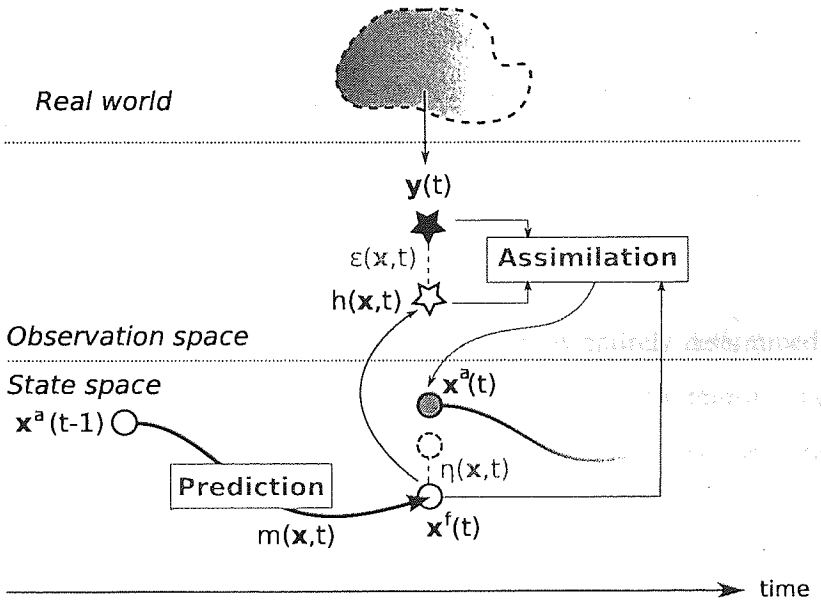


Figure 2.4: Schematic illustration of dynamic data assimilation (detailed). The state $\mathbf{x}$ is propagated forward in time using the model m (prediction). The observation $\mathbf{y}$ is then used to correct the predicted state $\mathbf{x}^f$, giving an updated state $\mathbf{x}^a$ (assimilation). The state is related to the observation through the observation operator h.

2.3.2 Stochastic formulation

The presence of noise both in the evolution and assimilation steps leads naturally to a probabilistic formulation of the problem, where the interest focuses not only on the state $\mathbf{x}$ alone, but also on the uncertainty associated with its estimate. In other words, one tries to infer the state's joint probability density function at all times, given the observations: $p(\mathbf{X}_t|\mathbf{Y}_t)$, where $\mathbf{X}_t = \{\mathbf{x}_0, ..., \mathbf{x}_t\}$ and $\mathbf{Y}_t = \{\mathbf{y}_1, ..., \mathbf{y}_t\}$.

If one is only interested in the state at the current time, as is often the case for real time process tracking applications, then the marginal distribution of the current state, $p(\mathbf{x}_t|\mathbf{Y}_t)$, is estimated, rather than the full joint probability density function. This distribution is known as the *filtering distribution*.

Propagation of the state

We assume here that the state's probability density function $p(\mathbf{X}_{t-1}|\mathbf{Y}_{t-1})$ is known and look at the effect of propagating the state through the model. When the state is propagated forward to the next time of interest (t), the joint probability density function is augmented with the new estimate of the state, so that we are now interested in $p(\mathbf{X}_t|\mathbf{Y}_{t-1}) = p(\mathbf{x}_t, \mathbf{X}_{t-1}|\mathbf{Y}_{t-1})$. This distribution is known as the joint *predicted distribution* of the state.

Clearly, this distribution depends on the previous estimate $p(\mathbf{X}_{t-1}|\mathbf{Y}_{t-1})$ and the nature of the model, expressed by the transition distribution $p(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{t-1})$:

$$p(\mathbf{X}_t|\mathbf{Y}_{t-1}) = p(\mathbf{x}_t, \mathbf{X}_{t-1}|\mathbf{Y}_{t-1}) \tag{2.7}$$

$$= p(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{t-1})\, p(\mathbf{X}_{t-1}|\mathbf{Y}_{t-1}). \tag{2.8}$$

Markov models

We have implicitly assumed that the evolution of the state is entirely determined by the model and the initial value of the state. This assumption is common for deterministic models, and is a result of the first-order differential equation (2.1). When Equation (2.1) is discretised into (2.5), it becomes clear that $\mathbf{x}_t$ only depends on $\mathbf{x}_{t-1}$.

A direct consequence of this assumption is that the distribution of the state is conditioned on the last estimate only: $p(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{Y}_{t-1})$. A system exhibiting this property is called a *Markov system*. Such a system gives a simplified predicted distribution:

$$p(\mathbf{X}_t|\mathbf{Y}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{Y}_{t-1})\, p(\mathbf{X}_{t-1}|\mathbf{Y}_{t-1}). \tag{2.9}$$

It is common in data assimilation to make the assumption that the system is Markov, for the reasons given above.

Update of the state: Bayesian inference (static)

Consider the simple case where our initial belief in the state is quantified by a probability distribution $p(\mathbf{x})$, and our estimate of the observation's accuracy given the state is given by a distribution $p(\mathbf{y}|\mathbf{x})$. By writing the joint probability density function of the state and the observation in two different ways:

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y})\, p(\mathbf{y}) = p(\mathbf{y}|\mathbf{x})\, p(\mathbf{x}), \tag{2.10}$$

we obtain Bayes' rule (Bishop, 1996; Bernardo and Smith, 1994; Lewis et al., 2006):

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})\, p(\mathbf{x})}{p(\mathbf{y})}. \tag{2.11}$$

In Bayesian language, $p(\mathbf{x})$ is called the *prior* (it expresses our uncertainty in the state prior to seeing observations), $p(\mathbf{y}_t|\mathbf{x}_t)$ is referred to as the *likelihood* (it expresses how likely the observation is given the current state) and the normalisation factor $p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x}$ is called the *evidence*. In practice, the expression of the likelihood will depend on the assumptions made on h_t and ε_t in Equation (2.6).

Update of the state: Bayesian inference (dynamic)

In a dynamic context, the prior is usually provided by a forecast and is thus equivalent to the predicted distribution of the state as given by Equation (2.8). The joint posterior, computed using Bayes rule, thus becomes:

$$p(\mathbf{X}_t|\mathbf{Y}_t) = \frac{p(\mathbf{Y}_t|\mathbf{X}_t)\, p(\mathbf{X}_t|\mathbf{Y}_{t-1})}{p(\mathbf{Y}_t)}. \tag{2.12}$$

A recursive formulation can be obtained by substituting (2.8) into (2.12):

$$p(\mathbf{X}_t|\mathbf{Y}_t) = \frac{p(\mathbf{Y}_t|\mathbf{X}_t)\, p(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{t-1})}{p(\mathbf{Y}_t)} p(\mathbf{X}_{t-1}|\mathbf{Y}_{t-1}). \tag{2.13}$$

The posterior distribution can be derived exactly when both the model m and the observation operator h are linear and the various errors are Gaussian. Unfortunately, either or both operators are non-linear in most problems of interest. The exact posterior distribution cannot generally be derived directly in such cases, and one has to resort to approximation techniques.

Conditionally uncorrelated observations

When the observations are uncorrelated, i.e. each observation only depends on the state at the same time, then the joint likelihood factorises as follows:

$$p(\mathbf{Y}_t|\mathbf{X}_t) = \prod_{k=1}^{t} p(\mathbf{y}_k|\mathbf{x}_k) \tag{2.14}$$

This is a common assumption in data assimilation, the realism of which will depend essentially on the nature of the measurement procedure used.

2.4 Summary of this chapter

2.4.1 Summary of this chapter

This chapter discussed the overall concept of data assimilation. The historical context was briefly highlighted, showing how data assimilation evolved from a computational tool used by 18th century astronomers to become the full branch of science with ubiquitous applications as we know

it today. The key notions of *model*, *state* and *observations* were then introduced, both from a deterministic and stochastic point of view. The existence of errors in both models and observations was highlighted and some of the consequences mentioned. Assumptions commonly made in the data assimilation community, namely the hypothesis of Gaussian noise and Markov models, were discussed. The formulation of the data assimilation problem was given, both from a deterministic and a stochastic point of view.

This aim of this chapter was to give the reader a general overview of the basic notions in data assimilation. We have tried to keep this chapter concise while providing a sufficient, though by no means complete, account of data assimilation. This means that some points could only be addressed briefly. Several references have been given in introduction for the interested reader to find more thorough discussions on the different concepts introduced in this chapter.

The following two chapters discuss the practical application of data assimilation to the static context (no model) and dynamic context (with model) respectively.

2.4.2 Summary of notations

Notations used in this chapter and the following are summarised in Table 2.1. These notations are provided at this stage for future reference and will be introduced in the relevant sections.

Notation	Meaning
$\mathbf{x}$	State of the system
$\mathbf{x}^b$	Background, i.e. prior estimate of state
$\mathbf{x}^f$	Predicted state (forecast)
$\mathbf{x}^a$	Analysis, i.e. updated state after observation has been assimilated
$\bar{\mathbf{x}}$	Mean state (possibly indexed: $\bar{\mathbf{x}}^b$, $\bar{\mathbf{x}}^f$, $\bar{\mathbf{x}}^a$)
$\mathbf{P}$	State covariance matrix (possibly indexed: $\mathbf{B} \equiv \mathbf{P}^b$, $\mathbf{P}^f$, $\mathbf{P}^a$)
m	Model/System operator
η	Model/System error
$\mathbf{Q}$	Model/System error covariance matrix
h	Observation operator
ε	Observation error
$\mathbf{R}$	Model/System error covariance matrix
t	Time
$\mathbf{X}_t$	State estimates up to and including time t: $\mathbf{X}_t = \{\mathbf{x}_0, \ldots, \mathbf{x}_t\}$
$\mathbf{Y}_t$	Observations up to and including time t: $\mathbf{Y}_t = \{\mathbf{y}_0, \ldots, \mathbf{y}_t\}$

Table 2.1: Summary of notations used in this chapter

3 Static data assimilation

CONTENTS

3.1 Foreword

Static data assimilation tackles the problem of finding the best estimate of the state's distribution when no dynamics are involved, i.e.:

$$m_t(\mathbf{x}_t) = \mathbf{x}_t. \tag{3.1}$$

In this case, the model is the identity and can effectively be discarded. The data assimilation problem thus reduces to estimating the state given one (or possibly several) observations.

We first discuss the deterministic approach to the problem in Section 3.2. Two cases are considered. In the first, it is assumed that no prior estimate of the state is available. The notion of Least Squares Estimate is introduced and a solution derived. In the second case, some a priori information about the state is available and an extended formulation of the Least Squares solution is given. An alternative solution based on variational methods, known in data assimilation as 3D VAR, is then considered. Most of the derivations in this chapter apply to the case where the state is related linearly to the observation. Application to the non-linear case is discussed.

The stochastic approach to the problem is the object of Section 3.3. It is assumed in that section that a priori information about the state is available, and that the probability density function of the observation error is known too, so that a Bayesian treatment of the problem can be applied. An optimal solution can be derived in the case of a linear observation operator and Gaussian distributions. Extensions to the non-linear/non-Gaussian case are mentionned.

Note that, since there is no dynamics involved in this chapter, time indexing is dropped in order to keep the notation as light as possible.

3.2 Deterministic approach

In this section, we consider the case where the relation between the state and the observations is completely deterministic. Although we know that the observations are imperfect, we assume we have no knowledge about the nature of the observation error.

3.2.1 No background information

Linear Least Squares Estimate - Single observation

We consider first the case where the state is related linearly to the observation:

$$\mathbf{y} = \mathbf{H}\mathbf{x}. \tag{3.2}$$

If a single observation $\mathbf{y}$ is available, and we have no a priori knowledge of the state, what is the "best" estimate of the state that can be achieved? Clearly, the notion of "best" estimate is relative

to some criterion against which the quality of the estimate is to be assessed. Intuitively, we want to get the projection of the state into observation space as "close" as possible to the observation, i.e. we want to minimise the residual $\mathbf{y} - \mathbf{H}\mathbf{x}$.

Several definitions of the distance between two vectors exist. For instance, the distance between $\mathbf{y}$ and $\mathbf{H}\mathbf{x}$ could be expressed as the absolute value between the two quantities $|\mathbf{y} - \mathbf{H}\mathbf{x}|$ (L-1 norm). It could also be expressed as the Euclidean distance (or L-2 norm), i.e. $||\mathbf{y} - \mathbf{H}\mathbf{x}|| = \sqrt{(\mathbf{y} - \mathbf{H}\mathbf{x})^{\mathrm{T}}(\mathbf{y} - \mathbf{H}\mathbf{x})}$ or using the L-inf norm, i.e. $\max |y_i - \mathbf{h}_i^{\mathrm{T}}\mathbf{x}|$ (where i denotes the i-th component of the vector, and $\mathbf{h}_i$ is the i-th row of the matrix $\mathbf{H}$). The choice preferred in most situation is the square of the Euclidean distance, since it is differentiable everywhere.

The estimate of $\mathbf{x}$ which minimises the squared Euclidean distance (or misfit) is known as the Least Squares Estimate, as it minimises the square distance to the observation. For a single observation, the misfit to the observation is given by:

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{y} - \mathbf{H}\mathbf{x})^{\mathrm{T}}(\mathbf{y} - \mathbf{H}\mathbf{x}), \tag{3.3}$$

$$= \frac{1}{2}\sum_{i=1}^{N}(y_i - \mathbf{h}_i^{\mathrm{T}}\mathbf{x})^2, \tag{3.4}$$

where N is the dimension of the observation space, y_i the observation along the i-th dimension and $\mathbf{h}_i^{\mathrm{T}}$ the i-th row of $\mathbf{H}$. The factor $\frac{1}{2}$ is introduced for convenience, in order to cancel the factor 2 which appears when differentiating the quadratic form.

The minimum of $J(\mathbf{x})$ is obtained by setting the gradient of $J(\mathbf{x})$ to 0:

$$\nabla J(\mathbf{x}) = -\mathbf{H}^{\mathrm{T}}(\mathbf{y} - \mathbf{H}\mathbf{x}) = 0, \tag{3.5}$$

giving the optimal estimate (in a least squares sense):

$$\boxed{\mathbf{x}^a = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{y}} \tag{3.6}$$

Note that if $\mathbf{H}$ is invertible, this expression reduces to the expected result $\mathbf{x}^a = \mathbf{H}^{-1}\mathbf{y}$.

Linear Least Squares Estimate - Multiple observations

If several observations are available, i.e. $\mathbf{y}_1, ..., \mathbf{y}_M$, the least square estimate becomes the one that minimises the distance to all observations. Equation (3.3) becomes:

$$J(\mathbf{x}) = \frac{1}{2}\sum_{k=1}^{M}(\mathbf{y}_k - \mathbf{H}\mathbf{x})^{\mathrm{T}}(\mathbf{y}_k - \mathbf{H}\mathbf{x}) \tag{3.7}$$

and its gradient is given by:

$$\nabla J(\mathbf{x}) = \sum_{k=1}^{M} -\mathbf{H}^{\mathrm{T}}(\mathbf{y}_k - \mathbf{H}\mathbf{x}), \tag{3.8}$$

$$= M\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{x} - \mathbf{H}^{\mathrm{T}}\left(\sum_{k=1}^{M}\mathbf{y}_k\right). \tag{3.9}$$

Setting the gradient to zero leads to the least squares estimate for multiple observations:

$$\boxed{\mathbf{x}^a = \frac{1}{M}(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\left(\sum_{k=1}^{M}\mathbf{y}_k\right)} \tag{3.10}$$

Note how, in the case where $\mathbf{H}$ is the identity (i.e. we are observing the state directly), the result reduces to the average of the observed values. This confirms the intuition that if one has no knowledge about the nature of the observation error, the best estimate (in a least square sense) is obtained by averaging the observations.

Generalised Linear Least Squares Estimate

A more general expression of the least square estimate can be obtained by using the *energy norm* (Lewis et al., 2006) rather than the L-2 norm in the definition of $J(\mathbf{x})$. For a symmetric positive definite matrix $\mathbf{W}$, the squared energy norm is defined, for any $\mathbf{x}$, by:

$$\| \mathbf{x} \|^2_{\mathbf{W}} = \mathbf{x}^{\mathrm{T}}\mathbf{W}\mathbf{x}. \tag{3.11}$$

Note that in the case where the $\mathbf{W}$ matrix is diagonal, this expression reduces to a weighted sum of squares.

Using the energy norm, we can rewrite $J(\mathbf{x})$ as:

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{y} - \mathbf{H}\mathbf{x})^{\mathrm{T}}\mathbf{W}(\mathbf{y} - \mathbf{H}\mathbf{x}) \tag{3.12}$$

and compute its gradient:

$$\nabla J(\mathbf{x}) = -\mathbf{H}^{\mathrm{T}}\mathbf{W}(\mathbf{y} - \mathbf{H}\mathbf{x}). \tag{3.13}$$

Setting the gradient to 0 yields the generalised least squares estimate of $\mathbf{x}$:

$$\boxed{\mathbf{x}^a = (\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{W}\,\mathbf{y}} \tag{3.14}$$

Typically, the matrix $\mathbf{W}$ provides a means to give more or less weight to the observation, depending on how confident we are in the quality of the observation procedure.

Non-linear Least Squares Estimate

So far, we have assumed that the observation operator was linear, i.e. $h = \mathbf{H}$. However, many applications require the use of a non-linear h. The optimal estimate is then expressed as:

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{y} - h(\mathbf{x}))^{\mathrm{T}}\mathbf{W}(\mathbf{y} - h(\mathbf{x})). \tag{3.15}$$

Generally, a close form expression of the least square estimate cannot be obtained in the case of a non-linear h. A possible alternative approach consists in minimising $J(\mathbf{x})$ with respect to the

state, using standard optimisation techniques. However, if h is smooth in the neighbourhood of a reference state $\mathbf{x}_\mathrm{r}$, it is more efficient to use a first-order Taylor expansion about $\mathbf{x}_\mathrm{r}$:

$$h(\mathbf{x}) \approx h(\mathbf{x}_\mathrm{r}) + \hat{\mathbf{H}}_\mathrm{r}(\mathbf{x} - \mathbf{x}_\mathrm{r}). \tag{3.16}$$

$\hat{\mathbf{H}}_\mathrm{r}$ denotes the Jacobian of h about $\mathbf{x}_\mathrm{r}$, i.e. $(\hat{\mathbf{H}}_\mathrm{r})_{i,j} = \frac{\partial h_i}{\partial \mathbf{x}_j}(\mathbf{x}_\mathrm{r})$. Inserting (3.16) into (3.15) yields:

$$J(\mathbf{x}) \approx \frac{1}{2}\left(\mathbf{y} - h(\mathbf{x}_\mathrm{r}) - \hat{\mathbf{H}}_\mathrm{r}(\mathbf{x} - \mathbf{x}_\mathrm{r})\right)^\mathrm{T} \mathbf{W} \left(\mathbf{y} - h(\mathbf{x}_\mathrm{r}) - \hat{\mathbf{H}}_\mathrm{r}(\mathbf{x} - \mathbf{x}_\mathrm{r})\right) \tag{3.17}$$

and the gradient:

$$\nabla J(\mathbf{x}) \approx -\hat{\mathbf{H}}_\mathrm{r}^\mathrm{T} \mathbf{W} \left(\mathbf{y} - h(\mathbf{x}_\mathrm{r}) - \hat{\mathbf{H}}_\mathrm{r}(\mathbf{x} - \mathbf{x}_\mathrm{r})\right), \tag{3.18}$$

$$\approx -\hat{\mathbf{H}}_\mathrm{r}^\mathrm{T} \mathbf{W} \left(\mathbf{y} - h(\mathbf{x}_\mathrm{r})\right) + \hat{\mathbf{H}}_\mathrm{r}^\mathrm{T} \mathbf{W} \hat{\mathbf{H}}_\mathrm{r}(\mathbf{x} - \mathbf{x}_\mathrm{r}). \tag{3.19}$$

Setting the gradient to zero yields the approximated non-linear least square estimate:

$$\boxed{\mathbf{x}^a \approx \mathbf{x}_\mathrm{r} + \left(\hat{\mathbf{H}}_\mathrm{r}^\mathrm{T} \mathbf{W} \hat{\mathbf{H}}_\mathrm{r}\right)^{-1} \hat{\mathbf{H}}_\mathrm{r}^\mathrm{T} \mathbf{W} \left(\mathbf{y} - h(\mathbf{x}_\mathrm{r})\right)} \tag{3.20}$$

Note that in the case of a linear $h = \hat{\mathbf{H}}_\mathrm{r} = \mathbf{H}$, this result reduces to the result obtained in the linear case.

3.2.2 With background information

We extend the case of a linear, deterministic observation operator $h = \mathbf{H}$ by considering that some background information about the state is available. The estimation problem becomes the search for the optimal state estimate which minimises the departure both from the observation and the background (in a least square sense), as illustrated on Figure 3.1. The background estimate is denoted $\mathbf{x}^b$ throughout this chapter.

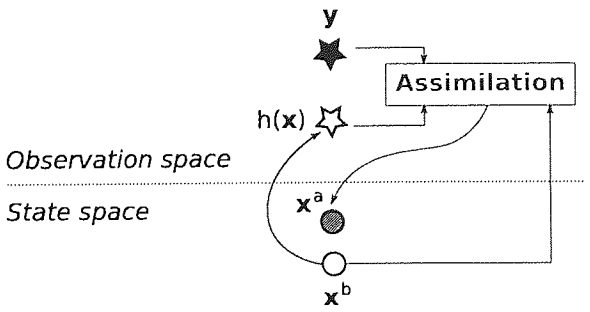


Figure 3.1: Deterministic data assimilation in a static context. The observation $\mathbf{y}$ is used to correct the background state $\mathbf{x}^b$, giving an updated state $\mathbf{x}^a$ (assimilation).

Linear Least Squares Estimate

It is clear that the optimal estimate will depend on our relative confidence in the background and in the observation. In the least squares formulation, the relative confidence in the observation and the prior is expressed by appropriate choice of weighting matrices. Instead of using the notation $\mathbf{W}$ for the weight matrix (as was done in the previous section), we introduce new notations. First, rather than characterising the weight using a symmetric positive definite matrix, we will use inverse matrices with the same properties (we know that the inverse of a symmetric positive definite matrix is also a symmetric positive definite matrix). This is to make the notation consistent with the section dealing with the stochastic approach, as we will see that in the case of Gaussian distributions, the weight matrix is then equivalent to the covariance matrix.

We denote $\mathbf{R}$ the inverse weight matrix for the observations and $\mathbf{B}$ the inverse weight matrix for the background. The least squares estimation problem is thus written as the sum of two parallel estimation problems:

$$J(\mathbf{x}) = J_b(\mathbf{x}) + J_o(\mathbf{x}) \tag{3.21}$$

$$= \frac{1}{2}(\mathbf{x}-\mathbf{x}^b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b) + \frac{1}{2}(\mathbf{y}-\mathbf{H}\mathbf{x})^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\mathbf{x}) \tag{3.22}$$

Note how the relative influences of the observation and the background are controlled by the relative values of the matrices $\mathbf{R}$ and $\mathbf{B}$. For instance, high confidence in the background will be expressed by choosing $\mathbf{B}$ smaller than $\mathbf{R}$, thus putting a heavier penalty on departures from the background (J_b) than on departures from the observation (J_o).

The optimal estimate is obtained by setting the gradient of $J(\mathbf{x})$ to 0:

$$\nabla J(\mathbf{x}) = \nabla J_b(\mathbf{x}) + \nabla J_o(\mathbf{x}) \tag{3.23}$$

$$= \mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b) - \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\mathbf{x}) \tag{3.24}$$

$$= (\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})\mathbf{x} - (\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y} + \mathbf{B}^{-1}\mathbf{x}^b), \tag{3.25}$$

yielding the least square estimate:

$$\mathbf{x}^a = (\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y} + \mathbf{B}^{-1}\mathbf{x}^b). \tag{3.26}$$

A clearer expression of $\mathbf{x}^a$ can be obtained by rewriting it as a displacement from $\mathbf{x}^b$. This is easily done by adding $\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\mathbf{x}^b - \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\mathbf{x}^b$ to the right hand side term in the product and factorising:

$$\boxed{\mathbf{x}^a = \mathbf{x}^b + (\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\mathbf{x}^b).} \tag{3.27}$$

Non-linear Least Squares Estimate

If the observation operator is non-linear, an approximate solution can be derived, following the derivations given in Section 3.2.1. The cost function for the non-linear case is given by:

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\mathbf{x}^b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b) + \frac{1}{2}(\mathbf{y}-h(\mathbf{x}))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-h(\mathbf{x})) \tag{3.28}$$

By using a Taylor expansion about a reference state $\mathbf{x}_{\mathrm{r}}$, we can apply the result from Equation (3.19) to obtain the linearised $\nabla J_o(\mathbf{x})$. The gradient of $J(\mathbf{x})$ is then given by:

$$\nabla J(\mathbf{x}) \approx \mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b) - \hat{\mathbf{H}}_{\mathrm{r}}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-h(\mathbf{x}_{\mathrm{r}})) + \hat{\mathbf{H}}_{\mathrm{r}}^{\mathrm{T}}\mathbf{R}^{-1}\hat{\mathbf{H}}_{\mathrm{r}}(\mathbf{x}-\mathbf{x}_{\mathrm{r}}) \tag{3.29}$$

$$\approx (\hat{\mathbf{H}}_{\mathrm{r}}^{\mathrm{T}}\mathbf{R}^{-1}\hat{\mathbf{H}}_{\mathrm{r}} + \mathbf{B}^{-1})(\mathbf{x}-\mathbf{x}_{\mathrm{r}}) - \hat{\mathbf{H}}_{\mathrm{r}}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-h(\mathbf{x}_{\mathrm{r}})) + \mathbf{B}^{-1}(\mathbf{x}_{\mathrm{r}}-\mathbf{x}^b). \tag{3.30}$$

Setting $\nabla J(\mathbf{x})$ to zero gives the linearised least square estimate:

$$\boxed{\mathbf{x}^a \approx \mathbf{x}_{\mathrm{r}} + (\hat{\mathbf{H}}_{\mathrm{r}}^{\mathrm{T}}\mathbf{R}^{-1}\hat{\mathbf{H}}_{\mathrm{r}} + \mathbf{B}^{-1})^{-1}\left[\hat{\mathbf{H}}_{\mathrm{r}}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-h(\mathbf{x}_{\mathrm{r}})) - \mathbf{B}^{-1}(\mathbf{x}_{\mathrm{r}}-\mathbf{x}^b)\right]} \tag{3.31}$$

Here again, if h is linear, the result reduces to the linear least square estimate.

3.2.3 Variational approach: 3D VAR

The variational method consist in replacing the problem of finding the exact minimum of the cost function (3.22):

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\mathbf{x}^b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b) + \frac{1}{2}(\mathbf{y}-\mathbf{H}\mathbf{x})^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\mathbf{x}) \tag{3.32}$$

with an optimisation problem. Rather than looking for a stationary point by setting the gradient to zero and computing the exact estimate, in a variational context, one looks for an approximate solution which provides a close enough approximation to the actual minimum. This solution is obtained by applying standard optimisation algorithms such as gradient descent or quasi-Newton to (3.22). Introductions to 3D variational methods can be found in Bouttier and Courtier (1999); Kalnay (2003); Lewis et al. (2006).

In the meteorological community, the variational problem is referred to as three dimensional variational (3D VAR) method, the three dimensions being that of the usual Euclidean space. 3D VAR is used operationally in several weather forecasting centres, including the ECMWF (Courtier et al., 1998; Andersson et al., 1998; Rabier et al., 2000), the UK Meteorological Office (Lorenc et al., 2000) and the Canadian Meteorological Centre (Gauthier et al., 1999; Laroche et al., 1999), although it is now being superseded by its successor, 4D VAR (see Section 4.2.3).

Non-linear case

In the case of a non-linear h, the cost function (3.28) needs to be minimised. The gradient of J can be approximated by (3.30) and be used, for instance, in a gradient descent algorithm. However, the linear approximation will only be relevant if h is smooth at the scales considered.

3.3 Stochastic approach

We consider here the stochastic approach to the static assimilation problem described in the previous sections and discuss its treatment in the Bayesian framework from Section 2.3.2. Remember that in a stochastic context, the state is represented by a probability density function $p(\mathbf{x})$, possibly conditioned on the observation. Initially, we assume that some background information about the state is available and is represented by the prior distribution $p(\mathbf{x}^b)$.

The observation is related to the state according to Equation (2.4):

$$\mathbf{y} = h(\mathbf{x}) + \varepsilon(\mathbf{x}). \tag{3.33}$$

The stochastic nature of the error term ε induces a probability distribution (likelihood) for the observation, conditioned on the state: $p(\mathbf{y}|\mathbf{x}^b)$. We are thus looking, in this context, for an updated estimate which takes into account the information provided by the prior and the likelihood, while characterising the resulting uncertainty, as illustrated on Figure 3.2.

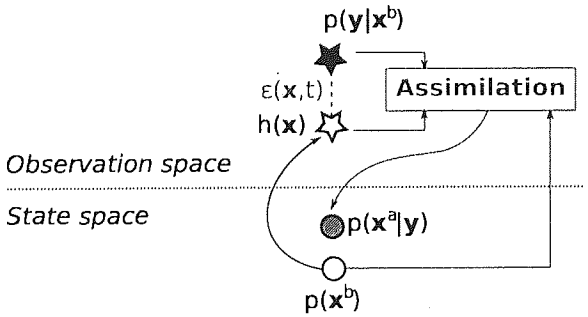


Figure 3.2: Stochastic data assimilation in a static context. The likelihood $p(\mathbf{y}|\mathbf{x}^b)$ is used to correct the prior $p(\mathbf{x}^b)$, giving an updated distribution $p(\mathbf{x}^a|\mathbf{y})$.

As explained in Section 2.3.2, we can use Bayes' rule to update our estimate of $\mathbf{x}$:

$$p(\mathbf{x}^a|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x}^b)\,p(\mathbf{x}^b)}{p(\mathbf{y})}, \tag{3.34}$$

where $p(\mathbf{x}^a|\mathbf{y})$ is the posterior (i.e. updated) probability density function of the state given $\mathbf{y}$ and our background estimate $\mathbf{x}^b$. That is data assimilation is essentially classical Bayesian inference for regression models.

3.3.1 Optimal solution in the linear case

Just as there was several possible "best" estimates in the deterministic case (depending on the notion of distance/norm chosen), there are several possible "optimum" distributions of the state given the observation. A traditional Bayesian treatment might either look for an estimate which maximises the likelihood of the observation (*maximum likelihood* estimate) or for an estimate with maximum probability given both the observation and the prior (*maximum a posteriori* estimate). In the following, we focus on the maximum a posteriori (MAP) approach and show that in some circumstances it provides a stochastic solution equivalent to least square approach discussed in the deterministic case.

Linear Gaussian case

We start by considering the simple case where the observation operator is linear, $h(\mathbf{x}) = \mathbf{H}\mathbf{x}$, and probability density functions are assumed Gaussian:

$$\mathbf{x}^b \sim \mathcal{N}(\bar{\mathbf{x}}^b, \mathbf{B}), \tag{3.35}$$

$$\varepsilon(\mathbf{x}) = \mathbf{y} - \mathbf{H}\mathbf{x} \sim \mathcal{N}(0, \mathbf{R}). \tag{3.36}$$

The background has mean $\bar{\mathbf{x}}^b$ and covariance $\mathbf{B}$. The observation error is assumed to be Gaussian white noise with covariance matrix $\mathbf{R}$.

We know already that for a Gaussian likelihood, a Gaussian prior leads, through Bayes' rule, to a Gaussian posterior (Bernardo and Smith, 1994), with mean $\bar{\mathbf{x}}^a$ and covariance matrix $\mathbf{P}^a$ to be determined:

$$\mathbf{x}^a \sim \mathcal{N}(\bar{\mathbf{x}}^a, \mathbf{P}^a). \tag{3.37}$$

We note that the estimate which maximises the posterior distribution also minimises the negative logarithm of that distribution, the two problems being equivalent due to the monotonicity of the logarithm function. The minimisation of the negative log-posterior, however, is made easier by involving computations with quadratic forms rather than exponentials. We will thus pursue this approach (see also Jazwinski (1970); Lorenc (1986); Lewis et al. (2006) for equivalent derivations).

Taking the negative logarithm of the posterior, and substituting using Bayes rule, we get:

$$-\ln p(\mathbf{x}|\mathbf{y}) = -\ln p(\mathbf{x}) - \ln p(\mathbf{y}|\mathbf{x}) + \ln p(\mathbf{y}). \tag{3.38}$$

The term in $\mathbf{y}$ is a constant with respect to $\mathbf{x}$, and can be discarded since it does not affect the minimisation. If we now recall that the expression of the Gaussian distribution is given by:

$$\mathcal{N}(\mathbf{x} \mid \bar{\mathbf{x}}, \mathbf{P}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\mathbf{P}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\bar{\mathbf{x}}-\mathbf{x})^{\mathrm{T}}\mathbf{P}^{-1}(\bar{\mathbf{x}}-\mathbf{x})} \tag{3.39}$$

where $\bar{\mathbf{x}}$ denotes the mean, $\mathbf{P}$ the covariance matrix and D the dimension of the state, we can expand (3.38) and we obtain, after the constant terms have been discarded:

$$-\ln p(\mathbf{x}|\mathbf{y}) \propto \frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}^b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}-\bar{\mathbf{x}}^b)+\frac{1}{2}(\mathbf{y}-\mathbf{H}\mathbf{x})^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\mathbf{x}). \tag{3.40}$$

The expression on the right hand side is identical to the Linear Least Squares formulation (3.22) which is also the cost function in the variational approach. In other words, Bayesian inference in the particular case of a linear $\mathbf{H}$ and Gaussian distributions is the stochastic equivalent of the Linear Least Squares Estimate.

It follows, by application of the result in (3.27), that the best estimate is given by:

$$\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^b + (\mathbf{B}^{-1}+\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\bar{\mathbf{x}}^b). \tag{3.41}$$

The covariance matrix $\mathbf{P}^a$ is easily obtained from (3.41):

$$\mathbf{P}^a = \mathbf{B}-(\mathbf{B}^{-1}+\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\mathbf{B}. \tag{3.42}$$

A more tractable expression for $\bar{\mathbf{x}}^a$ and $\mathbf{P}^a$ makes use of the Kalman gain matrix (Kalman, 1960): $\mathbf{K} = (\mathbf{B}^{-1}+\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}$. Multiplying $\mathbf{K}$ on the right hand side by the identity matrix $\mathbf{I} = (\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}+\mathbf{R})(\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}+\mathbf{R})^{-1}$ leads, after simplification, to the standard set of equations for the optimal estimator:

$$\mathbf{x}^a \sim \mathcal{N}(\bar{\mathbf{x}}^a, \mathbf{P}^a), \tag{3.43}$$

with:

$$\boxed{\begin{aligned}\bar{\mathbf{x}}^a &= \bar{\mathbf{x}}^b + \mathbf{K}(\mathbf{y}-\mathbf{H}\mathbf{x}^b),\\ \mathbf{P}^a &= (\mathbf{I}-\mathbf{K}\mathbf{H})\mathbf{B},\\ \mathbf{K} &= \mathbf{B}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}+\mathbf{R})^{-1}.\end{aligned}} \tag{3.44}$$

Non-linear case

If h is non-linear, the gain matrix $\mathbf{K}$ can be approximated by a linearised $\hat{\mathbf{K}}$ computed using the Taylor expansion in (3.16). $\hat{\mathbf{K}}$ is then used in the update equations for the mean and covariance:

$$\boxed{\begin{aligned}\bar{\mathbf{x}}^a &= \bar{\mathbf{x}}^b + \hat{\mathbf{K}}(\mathbf{y}-\mathbf{H}\mathbf{x}^b),\\ \mathbf{P}^a &= (\mathbf{I}-\hat{\mathbf{K}}\hat{\mathbf{H}}_{\mathrm{r}})\mathbf{B},\\ \hat{\mathbf{K}} &= \mathbf{B}\hat{\mathbf{H}}_{\mathrm{r}}^{\mathrm{T}}(\hat{\mathbf{H}}_{\mathrm{r}}\mathbf{B}\mathbf{H}_{\mathrm{r}}^{\mathrm{T}}+\mathbf{R})^{-1}.\end{aligned}} \tag{3.45}$$

3.3.2 Further considerations

Specification of the B matrix

A common difficulty is the specification of a good background covariance $\mathbf{B}$. There are two major methods to estimate the background covariance: the first is based on the correlation between two

forecasts from different lead times, the second is based on ensemble methods (see Section 4.3.4 for a description of ensemble methods).

All these methods are discussed in the context of dynamic data assimilation and might make use of notions which will be explicited in the next chapter. The reader only needs to know that in the dynamic case, the background is usually a forecast from some previous estimate of the state.

In the first method, proposed by Parrish and Derber (1992), the $\mathbf{B}$ matrix is estimated by the correlation between two forecasts from different past origins, e.g. $\mathbf{B} \approx \langle (\mathbf{x}_t^{-48h} - \mathbf{x}_t^{-12h})^{\mathrm{T}} (\mathbf{x}_t^{-48h} - \mathbf{x}_t^{-12h}) \rangle$. This method, referred to as NMC (after the Canadian National Meteorological Centre), has been used in many operational weather forecasting centres and is the favourite choice in data assimilation, because of its simplicity.

The NMC method, however, presents several drawbacks. As mentionned in (Fisher, 2003), poorly observed regions might have very similar forecasts, and see their background covariance underestimated as a result. Another issue is the use of forecasts at much longer lead times (12h and 48h) than those used to generate the background estimate (Fisher, 2003), leading to potential inconsistencies.

An alternative approach is the use of ensemble methods. Ensemble methods rely on propagating an ensemble of estimates (rather than a single state) and use this predicted ensemble to derive approximate statistics (mean and covariance) of the background. The ensemble is usually generated by perturbating the state using noise drawn from the background error's estimated distribution (i.e. in the deterministic case, Gaussian white noise with covariance $\mathbf{P}^a$). This approach is used operationally in the ECMWF variational assimilation system (Fisher, 2003).

The use of ensemble methods is an ad-hoc solution to estimate the propagation of the state's distribution. A more robust framework in which the state's distribution is sequentially estimated is the one provided by filtering methods (see Section 4.3). Because filtering methods are stochastic and thus track the temporal evolution of the state's probability density function, they can easily be used to provide an estimate of the background covariance. Applications of filtering methods can be found in Hamill and Snyder (2000), where a blend of the NMC method and an Ensemble Kalman Filter (see Section 4.3.4) are used to generate the $\mathbf{B}$ matrix. Buehner (2005); Buehner et al. (2005) also use an Ensemble Kalman Filter, this time in replacement of the NMC method.

Other alternative background covariance representation models have been proposed, most as extensions to the existing NMC method (Dee and Gaspari, 1996; Desroziers, 1997; Fisher, 2003, 2006). Further detail on error specification in data assimilation can be found in Lindskog (2007).

3.4 Summary of this chapter

This chapter presented several methods to address the data assimilation problem in the case where no dynamics (i.e. no model) are involved. Least Squares Estimation was introduced as an optimal solution (minimising the variance of the departures to the observations) in the case where the observation is related linearly to the state. The solution is the minimum of a cost function expressing the trade off between the background estimate and the observations.

An approximation in the non-linear case was given, which relies on a Taylor expansion of the observation operator about the state estimate. An alternative approach based on variational methods was presented. This method (3D VAR) replaces the computation of the optimal estimate with a minimisation problem. Namely, an approximation to the minimum of the cost function is obtained through some optimisation method (gradient-descent, quasi-Newton, etc.). The non-linear case is discussed.

The transposition of the Least Squares Estimate to the stochastic context leads, in the Gaussian case, to a set of equations for the update of the state's mean and covariance. Static data assimilation provides the basic methods used in conjunction with a dynamic model in standard data assimilation. Extensions to the non-Gaussian case have been omitted, as they will be discussed in the context of dynamic data assimilation.

The next chapter builds upon the methods introduced and discusses their application to the problem of estimating the state no only in state (or observation) space but also in time.

4 Dynamic data assimilation

CONTENTS

4.1 Foreword

Chapter 3 discussed several mechanisms to estimate the state given static observations, both in a deterministic and stochastic contexts. This chapter places the static estimation problem into a dynamic context, where the state is propagated forward in time and observations become available at given time intervals.

The observations can be assimilated in two ways. Either one at a time, to update the latest estimate of the state, or several at a time, in which case a best trajectory of the state over a given time window is sought. The first approach is called *filtering* while the second is referred to as *smoothing*. Figure 4.1 illustrates the filtering principle. In that case, dynamic data assimilation can be seen as a succession of static data assimilation steps in between which the state is propagated using the model. Figure 4.2 illustrates the smoothing approach.

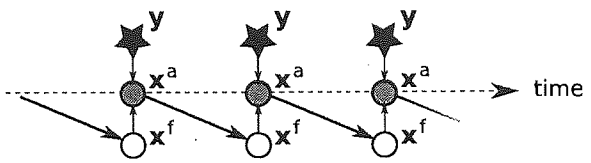


Figure 4.1: Dynamic data assimilation, filtering approach – The state is propagated forward in time (prediction step) using the model (thick arrow). When an observation $\mathbf{y}$ is available (star), the predicted state $\mathbf{x}^f$ is updated as in static data assimilation (assimilation step) giving a new analysis $\mathbf{x}^a$ (thin arrows). The process is then repeated.

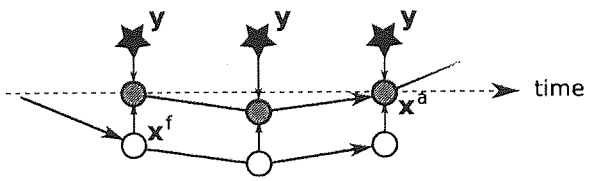


Figure 4.2: Dynamic data assimilation, smoothing approach – The state is propagated forward in time (prediction step) using the model (thick arrow). The optimal estimate is sought by computing the trajectory of the state which minimises the distances to all observations (stars) within a given time window (the window covers 3 observations in this example).

Both the dynamic model m and the observation operator h are assumed known. If any model parameter needs to be estimated in time, it is incorporated in the state vector $\mathbf{x}$, without other changes to the formulation required.

Notations

In order to distinguish between the predicted state and the assimilated (or updated) state, we denote them by $\mathbf{x}^f$ and $\mathbf{x}^a$ respectively (the 'f' and 'a' superscripts come from the meteorological

literature, where the predicted state is called the "forecast" and the updated state is called the "analysis").

Chapter outline

We follow a pattern similar to that of the previous chapter. Section 4.2 looks at the application of the Least Square and variational methods introduced in Chapter 3 to the dynamic case. Section 4.3 extends the discussion to the problem of stochastic state estimation.

4.2 Deterministic approach

In this section, the model and the observation operator are related to the state as follows:

$$\mathbf{x}_t = m_t(\mathbf{x}_{t-1}) \tag{4.1}$$

$$\mathbf{y}_t = h_t(\mathbf{x}_t). \tag{4.2}$$

Model error is assumed constant in time and represented by a background covariance matrix $\mathbf{B}$. Similarly, observation errors are represented by the covariance matrix $\mathbf{R}$.

4.2.1 Dynamic Least Square

If the observation operator is linear: $h_t = \mathbf{H}_t$ and one observation is used at a time to update the (time-evolving) state, the problem of deterministic data assimilation reduces to the following iterative approach:

0. Initialise the state to some background estimate:

$$\mathbf{x}_t^a = \mathbf{x}_0 \tag{4.3}$$

1. **Prediction step:** Propagate the state forward in time using (4.1):

$$\mathbf{x}_t^f = m_t(\mathbf{x}_{t-1}^a) \tag{4.4}$$

2. **Assimilation step:** Once an observation becomes available, use (3.27) to update the state:

$$\mathbf{x}_t^a = \mathbf{x}_t^f + (\mathbf{B}^{-1} + \mathbf{H}_t^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}_t)^{-1}\mathbf{H}_t^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}_t + \mathbf{H}_t\mathbf{x}_t^f). \tag{4.5}$$

Steps 1 and 2 are applied iteratively as the state is propagated in time. This method ensures the model remains consistent with the true process, and is optimal (in a least square sense) in the case where h is linear.

In the case of a non-linear h, the first-order Taylor expansion discussed in the previous chapter can be applied, and the approximate optimal state is given by (3.27), where the background state $\mathbf{x}^b$ is replaced by the predicted state $\mathbf{x}_t^f$.

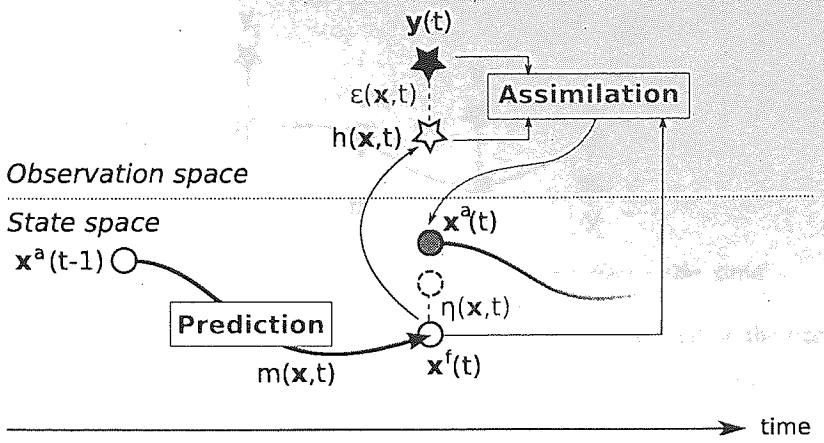


Figure 4.3: Deterministic data assimilation in a dynamic context. The state estimate is propagated forward in time and updated given an observation.

4.2.2 3D variational assimilation

Dynamic 3D VAR follows essentially the same procedure as Dynamic Least Squares, except the update step (step 2) is performed using the static 3D VAR algorithm (Section 3.2.3). In the case of a linear $\mathbf{H}$, the updated state minimises the cost function:

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x}_t - \mathbf{x}_t^f)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}_t - \mathbf{x}_t^f) + \frac{1}{2}(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t)^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t) \tag{4.6}$$

which is identical to (3.22) except the background has been replaced by the predicted state $\mathbf{x}^f$. In the case of a non linear h, a first order Taylor approximation can be applied to compute the gradient of $J(\mathbf{x})$.

4.2.3 4D variational assimilation

Overview

The 3D variational algorithm allows us to update the state at time intervals where observations are available. However, the optimisation does not directly involve the model, other than through the predicted state $\mathbf{x}^f$. 4D variational assimilation (4D VAR) extends 3D VAR in a way that allows the dynamic nature of the problem to be better addressed.

Rather than estimate the state at a single point in time, in 4D VAR the estimation problem is reformulated as a smoothing problem. Namely, one looks for the best (in a least square sense) trajectory of the state given the last N observations. The addition of the dynamic component into the 3D algorithm, as will be described below, resulted in the method being called 4D VAR (time being the 4th dimension).

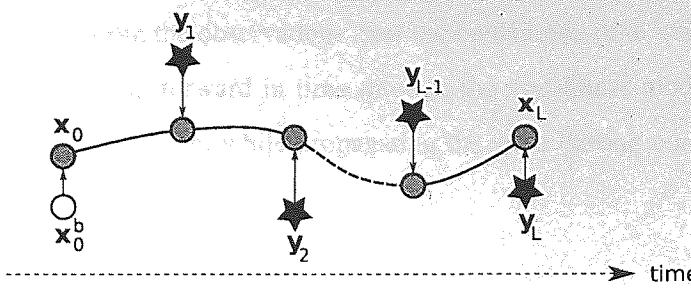


Figure 4.4: 4D VAR (strong constraint) – The optimal trajectory is conditioned on the background $\mathbf{x}_0^b$, the model and the observations (stars). Full circle denote the intermediate forecast states $\mathbf{x}_k$.

Strong constraint 4D VAR

In its strong constraint formulation, 4D VAR does not take model error into account. The model is assumed perfect and the trajectory of the state over the time window is not perturbed. In other words, the model acts as a strong constraint in the minimisation problem. Applications of strong constraint 4D VAR to weather forecasting problems can be found in (Lewis and Derber, 1985; Cram and Kaplan, 1985; Le Dimet and Talagrand, 1986; Zupanski, 1993; Courtier et al., 1994; Ide et al., 1997; Rabier et al., 2000; Mahfouf and Rabier, 2000; Klinker et al., 2000; Rawlins et al., 2007).

Recall from 3D VAR that variational assimilation is based on the minimisation of a cost function expressing a trade-off between a background estimate and an observation:

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x}_t - \mathbf{x}_t^f)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}_t - \mathbf{x}_t^f) + \frac{1}{2}(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t)^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}_t - \mathbf{H}_t\mathbf{x}_t) \tag{4.7}$$

4D VAR, in a similar fashion, seeks the value of the state that minimises the departure from the background term not given a single observation, but given a set of sequential observations. To make explicit these notions, consider a system with measurements occurring at times $1, 2, \ldots, t-1, t$. The assimilation is performed over a time window of length L, spanning from time $t-L$ to current time t. To simplify the indexing, we will index (discrete) time from the start of the time window in the following, so that $\mathbf{x}_0 = \mathbf{x}_{t-L}, \mathbf{x}_1 = \mathbf{x}_{t-L+1}, \ldots, \mathbf{x}_L = \mathbf{x}_t$. The 4D variational cost function can then be written:

$$J(\mathbf{x}_{0:L}) = J_b(\mathbf{x}_0) + J_o(\mathbf{x}_{0:L}) \tag{4.8}$$

with:

$$J_b(\mathbf{x}_0) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) \tag{4.9}$$

$$J_o(\mathbf{x}_{0:L}) = \frac{1}{2}\sum_{k=0}^{L}(\mathbf{y}_k - h(\mathbf{x}_k))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}_k - h(\mathbf{x}_k)) \ . \tag{4.10}$$

The $J_b(\mathbf{x}_t)$ term minimises the departure from the background term $\mathbf{x}_0^b$ at the beginning of the time window (obtained from our previous estimation of the state). The $J_o(\mathbf{x}_t)$ term minimises the

departure of the trajectory from the observations over the considered time window. The $\{\mathbf{x}_k\}_{k=0:t}$ are obtained by propagating $\mathbf{x}_0$ forward in time through the non-linear model. Remember that no model error is taken into account while propagating the state (strong constraint). Figure 4.4 illustrates this process.

Minimising $J(\mathbf{x}_{0:L})$

In order to minimise $J(\mathbf{x}_t)$, one wants to be able to apply some optimisation method. Newton methods or gradient descent methods require the gradient of $J(\mathbf{x}_t)$. The gradient of $J_b(\mathbf{x}_t)$ is straightforward to compute:

$$\nabla J_b(\mathbf{x}_0) = \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) \tag{4.11}$$

The gradient of $J_o(\mathbf{x}_{0:L})$, however, can be problematic in that for a non-linear model and a non-linear observation operator, $J_o(\mathbf{x}_{0:L})$ is no longer a quadratic function in $\mathbf{x}$. To overcome this problem, one can resort to linearisation.

Let us denote $m_{t_1:t_2}$ the model operator propagating $\mathbf{x}$ from t_1 to t_2, and assume by definition that $m_{t:t}$ is the identity operator. Note that the notations m_t and $m_{t-1:t}$ are equivalent. The model can be applied sequentially, so that:

$$\mathbf{x}_{t_2} = m_{t_1:t_2}(\mathbf{x}_{t_1}) \tag{4.12}$$

$$= m_{t_2}(m_{t_2-1}(\ldots m_{t_1+1}(\mathbf{x}_{t_1})\ldots)). \tag{4.13}$$

For a given time k in the time window, the departure from the observation can be approximated using a Taylor expansion about the background state:

$$\mathbf{y}_k - h(\mathbf{x}_k) = \mathbf{y}_k - h(m_{0:k}(\mathbf{x}_0)) \tag{4.14}$$

$$\approx \mathbf{y}_k - h(m_{0:k}(\mathbf{x}_0^b)) - \hat{\mathbf{H}}_k\hat{\mathbf{M}}_{0:k}(\mathbf{x}_0 - \mathbf{x}_0^b), \tag{4.15}$$

where $\hat{\mathbf{H}}_k$ and $\hat{\mathbf{M}}_{0:k} = \hat{\mathbf{M}}_k\hat{\mathbf{M}}_{k-1}\ldots\hat{\mathbf{M}}_1$ are respectively the tangent linear observation operator evaluated about $\mathbf{x}_k$ and the tangent linear model, evaluated about every observation time in the window.

An estimate of the gradient of J_o can be derived from (4.15):

$$\nabla J_o(\mathbf{x}_{0:L}) = -\sum_{k=0}^{L}\hat{\mathbf{M}}_{0:k}^{\mathrm{T}}\hat{\mathbf{H}}_k^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}_k - h(\mathbf{x}_k)) \tag{4.16}$$

$$= -\sum_{k=0}^{L}\hat{\mathbf{M}}_1^{\mathrm{T}}\ldots\hat{\mathbf{M}}_k^{\mathrm{T}}\hat{\mathbf{H}}_k^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}_k - h(\mathbf{x}_k)) \tag{4.17}$$

This expression can be factorised into a more computationally efficient form as follows, where we use the notation $\mathbf{d}_k = \mathbf{R}^{-1}(\mathbf{y}_k - h(\mathbf{x}_k))$:

$$-\nabla J_o(\mathbf{x}_{0:L}) = \hat{\mathbf{H}}_0^{\mathrm{T}}\mathbf{d}_0 + \hat{\mathbf{M}}_0^{\mathrm{T}}\left[\hat{\mathbf{H}}_1^{\mathrm{T}}\mathbf{d}_1 + \hat{\mathbf{M}}_1^{\mathrm{T}}\left[\cdots\right.\right. \\ \left.\left. \cdots + \hat{\mathbf{M}}_{L-2}^{\mathrm{T}}\left[\hat{\mathbf{H}}_{L-1}^{\mathrm{T}}\mathbf{d}_{L-1} + \hat{\mathbf{M}}_{L-1}^{\mathrm{T}}\hat{\mathbf{H}}_L^{\mathrm{T}}\mathbf{d}_L\right]\cdots\right]\right]. \tag{4.18}$$

Equation (4.18) can also be expressed as a backward recursive problem:

$$-\nabla J_o(\mathbf{x}_{0:L}) = \nabla \mathcal{J}_o(\mathbf{x}_{0:L}) \tag{4.19}$$

$$\nabla \mathcal{J}_o(\mathbf{x}_{L:L}) \triangleq \hat{\mathbf{H}}_L^{\mathrm{T}}\mathbf{d}_L \tag{4.20}$$

$$\nabla \mathcal{J}_o(\mathbf{x}_{k:L}) \triangleq \hat{\mathbf{H}}_k^{\mathrm{T}}\mathbf{d}_k + \hat{\mathbf{M}}_k^{\mathrm{T}}\nabla \mathcal{J}_o(\mathbf{x}_{k+1:L}) \tag{4.21}$$

where the symbol $\triangleq$ means "is defined as".

Once the optimal trajectory has been obtained through minimisation of $J(\mathbf{x})$, the model is propagated forward until a new observation becomes available at time $L+1$. The optimisation procedure is then applied again, this time with a shifted time window covering the time range $[1, L+1]$.

Outer/Inner loops

The optimisation step typically involves an *outer loop* inside which the tangent linear model is computed about the state and the state is optimised using an *inner loop*. The outer loop ensures the tangent linear is updated as the state is optimised. Figure 4.5 details this optimisation step.

0. Initialise the state to its current estimate: $\hat{\mathbf{x}} = \mathbf{x}_0^b$ ($\hat{\mathbf{x}}$ denotes the state being optimised)
1. *Outer loop:*
   (a) Compute the tangent linear model about $\hat{\mathbf{x}}$: $\hat{\mathbf{M}}_{0:L}$
   (b) *Inner Loop:*
   Minimise $J(\hat{\mathbf{x}})$ using $\hat{\mathbf{M}}_{0:L}$
2. Set $\mathbf{x}^a = \hat{\mathbf{x}}$.

Figure 4.5: 4D VAR optimisation algorithm

Weak constraint 4D VAR

The weak constraint formulation of 4D VAR (Gustafsson, 1992; Zupanski, 1997; Tremolet, 2006, 2007) addresses the lack of model error in strong constraint 4D VAR. Assuming an imperfect

model with model error as introduced in Section 2.2.2:

$$\mathbf{x}_t = m_t(\mathbf{x}_{t-1}) + \eta_t, \tag{4.22}$$

the augmented cost function is written:

$$J(\mathbf{x}_{0:L}, \eta_{1:L}) = J_b(\mathbf{x}_0) + J_o(\mathbf{x}_{0:L}, \eta_{1:L}) + J_m(\eta_{1:L}) \tag{4.23}$$

with:

$$J_b(\mathbf{x}_0) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^b)^{\mathrm{T}} \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^b) \tag{4.24}$$

$$J_o(\mathbf{x}_{0:L}, \eta_{1:L}) = \frac{1}{2} \sum_{k=0}^{L} (\mathbf{y}_k - h(\mathbf{x}_k))^{\mathrm{T}} \mathbf{R}^{-1} (\mathbf{y}_k - h(\mathbf{x}_k)) \tag{4.25}$$

$$J_m(\eta_{1:L}) = \frac{1}{2} \sum_{k=1}^{L} \eta_k^{\mathrm{T}} \mathbf{Q}_k \eta_k. \tag{4.26}$$

The propagation of the state from time 0 to time k, previously given by (4.13), is now given by:

$$\mathbf{x}_k = m_k(m_{k-1}(\ldots m_1(\mathbf{x}_0) + \eta_1 \ldots) + \eta_{k-1}) + \eta_k \tag{4.27}$$

The propagation of the state can be expressed using the tangent linear model (for a non-linear model):

$$\mathbf{x}_k = m_{0:k}(\mathbf{x}_0) \tag{4.28}$$

$$\approx m_{0:k}(\mathbf{x}_0^b) + \hat{\mathbf{M}}_{0:k}(\mathbf{x}_0 - \mathbf{x}_0^b) + \sum_{i=1}^{k} \hat{\mathbf{M}}_{1:i} \eta_i. \tag{4.29}$$

Using this expression in (4.10) allows us to derive the gradient of J_o with respect to $\mathbf{x}$:

$$\begin{aligned} \nabla_{\mathbf{x}} J_o(\mathbf{x}_{0:L}, \eta_{1:L}) &= -\sum_{k=0}^{L} \left(\hat{\mathbf{M}}_{0:k} + \sum_{i=1}^{k} \hat{\mathbf{M}}_{1:i} \eta_i \right)^{\mathrm{T}} \hat{\mathbf{H}}_k^{\mathrm{T}} \mathbf{d}_k \\ &= -\sum_{k=0}^{L} \hat{\mathbf{M}}_{0:k}^{\mathrm{T}} \hat{\mathbf{H}}_k^{\mathrm{T}} \mathbf{d}_k - \sum_{k=1}^{L} \eta_k^{\mathrm{T}} \sum_{i=k}^{L} \hat{\mathbf{M}}_{i:k}^{\mathrm{T}} \hat{\mathbf{H}}_k^{\mathrm{T}} \mathbf{d}_k. \end{aligned} \tag{4.30}$$

Similarly, the gradient of J with respect to η evaluates to:

$$\nabla_{\eta} J(\mathbf{x}_{0:L}, \eta_{1:L}) = \nabla_{\eta} J_o(\mathbf{x}_{0:L}, \eta_{1:L}) + \nabla_{\eta} J_m(\eta_{1:L}) \tag{4.31}$$

$$= -\sum_{k=1}^{L} \sum_{i=k}^{L} \hat{\mathbf{M}}_{i:k}^{\mathrm{T}} \hat{\mathbf{H}}_k^{\mathrm{T}} \mathbf{d}_k + \sum_{k=1}^{L} \mathbf{Q}_k^{-1} \eta_k^{\mathrm{T}} \tag{4.32}$$

Stochastic extension: Path sampling

(Apte et al., 2008) recently proposed a stochastic extension to 4D VAR based on path sampling techniques (Hairer et al., 2005; Apte et al., 2007). The method involves sampling from the posterior by solving a Langevin equation or using a Hybrid Markov Chain Monte Carlo approach.

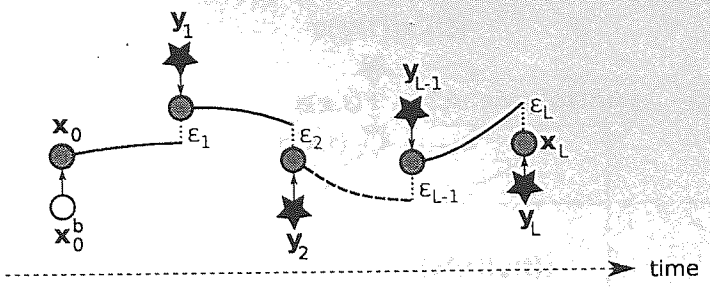


Figure 4.6: 4D VAR (weak constraint) – The optimal trajectory is conditioned on the background $\mathbf{x}_0^b$, the model, the model error and the observations (stars). Full circle denote the intermediate forecast states $\mathbf{x}_k$, while dotted segments represent the model error terms η_k.

The framework provided allows to estimate the smoothing distribution in a fully stochastic, non-Gaussian fashion. The method provides a promising alternative to standard 4D VAR and Kalman filters (next section) in cases where the Gaussian assumption is poor. The computational burden of the method, however, might be a limit to its application in high dimension.

Relationship with Kalman filters

It has long been know that for a linear model and a linear observation operator, the variational approach and the Kalman filter provide equivalent formulations of the same problem (Jazwinski, 1970) and thus yield equivalent results. More recently, Kalman filters and VAR methods have been compared on non-linear models. Li and Navon (2001); Fisher et al. (2005) discuss the equivalence of weak constraint 4D VAR with the Kalman smoother, and show that 4D VAR outperforms the Extended Kalman Filter when used with a sufficient long time window.

4.3 Stochastic approach

In this section, we look at the stochastic treatment of data assimilation in the dynamic context. The model and observation are related to the state according to:

$$\mathbf{x}_t = m_t(\mathbf{x}_{t-1}) + \eta_t, \tag{4.33}$$

$$\mathbf{y}_t = h_t(\mathbf{x}_t) + \varepsilon_t. \tag{4.34}$$

The generic formulation of the problem has been outlined in Section 2.3.2. Basically, one is interested in evaluating the joint probability density function of the state given all observations up to and including a given time t: $p(\mathbf{X}_t|\mathbf{Y}_t)$.

For many applications, however, knowing the marginal probability density function of the state at time t is sufficient, and one does not actually need the full joint distribution of $\mathbf{x}$. The marginal

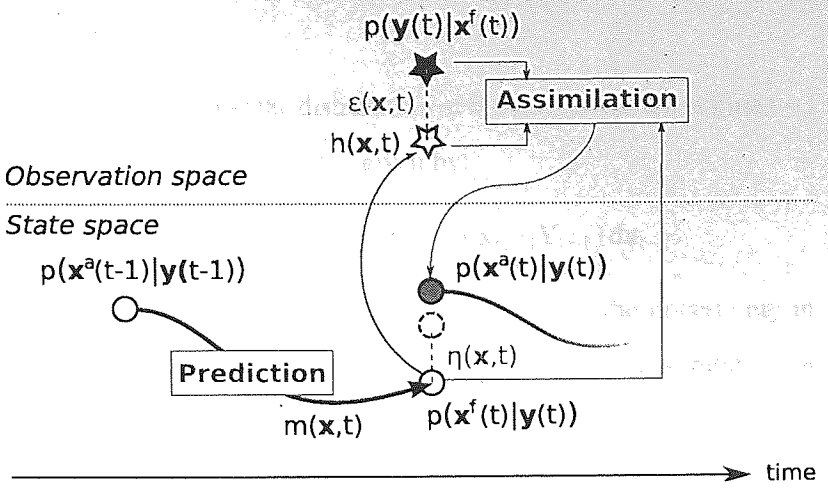


Figure 4.7: Stochastic data assimilation (filtering) in a dynamic context. The state's distribution is propagated forward in time and updated given the observation's likelihood.

distribution can be obtained by integrating the joint distribution over all but the last state estimates:

$$p(\mathbf{x}_t|\mathbf{Y}_t) = \int p(\mathbf{X}_t|\mathbf{Y}_t)\, d\mathbf{X}_{t-1}. \tag{4.35}$$

The marginal distribution can be estimated in two ways. If one is only interested in the marginal distribution of the state conditioned on the latest observation, i.e. $p(\mathbf{x}_t|\mathbf{y}_t)$ the data assimilation problem becomes known as *filtering*. If on the other hand, one wants to estimate the state given all (or a subset of all) previous observations, i.e. $p(\mathbf{x}_t|\mathbf{Y}_t)$, then the problem is known as *smoothing*. Figures 4.1 and 4.2 give a simple overview of the filtering and smoothing approaches respectively. Figure 4.7 illustrates the filtering principle in more detail.

If the model and the observation operator are linear, and if all distributions are Gaussian, then the Kalman filter (Kalman, 1960) provides an optimal (variance minimising) solution to the filtering problem. If the operators are non-linear, sub-optimal methods can be derived. The Extended Kalman Filter (Jazwinski, 1970; Maybeck, 1979) and the Ensemble Kalman Filter (Evensen, 1994) provide respectively a linearised and a Monte Carlo approximations to the Kalman Filter. Another Monte-Carlo approach, the Particle Filter (Doucet et al., 2001), allows the Gaussian assumption to be relaxed. These filtering methods are discussed in the remaining of this section. Smoothing methods are not addressed in this work.

4.3.1 Bayesian formulation: the filtering case

We discuss here the formulation of the Bayesian framework (Section 2.3.2) for the particular case of filtering. The model is assumed Markovian and the observations conditionally uncorrelated in time.

Propagation of the state

If $p(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and the distribution of the state is known at time $t-1$: $p(\mathbf{x}_{t-1}|\mathbf{Y}_{t-1})$, then the predicted distribution of the state is given by:

$$p(\mathbf{x}_t|\mathbf{Y}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1}|\mathbf{Y}_{t-1})\, d\mathbf{x}_{t-1}. \tag{4.36}$$

The state's predicted distribution is obtained by integrating out the uncertainty in $\mathbf{x}_{t-1}$. The expression of $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ depends on the model (Equation 2.5) and the assumptions on model error.

Update of the state

Given an observation $\mathbf{y}_t$ with likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$, the posterior distribution of the state is obtained by applying Bayes' rule (2.11):

$$p(\mathbf{x}_t|\mathbf{Y}_t) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)\, p(\mathbf{x}_t|\mathbf{Y}_{t-1})}{p(\mathbf{Y}_t)}. \tag{4.37}$$

In the case of Gaussian distributions and linear model and observation operator, the filtering posterior can be derived exactly, leading the the Kalman Filter algorithm. If either the distributions are non-Gaussian or the operators are non-linear, various approximations can be used, which lead to several filtering algorithms. These algorithms are detailed in the remaining of this chapter.

4.3.2 Kalman Filter

The Kalman Filter (Kalman, 1960; Jazwinski, 1970; Maybeck, 1979; Rhodes, 1971) is a filtering algorithm which, under certain assumptions given below, provides an optimal least square solution to the problem of dynamic state estimation. The approach is merely a probabilistic formulation of the least square method discussed in Sections 4.2.1 and 3.3.1.

The solution given by the Kalman Filter (KF) is optimal in the case where:

1. the model $\mathbf{M}_t$ and observation operator $\mathbf{H}_t$ are linear,
2. the errors can be represented by Gaussian white noise: $\eta_t \sim \mathcal{N}(0, \mathbf{Q}_t)$; $\varepsilon_t \sim \mathcal{N}(0, \mathbf{R}_t)$,
3. the state has a Gaussian distribution: $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P})$, where $\bar{\mathbf{x}}$ denotes the mean state and $\mathbf{P} = \langle (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^{\mathrm{T}} \rangle$ is the covariance matrix.

The filter is initialised by setting $p(\mathbf{x}_0^a) = p(\mathbf{x}_0) = \mathcal{N}(\mathbf{x} \mid \bar{\mathbf{x}}_0, \mathbf{P}_0)$. The initial distribution is defined so as to reflect our initial knowledge (or lack of knowledge) about the system of interest, before any observation has been assimilated.

Once initialised, the KF follows the sequential two-step procedure described in 4.2.1. The *prediction step* propagates the state's distribution forward in time, while the *assimilation step* updates the state's distribution given a new observation. These two steps are detailed below.

Prediction step

The discrete evolution equation (4.33) becomes, in the linear case:

$$\mathbf{x}_t = \mathbf{M}_t \mathbf{x}_{t-1} + \eta_t. \tag{4.38}$$

Using the fact that $\eta_t \sim \mathcal{N}(0, \mathbf{Q}_t)$, the predicted state's distribution can be written:

$$p(\mathbf{x}_t^f) = \mathcal{N}(\mathbf{x}_t \mid \bar{\mathbf{x}}_t^f, \mathbf{P}_t^f), \tag{4.39}$$

where

$$\bar{\mathbf{x}}_t^f = \mathbf{M}_t \bar{\mathbf{x}}_{t-1}^a, \tag{4.40}$$

$$\mathbf{P}_t^f = \mathbf{M}_t \mathbf{P}_{t-1}^a \mathbf{M}_t^{\mathrm{T}} + \mathbf{Q}_t. \tag{4.41}$$

Assimilation step

Given a new observation, the state's distribution is updated following the derivations from Section 3.3.1, giving:

$$p(\mathbf{x}_t^a | \mathbf{x}_t^f, \mathbf{y}_t) = \mathcal{N}(\mathbf{x}_t \mid \bar{\mathbf{x}}_t^a, \mathbf{P}_t^a), \tag{4.42}$$

$$\mathbf{x}_t^a = \mathbf{x}_t^f + \mathbf{K}_t (\mathbf{y}_t - \mathbf{H}^{\mathrm{T}} \mathbf{x}_t^f), \tag{4.43}$$

$$\mathbf{P}_t^a = (\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_t^f \tag{4.44}$$

$$\mathbf{K}_t = \mathbf{P}_t^f \mathbf{H}^{\mathrm{T}} (\mathbf{H} \mathbf{P}_t^f \mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}. \tag{4.45}$$

Because of the linearity assumption, the above computations maintain the Gaussian nature of the state's distribution.

Square root formulation

Short after its introduction, the Kalman Filter was reported to suffer from filter divergence – i.e. the filter would lose track of the true process – in certain situations where the covariance matrix $\mathbf{P}$ becomes ill-conditioned. The most acknowledged causes of filter divergence include the processing of very accurate observations, observations with great variations in accuracy across the observed domain (some regions having high accuracy while others are hardly observable), the fact that the linear/Gaussian assumption is unrealistic for the process considered, and numerical errors due to the limited precision of the model (Kaminski et al., 1971; Fitzgerald, 1971). A mathematical analysis of filter divergence and conditions under which the distance of the model to the true process remains bounded can be found in (Price, 1968; Fitzgerald, 1971).

In order to increase the numerical stability of the algorithm, a square root formulation was derived based on Cholesky decomposition of the $\mathbf{P}$ matrix (Potter (1964); Andrews (1968); Bierman

(1977), see also Kaminski et al. (1971); Maybeck (1979)). If $\mathbf{S}$ is a triangular matrix resulting from the Cholesky decomposition of $\mathbf{P}$, so that $\mathbf{P} = \mathbf{S}\mathbf{S}^T$, then a formulation of the Kalman Filter equations based on $\mathbf{S}$ can be derived (i.e. $\mathbf{S}$ is estimated rather than $\mathbf{P}$). This increases numerical stability, in particular since the effective numerical precision is doubled (Kaminski et al., 1971). Even though these errors might still affect the estimate of $\mathbf{S}$ (to a lesser extend than in the standard formulation of the KF), $\mathbf{P}$ is guaranteed to remain positive definite. It is to be noted that the Square Root formulation only addresses filter divergence if the divergence is due to numerical issues.

Several "Square Root Kalman Filters" have been devised, but their derivations are not reproduced here. The interested reader can find these derivations in Kaminski et al. (1971); Bierman (1977) and other references therein.

4.3.3 Extended Kalman Filter

If the model m is non-linear, the prediction equations (4.40) and (4.41) for the linear case do not hold anymore. A common solution to the problem is to resort to using a linearised model to compute the predicted covariance. The resulting filter is known as the Extended Kalman Filter (Jazwinski, 1970; Maybeck, 1979; Evensen, 2007).

The derivation requires applying a Taylor expansion to (4.33):

$$\mathbf{x}_t^f = m(\mathbf{x}_{t-1}^a) + \eta_t \tag{4.46}$$

$$\approx m(\bar{\mathbf{x}}_{t-1}^a) + \hat{\mathbf{M}}_{t-1}(\mathbf{x}_{t-1}^a - \bar{\mathbf{x}}_{t-1}^a) + \eta_t, \tag{4.47}$$

$$\approx \bar{\mathbf{x}}_t^f + \hat{\mathbf{M}}_{t-1}(\mathbf{x}_{t-1}^a - \bar{\mathbf{x}}_{t-1}^a) + \eta_t, \tag{4.48}$$

where $\hat{\mathbf{M}}_{t-1} = \frac{\mathrm{d}m}{\mathrm{d}\mathbf{x}_t}(\mathbf{x}_t^a)$ is the tangent linear model, i.e. the Jacobian of m, computed about the current state estimate. Second and higher order terms in the Taylor expansion have been discarded, but could be incorporated for greater accuracy.

The covariance of the predicted state can then be computed using the approximation (4.48):

$$\mathbf{P}_t^f = \left\langle (\mathbf{x}_t^f - \bar{\mathbf{x}}_t^f)(\mathbf{x}_t^f - \bar{\mathbf{x}}_t^f)^{\mathrm{T}} \right\rangle \tag{4.49}$$

$$\approx \left\langle \left[\hat{\mathbf{M}}_{t-1}(\mathbf{x}_{t-1}^a - \bar{\mathbf{x}}_{t-1}^a) + \eta_t\right] \left[\hat{\mathbf{M}}_{t-1}(\mathbf{x}_{t-1}^a - \bar{\mathbf{x}}_{t-1}^a) + \eta_t\right]^{\mathrm{T}} \right\rangle \tag{4.50}$$

$$\approx \hat{\mathbf{M}}_{t-1}\, \mathbf{P}_{t-1}^a\, \hat{\mathbf{M}}_{t-1} + \mathbf{Q}_t \tag{4.51}$$

Notice how the result looks similar to the standard Kalman Filter's predicted covariance (Eq. (4.41)), only the model has been replaced by its tangent linear estimate. The mean is still propagated using the full non-linear model. This gives the following prediction equations for the

Extended Kalman Filter:

$$p(\mathbf{x}_t^f) = \mathcal{N}(\mathbf{x}_t^f \mid \bar{\mathbf{x}}_t^f, \mathbf{P}_t^f) \tag{4.52}$$

$$\bar{\mathbf{x}}_t^f = m(\bar{\mathbf{x}}_{t-1}^a) \tag{4.53}$$

$$\mathbf{P}_t^f \approx \hat{\mathbf{M}}_{t-1}\, \mathbf{P}_{t-1}^a\, \hat{\mathbf{M}}_{t-1} + \mathbf{Q}_t \tag{4.54}$$

The update step is left unchanged, although in the case of a non-linear observation operator h, a similar linearisation procedure can be applied to obtain $\hat{\mathbf{H}}$.

The Extended Kalman Filter (EKF) maintains a Gaussian approximation to the true propagated distribution of the state, which only holds if the requirements for the linearisation are met (smooth model at the integration time considered). If the model is strongly non-linear at the time step of interest, linearisation errors will cease to be negligible, which can lead to filter divergence (Evensen, 1992). Note that since the model is non-linear, there is no guarantee that the predicted distribution will still be Gaussian (it is more likely not to be). The Gaussian estimate might thus give a very poor representation of the actual posterior distribution.

In order to overcome the linearisation issues found in the EKF, alternative ensemble-based methods have been devised, which allow for the covariance to be propagated using the full non-linear model. These perform approximations on the distributions rather than on the model. Typically, the state's distribution is represented by an ensemble of particles, from which the real statistics can be inferred. The two main resulting filters, the Ensemble Kalman Filter and the Particle Filter, will be discussed in the following sections.

4.3.4 Ensemble Kalman Filter

Ensemble filters use the intuition that it is easier to approximate a probability distribution than it is to approximate a non-linear operator. Instead of trying to propagate the exact distribution through a linearised model (as does the EKF), ensemble methods use a Monte Carlo approximation to the distribution and propagate it through the exact model. Three main methods can be found in the literature: the Ensemble Kalman Filter (this section), the Unscented Kalman Filter (Section 4.3.5) and the Particle Filter (Section 4.3.6).

In the Ensemble Kalman Filter (EnKF), the state's distribution is represented by an ensemble of particles (typically of order 100), the mean and covariance of which approximate these of the real distribution, assumed to remain Gaussian (Evensen, 1994, 2007). The initial ensemble is constructed by sampling from the initial distribution $p(\mathbf{x}_0)$. Each ensemble member $\mathbf{x}_i$ is then propagated using the (non-linear) model, giving a predicted ensemble. The predicted mean and

covariance are estimated from the predicted ensemble, as follows:

$$\bar{\mathbf{x}}_t^f \approx \frac{1}{N}\sum_{i=1}^{N}[\mathbf{x}_i]_t^f,$$

$$\mathbf{P}_t^f \approx \frac{1}{N-1}\sum_{i=1}^{N}\left([\mathbf{x}_i]_t^f - \bar{\mathbf{x}}_t^f\right)\left([\mathbf{x}_i]_t^f - \bar{\mathbf{x}}_t^f\right)^{\mathrm{T}},$$

where N is the ensemble size and $[\mathbf{x}_i]$ denotes the i-th ensemble member.

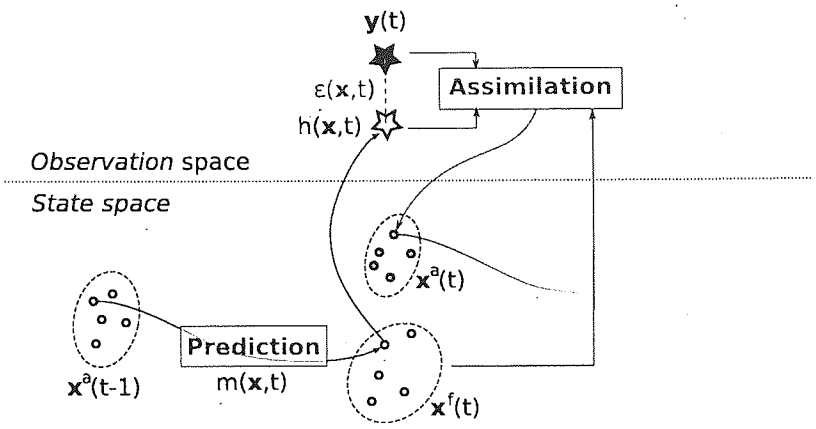


Figure 4.8: Ensemble Kalman Filter. The state's distribution is represented by an ensemble of realisations (sample) which is propagated through the full non-linear model. Each ensemble member is updated given the observation.

The update step consists in updating each ensemble member given the observations $\mathbf{y}_t$. Ideally, one would need to use an ensemble of observations, so that to each particle corresponds a different observation. This would ensure the covariance structure of the ensemble is maintained in agreement with the observation's error covariance. However, generating an ensemble of observations in practice would be too costly, and only a single measurement $\mathbf{y}_t$ is usually available.

In its original formulation (Evensen, 1992), the EnKF used the observations $\mathbf{y}_t$ to update all ensemble members. (Burgers et al., 1998) showed that using the same observations to update all ensemble members resulted in the state's covariance matrix being underestimated. Namely, the correct covariance is given by:

$$\mathbf{P}^a = (\mathbf{I}-\mathbf{KH})\mathbf{P}^f(\mathbf{I}-\mathbf{KH})^{\mathrm{T}} + \mathbf{KRK}^T \tag{4.55}$$

$$= (\mathbf{I}-\mathbf{KH})\mathbf{P}^f. \tag{4.56}$$

but the effect of using a single observation gives the following estimate of $\mathbf{P}^a$ (Whitaker and Hamill, 2002):

$$\mathbf{P}^a = (\mathbf{I}-\mathbf{KH})\mathbf{P}^f(\mathbf{I}-\mathbf{KH})^{\mathrm{T}}. \tag{4.57}$$

To address the issue of the missing term, they suggested an ensemble of observations was generated by sampling from a Gaussian distribution with mean $\mathbf{y}_t$ and covariance to be specified (usually set to $\mathbf{R}$) (Burgers et al., 1998; Houtekamer and Mitchell, 1998).

However, Whitaker and Hamill (2002) showed that the effect of perturbing the observations resulted in increased sampling error, leading to an underestimation of the covariance in the assimilation step. They provided a square-root formulation of the EnKF in which that covariance is correctly estimated, by seeking a matrix $\hat{\mathbf{K}}$ satisfying:

$$\begin{cases} \mathbf{P}^a &= (\mathbf{I} - \hat{\mathbf{K}}\mathbf{H})\mathbf{P}^f(\mathbf{I} - \hat{\mathbf{K}}\mathbf{H})^{\mathrm{T}} \\ &= (\mathbf{I} - \hat{\mathbf{K}}\mathbf{H})\mathbf{P}^f, \end{cases} \tag{4.58}$$

which led to the solution:

$$\hat{\mathbf{K}} = \mathbf{P}^f\mathbf{H}^T\left[(\mathbf{H}\mathbf{P}^f\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-\frac{1}{2}}\right]^{\mathrm{T}} \times \left[(\mathbf{H}\mathbf{P}^f\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{\frac{1}{2}} + \mathbf{R}^{\frac{1}{2}}\right]^{-1}, \tag{4.59}$$

where $\mathbf{A}^{\frac{1}{2}}$ denotes the square root of a matrix $\mathbf{A}$ (obtained, for instance, by Cholesky decomposition). That formulation does not require the use of perturbed observations and provides a correct estimate of $\mathbf{P}^a$. Several other square root formulations have been provided, a review of which can be found in (Tippet et al., 2003).

4.3.5 Unscented Kalman Filter

Another alternative to the Extended Kalman Filter which relies on approximating the state's distribution using an ensemble is the Unscented Kalman Filter (Julier et al., 1995; Wan and Van Der Merwe, 2000; Julier et al., 2004). In the Unscented Kalman Filter (UKF), an ensemble of "sigma points" is generated which has the correct mean and variance. Unlike the EnKF, this ensemble is not a random sample from the state's distribution, but a set of points designed to capture the exact first two moments.

A set of $2N+1$ sigma points, N being the state vector's dimension, is generated by selecting the mean and $2N$ points on covariance contours. For instance, if $N = 2$, the mean and 4 points on the axis of an elliptic covariance contour are selected. The points are weighted in such a way as to allow the covariance contour to be more or less close to the mean, while retaining the correct covariance. The sigma points and their weights are denoted respectively $\mathbf{x}^i$ and w^i, for $i = 1..N$.

The selection of the sigma points, as described in (Julier et al., 2004), is illustrated in Figure 4.9. The ensemble $\{\mathbf{x}^i, w^i\}$ allows the mean and covariance to be computed using:

$$\bar{\mathbf{x}} = \sum_{i=0}^{2N} w^i\mathbf{x}^i \tag{4.62}$$

$$\mathbf{P} = \sum_{i=0}^{2N} w^i(\mathbf{x}^i - \bar{\mathbf{x}})(\mathbf{x}^i - \bar{\mathbf{x}})^{\mathrm{T}}. \tag{4.63}$$

The choice of w^0 controls the spread of the sigma-point. Positive values of w^0 gather the sigma points around the mean, while negative values move them away from it. (Note that a negative w^0 is

1. Select the mean as the first sigma point: $\mathbf{x}^0 = \bar{\mathbf{x}}$ and choose its weight w^0

2. Compute a square root matrix $\mathbf{S}$ of the weighted covariance matrix:

$$\mathbf{S}\mathbf{S}^{\mathrm{T}} = \frac{N}{1-w_0}\mathbf{P} \tag{4.60}$$

3. Select $2N$ sigma points as perturbations from the mean state by plus or minus the i-th column of $\mathbf{S}$:

$$\mathbf{x}^i = \bar{\mathbf{x}} + \mathbf{S}_i$$
$$\mathbf{x}^{N+i} = \bar{\mathbf{x}} - \mathbf{S}_i$$

and weight them according to:

$$w^i = w^{N+i} = \frac{1-w_0}{2N}. \tag{4.61}$$

Figure 4.9: Unscented Kalman Filter: algorithm for the selection of sigma points

not inconsistent, since the weighted sigma points are only a means to approximate the state's first and second moments, but do not provide a representation of the state's probability density function (Julier et al., 2004), unlike the EnKF and other Monte Carlo approximations). For instance, if the model is strongly non-linear, errors due to sampling "non-local sampling effects" can lead to "significant difficulties" (Van der Merwe et al., 2000) and one might want to keep the sigma points close to the mean. Further discussion on the scaling of the ensemble spread can be found in (Julier, 2002).

4.3.6 Sequential Monte-Carlo (Particle Filter)

The Particle Filter (see for example Doucet et al. (2001); Arulampalam et al. (2002)) is a filter based on Monte Carlo methods. Its main advantage over Kalman-based filters is that it does not restrict the state's probability density function to be Gaussian.

Monte Carlo methods provide a means to approximate continuous distributions with a (discrete) set of samples $\{\mathbf{x}_i\}_{i=1:N}$ called "particles". Assuming N independent and identically distributed (i.i.d.) realisations $\{\mathbf{x}_i\}_{i=1:N}$ can be drawn from some probability distribution p, then the probability of the random variable $\mathbf{x}$ lying within the interval $\mathrm{d}\mathbf{x}$ can be approximated by:

$$P_N(\mathrm{d}\mathbf{x}) = \frac{|\{\mathbf{x}_i \in \mathrm{d}\mathbf{x}\}|}{N} = \frac{1}{N}\sum_{i=1}^{N}\delta_{\mathbf{x}_i}(\mathrm{d}\mathbf{x}) \tag{4.64}$$

where $|\{\mathbf{x}_i \in d\mathbf{x}\}|$ denotes the number of realisations falling within $d\mathbf{x}$, and $\delta_{\mathbf{x}_i}(d\mathbf{x})$ is the delta Dirac mass located in $\mathbf{x}_i$ (taking value 1 if $\mathbf{x}_i$ lies within $d\mathbf{x}$ and 0 otherwise). Figure 4.10 illustrate this principle using 10 samples (spheres) from a distribution with 4 possible outcomes. Here, $P_N(d\mathbf{x}) = 4/10 = 0.4$.

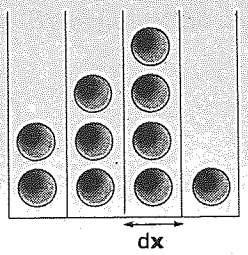


Figure 4.10: Binned samples from an unknown distribution.

This representation of distributions allows expectations, on which most common statistical indicators rely, to be easily derived. Recall the definition of the expectation of some quantity $f(\mathbf{x})$ over some distribution p:

$$E[f(\mathbf{x})] = \int f(\mathbf{x})\, \mathrm{p}(\mathbf{x})\, d\mathbf{x}. \tag{4.65}$$

From (4.64), it results that $E[f(\mathbf{x})]$ accepts the following Monte Carlo estimate:

$$E_N[f(\mathbf{x})] = \int f(\mathbf{x})\, P_N(\mathbf{x})\, d\mathbf{x} = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i). \tag{4.66}$$

Particle Filters address the general problem of estimating the full posterior distribution of the state, $\mathrm{p}(\mathbf{X}_t|\mathbf{Y}_t)$, from which the usual marginal filtering distribution $\mathrm{p}(\mathbf{x}_t|\mathbf{Y}_t)$ can easily be inferred. If one were able to represent the posterior distribution using a Monte-Carlo approximation, one would be able to propagate it through the full non-linear model and obtain an estimate of the predicted distribution $\mathrm{p}(\mathbf{X}_{t+1}|\mathbf{Y}_t)$ which relies neither on linearisation nor Gaussian approximation. Unfortunately, most posterior distributions resulting from the propagation of some prior through a non-linear model are intractable and the Monte-Carlo approach cannot be applied directly. A solution is provided by a method called *importance sampling*, which is described in the next section.

Importance sampling

The principle of importance sampling is the following: for distributions which cannot be sampled from directly, a set of particles can still be obtained by sampling from another, more tractable distribution, and weighting the particles appropriately when computing expectations.

If we consider an (arbitrary) *importance distribution* $\mathrm{q}(\mathbf{X}_t|\mathbf{Y}_t)$ which can easily be sampled from, the expectation with respect to the posterior $\mathrm{p}(\mathbf{X}_t|\mathbf{Y}_t)$ can be written:

$$E[f(\mathbf{X}_t)] = \int f(\mathbf{X})\, \mathrm{p}(\mathbf{X}_t|\mathbf{Y}_t)\, d\mathbf{X}_t \tag{4.67}$$

$$= \int f(\mathbf{X})\, \frac{\mathrm{p}(\mathbf{X}_t|\mathbf{Y}_t)}{\mathrm{q}(\mathbf{X}_t|\mathbf{Y}_t)}\, \mathrm{q}(\mathbf{X}_t|\mathbf{Y}_t)\, d\mathbf{X}_t \tag{4.68}$$

$$= \frac{1}{\mathrm{p}(\mathbf{Y}_t)} \int f(\mathbf{X})\, \frac{\mathrm{p}(\mathbf{Y}_t|\mathbf{X}_t)\, \mathrm{p}(\mathbf{X}_t|\mathbf{Y}_{t-1})}{\mathrm{q}(\mathbf{X}_t|\mathbf{Y}_t)}\, \mathrm{q}(\mathbf{X}_t|\mathbf{Y}_t)\, d\mathbf{X}_t \tag{4.69}$$

where the posterior has been substituted using Bayes' rule in (4.69). By defining the *importance weight*:

$$w(\mathbf{X}_t) = \frac{\mathrm{p}(\mathbf{Y}_t|\mathbf{X}_t)\,\mathrm{p}(\mathbf{X}_t|\mathbf{Y}_{t-1})}{\mathrm{q}(\mathbf{X}_t|\mathbf{Y}_t)} \propto \frac{\mathrm{p}(\mathbf{X}_t|\mathbf{Y}_t)}{\mathrm{q}(\mathbf{X}_t|\mathbf{Y}_t)} \tag{4.70}$$

and noting that:

$$\mathrm{p}(\mathbf{Y}_t) = \int \mathrm{p}(\mathbf{Y}_t|\mathbf{X}_t)\,\mathrm{p}(\mathbf{X}_t|\mathbf{Y}_{t-1})\,\mathrm{d}\mathbf{X}_t \tag{4.71}$$

$$= \int w(\mathbf{X}_t)\,\mathrm{q}(\mathbf{X}_t|\mathbf{Y}_t)\,\mathrm{d}\mathbf{X}_t, \tag{4.72}$$

$E[f(\mathbf{X})]$ can be rewritten:

$$E[f(\mathbf{X}_t)] = \frac{\int f(\mathbf{X}_t)\,w(\mathbf{X}_t)\,\mathrm{q}(\mathbf{X}_t|\mathbf{Y}_t)\,d\mathbf{X}_t}{\int w(\mathbf{X}_t)\,\mathrm{q}(\mathbf{X}_t|\mathbf{Y}_t)\,\mathrm{d}\mathbf{X}_t} \tag{4.73}$$

$$= \frac{E[w(\mathbf{X}_t)\,f(\mathbf{X}_t)]_{\mathrm{q}}}{E[w(\mathbf{X}_t)]_{\mathrm{q}}}, \tag{4.74}$$

where the q index indicates that the expectation is carried over the distribution q, not p. Applying (4.66) to (4.74) leads to the Monte Carlo estimate:

$$\tilde{E}_N[f(\mathbf{X}_t)] = \frac{\frac{1}{N}\sum_{i=1}^{N} w(\mathbf{X}_t^i) f(\mathbf{X}_t^i)}{\frac{1}{N}\sum_{i=1}^{N} w(\mathbf{X}_t^i)} \tag{4.75}$$

$$= \sum_{i=1}^{N} \tilde{w}_t^i f(\mathbf{X}_t^i) \tag{4.76}$$

where $\tilde{w}_t^i = w(\mathbf{X}_t^i)/\sum_{j=1}^{N} w(\mathbf{X}_t^j)$ are the *normalised importance weights*. Applying this result to the delta function gives the following expression for the posterior estimate, which is the importance sampling equivalent of (4.64):

$$\tilde{P}_N(\mathrm{d}\mathbf{X}_t|\mathbf{Y}_t) = \sum_{i=1}^{N} \tilde{w}_t^i\,\delta_{\mathbf{X}_t^i}(\mathrm{d}\mathbf{X}_t). \tag{4.77}$$

Choosing the importance distribution

Choosing the importance distribution to have certain properties can simplify the computations considerably. In particular, it is convenient for sequential filtering to chose an importance distribution which factorises as:

$$\mathrm{q}(\mathbf{X}_t|\mathbf{Y}_t) = \mathrm{q}(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t)\;\mathrm{q}(\mathbf{X}_{t-1}|\mathbf{Y}_{t-1}). \tag{4.78}$$

Using the fact that the process is assumed Markov:

$$\mathrm{p}(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{t-1}) = \mathrm{p}(\mathbf{x}_t|\mathbf{x}_{k-1}) \tag{4.79}$$

and that the observations are conditionally uncorrelated:

$$\mathrm{p}(\mathbf{Y}_t|\mathbf{X}_t) = \prod_{k=1}^{t} \mathrm{p}(\mathbf{y}_k|\mathbf{x}_k), \tag{4.80}$$

a recursive formulation of the weights (4.70) can be derived (see Arulampalam et al. (2002) for full derivation):

$$\tilde{w}_t \propto \tilde{w}_{t-1}^i \frac{p(\mathbf{y}_t|\mathbf{x}_t^i)\, p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i|\mathbf{X}_{t-1}^i, \mathbf{Y}_t)} \tag{4.81}$$

This formulation allows for the updated set of weights to be computed by simply scaling the weights according to the right hand-side ratio, and normalising them.

The choice of the importance distribution will depend mostly on the problem addressed, and has a strong impact on the quality of the assimilation. Doucet (1998) shows that, for the filtering density, the distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}^i, \mathbf{y}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1}^i, \mathbf{y}_t)$ is optimal, in that it maximises the variance of the weight. Furthermore, the resulting weight is shown to be independent of the sample $\mathbf{x}_{t-1}^i$. However, as noted in Arulampalam et al. (2002), this distribution is usually not a practical choice, as one needs to be able to sample from it and it yields an expression of the weight involving an integral which is not easily computed. As can easily see from (4.81), a particularly convenient alternative is to choose $q(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t)$ to be the transition prior $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, in which case the weights simply need to be multiplied by the likelihood of the corresponding particle:

$$\tilde{w}_t^i \propto \tilde{w}_{t-1}^i\, p(\mathbf{y}_t|\mathbf{x}_t^i) \tag{4.82}$$

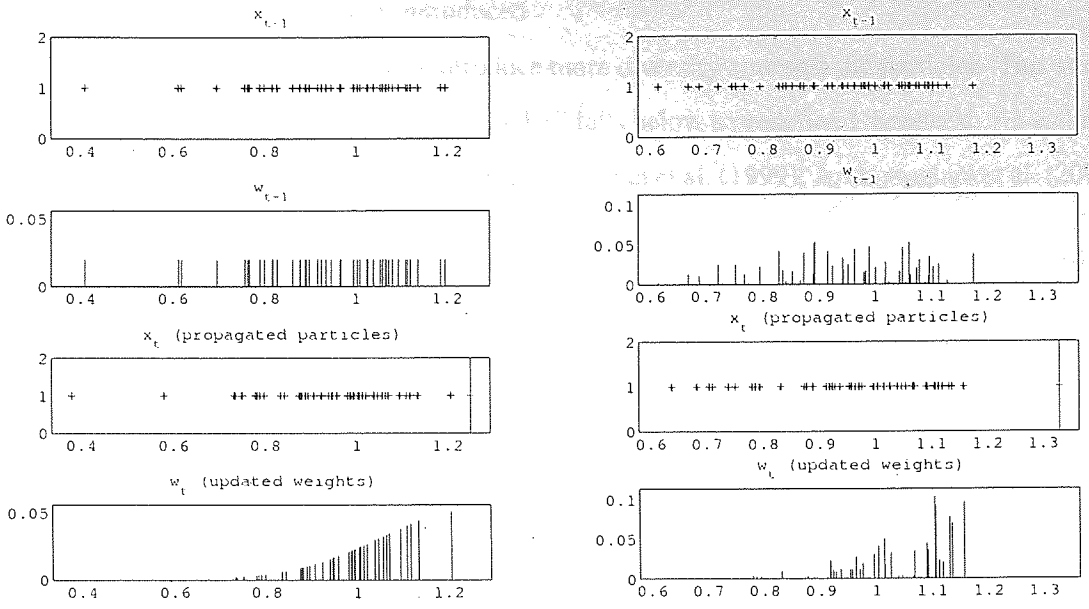
Implementation of the filter

The filter is initialised by sampling $\{\mathbf{X}_{i,0}\}_{i=1:N}$ from the prior $p(\mathbf{X}_0)$ and setting the corresponding weights $\{w_0^i\}$ to $1/N$.

Prediction step Given a weighted sample $\{w_{t-1}^i, \mathbf{X}_{t-1}^i\}$, the particles are propagated forward to time t according to (2.5), using the full non-linear model, while the weights remain unchanged. This gives an ensemble of weighted particles $\{w_{t-1}^i, \mathbf{X}_t^i\}$ which approximates the predictive distribution $p(\mathbf{X}_t|\mathbf{X}_{t-1})$.

Assimilation step The weights are updated according to (4.82) and normalised, with the particles left unchanged. The weighted sample $\{w_t^i, \mathbf{X}_t^i\}$ provides an approximation to the posterior $p(\mathbf{X}_t|\mathbf{y}_t)$, which then becomes the prior for the following time step.

Figure 4.11 shows the filter in action on the 1D double-well system. The filter uses 50 weighted particles. 4.11 (a) shows the initial prediction/update using the prior (t=0) while Figure 4.11 (b) and (c) show similar information at subsequent time steps.

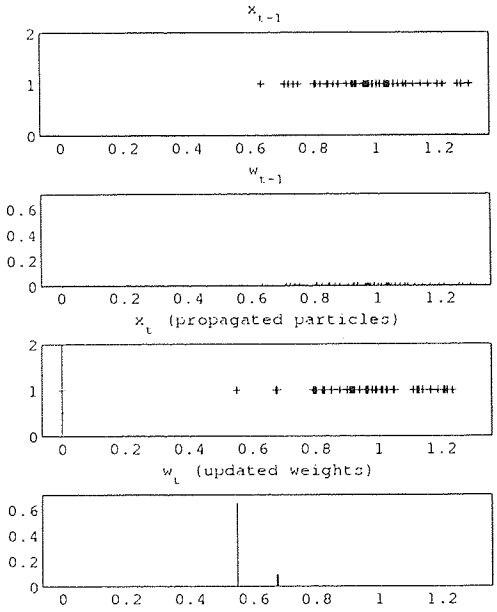


(a) (b)

Figure 4.11: Prediction and update of weighted particles on a simple 1-d model at three different times. The plot shows, from the top, the particles and weights of the prior (first and second rows), the propagated particles (third row) and the updated weights (fourth row). The observation (third row) is plotted as a cross with dotted grid line.

Degeneracy issue

One can see from Figures 4.11 (b) and (c) that the total particle weight, initially shared amongst all particles equally, tends to gather over fewer and fewer particles as observations are assimilated. The evolution of the weights is conditioned on the likelihood of the observations. The weights of particles which lie in regions of low likelihood tend to decrease, while the weights of particle lying in regions of high likelihood increase. Since there is no creation of new particles nor relocation of existing ones, the set of "effective particles" (i.e. particles with a non negligible weight) is likely to shrink and collapse as even the best particle will eventually diverge from the true process. This phenomenon is known as filter degeneracy and would eventually lead to filter divergence.



(c)

To prevent the collapse of the set of par-

ticles, a resampling step is usually introduced

after updating the weights, in order to introduce more diversity amongst the particles. This step is triggered when the number of "effective particles" falls below a predefined threshold. An estimate of the number of "effective particle" is given by Bergman et al. (1999); Arulampalam et al. (2002):

$$\tilde{N}_{eff}(\mathrm{t}) = \frac{1}{\sum_{i=1}^{N}(w_{i,\mathrm{t}})^2}. \tag{4.83}$$

When $\tilde{N}_{eff}(t)$ goes below a specified threshold, a new set of weights and particles is generated in agreement with the estimated posterior distribution. This step is called *resampling*. There are several algorithms available to achieve resampling: multinomial resampling, systematic resampling, stratified resampling, residual resampling, branching methods, etc. A review of the most common of these resampling methods is given below. See also Douc et al. (2005); Hol et al. (2006) for similar reviews.

Resampling methods

Resampling consists in having particles with high weights duplicated while particles with very small weights are discarded, in order to generate a new set of equally likely particles. The new set of weighted particles $\{\hat{w}^i, \hat{\mathbf{x}}^i\}$ should satisfy $\hat{w}^i = 1/N$ and $\sum_{i=1}^{N} \delta_{\mathbf{x}^i}(\hat{\mathbf{x}}^i)/N \approx w_i$, where $\delta(\mathbf{x}^i) = 1$ and $\delta(x \neq x^i) = 0$. In other words, the particle i is replicated n times so that $n/N \approx w_i$.

- **Multinomial Resampling**

  The Multinomial Resampling algorithm (Gordon et al., 1993) consists in generating an ensemble of N points $\{u^i\}_{i=1..n}$ in the interval $[0, 1[$ by sampling from the uniform distribution $U[0, 1[$. The new particles are obtained by selection (possibly more than once) and rejection of the existing particles, based on the inverse cumulative posterior distribution. In other words, if $w^j < u^i \leq w^{j+1}$, particle j is selected. The algorithm is illustrated in Figure 4.12.

- **Systematic Resampling**

  Systematic Resampling (Kitagawa, 1996; Arulampalam et al., 2002) functions in very much the same way as Multinomial Resampling, except the u^i are spread homogeneously in the interval $[0, 1[$. Namely, u^1 is drawn from $U[0, \frac{1}{N}[$, and $u^{j+1} = u^j + \frac{1}{N}$. The rest of the algorithm is identical. Figure 4.13 illustrates this algorithm.

- **Stratified Resampling** Stratified Resampling (Kitagawa, 1996) is a variant of Multinomial Resampling where the interval $[0, 1[$ is split into N identical segments and the u^i are sampled uniformly from segment i. That is to say, u^i is drawn from $U[\frac{i-1}{N}, \frac{i}{N}[$. A summary of the algorithm is given in Figure 4.12.

- **Residual Resampling** In Residual Resampling (Liu and Chen, 1998), each particle $\mathbf{x}^i$ is selected n^i times, where $n^i = \lfloor Nw^i \rfloor$ is the number of particles corresponding to a proportion w^i of the total population (rounded down to the closest integer). It is clear that, because of the round off, this method might only select $M < N$ particles from the initial set. The remaining $N - M$ particles are selected as follows. Each particle is given a "residual weight" w_r^i equal to the amount of w^i lost in the round off, i.e. $w_r^i = Nw^i - n^i$. The w_r^i are then normalised: $\tilde{w}_r^i = w_r^i / \sum w_r^i$. $N - M$ particles are then selected from the set $\{\mathbf{x}^i, \tilde{w}_r^i\}$ using one of the previous resampling methods. An illustrative example is given on Figure 4.12.

Multinomial Resampling

1. Sample $\{u_i\}_{i=1..N}$ from the uniform distribution $U[0, 1/N[$. Set $i = 1$, $j = 1$.
2. While $C(\mathbf{x}^i) \leq u^j$, increment i.
3. Set $\hat{\mathbf{x}}^j = \mathbf{x}^i$.
4. Increment j and go back to 2.
5. Once done, set $\hat{w}^j = \frac{1}{N}$ for all j.

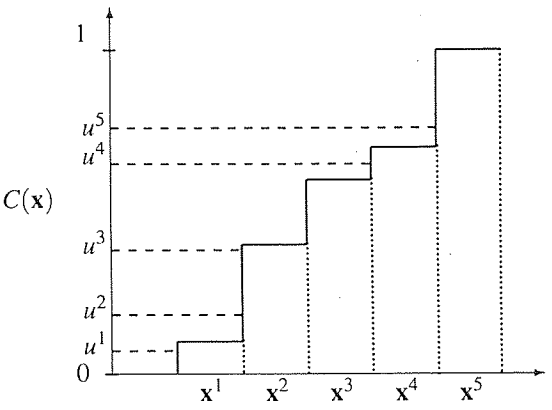


Figure 4.12: Multinomial Resampling – In this example, the algorithm selects particle 1, duplicates particle 2, discards particle 3, selects particles 4 and 5, resulting in the new set of particles $\{\hat{\mathbf{x}}^i\} = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^2, \mathbf{x}^4, \mathbf{x}^5\}$. The weights are then reinitialised.

Systematic Resampling

1. Sample a starting point u_1 from the uniform distribution $U[0, 1/N[$. Set $j = 1$.
2. While $C(\mathbf{x}^i) \leq u^j$, increment i.
3. Set $\hat{\mathbf{x}}^j = \mathbf{x}^i$.
4. Set $u_{j+1} = u_j + \frac{1}{N}$.
5. Increment j and go back to 2.
6. Once done, set $\hat{w}^j = \frac{1}{N}$ for all j.

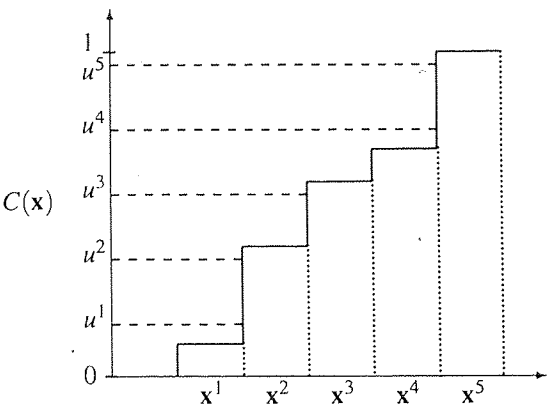


Figure 4.13: Systematic Resampling – In this example, the algorithm discards particle 1, duplicates particle 2, keeps particle 3, discards particle 4 and duplicates particle 5, resulting in the new set of particles $\{\hat{\mathbf{x}}^i\} = \{\mathbf{x}^2, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^5, \mathbf{x}^5\}$. The weights are then reinitialised.

Stratified Resampling

1. Sample a starting point u_1 from the uniform distribution $U[0, 1/N[$. Set $j = 1$.
2. While $C(\mathbf{x}^i) \leq u^j$, increment i.
3. Set $\hat{\mathbf{x}}^j = \mathbf{x}^i$.
4. Sample u_{j+1} from $U[\frac{i-1}{N}, \frac{i}{N}[$.
5. Increment j and go back to 2.
6. Once done, set $\hat{w}^j = \frac{1}{N}$ for all j.

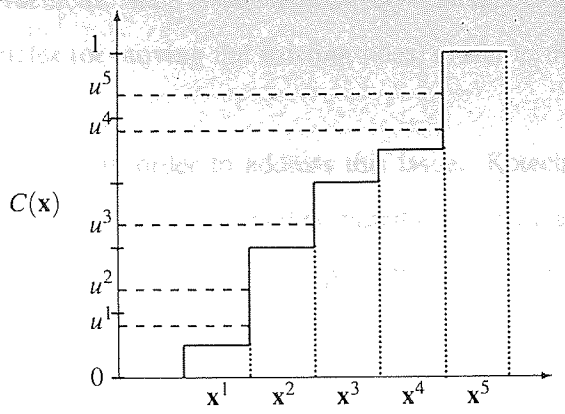


Figure 4.14: Stratified Resampling – In this example, the algorithm discards particle 1, duplicates particle 2, selects particle 3, discards particle 4 and duplicates particle 5, resulting in the new set of particles $\{\hat{\mathbf{x}}^i\} = \{\mathbf{x}^2, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^5, \mathbf{x}^5\}$. The weights are then reinitialised.

Residual Resampling

i	w^i	$n^i = \lfloor Nw^i \rfloor$	$w_r^i = Nw^i - n^i$	$\tilde{w}_r^i = w_r^i / \sum w_r^i$
1	0.11	0	0.55	0.18
2	0.30	1	0.50	0.17
3	0.19	0	0.95	0.32
4	0.11	0	0.55	0.18
5	0.29	1	0.45	0.15
Total	1.00	2	3.00	1.00

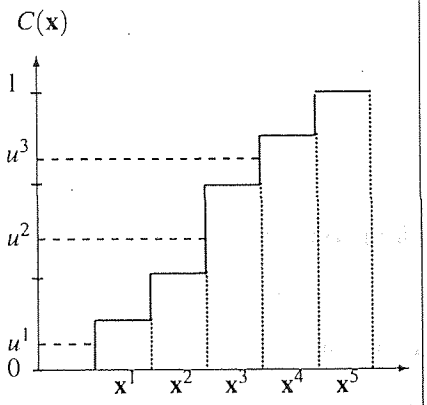


Figure 4.15: Residual Resampling – In this example, $N = 5$. The algorithm selects particles 2 and 5 once each according to n^i. The residual weights $\tilde{w}_r^i$ are then computed and Stratified Resampling is used to select the 3 remaining particles (in this case particles 1, 3 and 5). The new set of particles is thus $\{\hat{\mathbf{x}}^i\} = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^5, \mathbf{x}^5\}$. The weights are then reinitialised.

In all the resampling methods listed above, the new set of particles generated contains exact duplicates. These are expected to become distinct particles with distinct trajectories once the particles are propagated through the stochastic model. A good estimation of model error is thus critical to the filter's performance.

Further resampling schemes

The main issue with particle filters is that no particles are created in regions of high likelihood. It might happen that all particles drift away from the true process (as a result of an incorrect

description of model error). In such circumstances, there is no chance of bringing them back in line with the process using the above resampling methods, since these methods rely on duplicating existing particles, not actually creating new particles (or moving the existing ones) closer to the process.

Some new resampling methods have been devised in order to address this issue. Kotecha and Djuric (2003a); Xiong and Navon (2006) propose a resampling method based on a Gaussian approximation to the posterior. The new set of particles is generated by sampling from a Gaussian distribution with mean and covariance estimated from the current particles and weights. Although the methods is less prone to filter degeneracy, using a Gaussian resampling step means losing the main advantage of Particle Filters over other filtering methods which is the absence of the Gaussian constraint. A further extension proposed by Kotecha and Djuric (2003b) is the replacement of the set of weighted particles by a mixture of Gaussians, each Gaussian being independently managed as a separate Extended Kalman Filter. The Unscented Particle Filter (Van der Merwe et al., 2000; van der Merwe et al., 2001) provides a similar approach except each Gaussian in the mixture is treated as an Unscented Kalman Filter.

4.4 Summary of this chapter

Building up on the previous chapter, data assimilation methods for the dynamic estimation problem have been introduced. Dynamic least square estimation and dynamic 3D VAR have been shown to be straightforward extensions of their static counterparts, with the only difference being that the background estimate is provided by a previous forecast. An extension to 3D VAR, 4D VAR, has been introduced. 4D VAR provides a smoothing approach in which the trajectory of the state is estimated over a fixed time window. The optimal estimate minimises the departures of the trajectory to several sequential observations rather than a single observation.

Moving to a stochastic viewpoint, a review of filtering methods was provided. The Kalman Filter was derived as an intuitive extension of the linear Gaussian optimum from Section 3.3.1 to the dynamic context. Approximation in the non-linear case based on linearisation (Extended Kalman Filter) or Monte-Carlo representation of the state's distribution (Ensemble Kalman Filter, Unscented Kalman Filter) were discussed. A Monte-Carlo based approach for the non-Gaussian case (Particle Filter) was also introduced.

In the following chapter, we look at two experiments in which several of the methods discussed in this chapter are applied to two toy models.

5 Data assimilation with the Lorenz systems

CONTENTS

5.1 Foreword

Although several studies exist in which some of the data assimilation methods introduced in Chapter 4 are compared on non-linear models, there is a lack of a standard experimental setting allowing a fair comparison of data assimilation methods. It is important, in order to assess progress in the field, to be able to test new assimilation methods against old ones, and to do so, a set of experiments with transparent settings must be agreed on. This set of experiments should ideally allow for various factors to be evaluated, including performance in linear and non-linear regimes, robustness and computational cost. Along with the set of experiments, a set of validation methods should be agreed on, which would allow to assess the performance of each method according to various criteria. It is our hope that this chapter will provide a starting point for the development of a common benchmark in the data assimilation community.

Choice of models

The data assimilation methods selected in this experiment include the Extended Kalman Filter, the Ensemble Kalman Filter, the Particle Filter and 4D VAR, both in its strong and weak constraint formulation. Two non-linear models have been selected on which the selected data assimilation methods are to be evaluated. These models, the Lorenz 63 and Lorenz 96 models, were chosen for the following reasons:

- They are both widely used in the data assimilation literature,
- They are non-linear,
- They are of low and medium dimension (3 dimensions for Lorenz 63, 40 dimensions for Lorenz 96),
- They are simple to implement,
- They can be given a meteorological interpretation.

Related studies

Several similar studies involving some of the methods tested here can be found in the literature: Miller et al. (1994) implemented an Extended Kalman Filter on the Lorenz 63 system; Evensen (1997) ran an Ensemble Kalman Filter on the Lorenz 63 system, Harlim and Hunt (2007) tested variants of the Ensemble Kalman Filter on both the Lorenz 63 and Lorenz 96 systems; a comparison of 4D VAR and the Ensemble Kalman Filter on the Lorenz 63 system is given in Kalnay et al. (2007); 4D VAR is also compared on the Lorenz 96 system with a hybrid 4D Ensemble Kalman Filter in Fertig et al. (2007); a Particle Filter is tested on the Lorenz 63 system by Pham (2001).

Chapter outline

This chapter is organised as follows. First, both models are introduced and details on how observations are generated are given. Then, the experiment set-up, which is similar for both experiments, is discussed. Various parameters, across all methods, play a part on the way data assimilation is performed. These parameters are listed and choices for their values justified. Results are presented and the comparative performances of the methods discussed.

Notations

Several abbreviations are used in this section to identify the different assimilation methods. They are summarised in Table 5.1.

Abbreviation	Method's full name
KF	Kalman Filter
EKF	Extended Kalman Filter
EnKF	Ensemble Kalman Filter
PF	Particle Filter
4DVAR-S	Strong constraint 4D-VAR
4DVAR-W	Weak constraint 4D-VAR

Table 5.1: Summary of abbreviations used for data assimilation methods

5.1.1 The Lorenz 63 system

The Lorenz 63 system (Lorenz, 1963) is a simple 3-dimensional system often used in data assimilation research and development to test the robustness of a given data assimilation method to non-linearity. The Lorenz 63 system models turbulent hydrodynamic flow such as those observed in the atmosphere. It exhibits chaotic behaviour arising from strong sensitivity to initial conditions.

The state of the system, $\mathbf{x} = (x, y, z)$, is governed by the following set of equations:

$$\begin{cases} \frac{dx}{dt} &= -\sigma\,(x-y) \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dz}{dt} &= -bz + xy \end{cases} \tag{5.1}$$

where σ, r and b are constants. The physical interpretation of these constants is the following: x represents the intensity of convective motion, y the temperature difference between ascending and descending currents and z the distortion of the vertical temperature profile from linearity.

Parameter values are set as in Lorenz (1963): $\sigma = 10.0$, $r = 28.0$ and $b = 8/3$, which is the standard used in most experiments reported in the literature.

Tangent Linear Model

Propagation of the state is done using a Runge-Kutta scheme to solve the system of equations. EKF and 4DVAR need the tangent linear model to be derived. This is achieved by using a Euler approximation:

$$\begin{cases} x_t &= m^x(\mathbf{x}_{t-1}) &= x_{t-1} - \Delta t \times \sigma\,(x_{t-1} - y_{t-1}) \\ y_t &= m^y(\mathbf{x}_{t-1}) &= y_{t-1} + \Delta t \times (r x_{t-1} - y_{t-1} - x z_{t-1}) \\ z_t &= m^z(\mathbf{x}_{t-1}) &= y_{t-1} + \Delta t \times (-b z_{t-1} + x_{t-1} y_{t-1}) \end{cases} \tag{5.2}$$

where the model is denoted $m = (m^x, m^y, m^z)$. The tangent linear modelabout $\mathbf{x}_t$ is easily derived from (5.2):

$$\hat{\mathbf{M}}_t = \begin{pmatrix} \frac{\partial m^x}{\partial x}(\mathbf{x}_t) & \frac{\partial m^x}{\partial y}(\mathbf{x}_t) & \frac{\partial m^x}{\partial z}(\mathbf{x}_t) \\ \frac{\partial m^y}{\partial x}(\mathbf{x}_t) & \frac{\partial m^y}{\partial y}(\mathbf{x}_t) & \frac{\partial m^y}{\partial z}(\mathbf{x}_t) \\ \frac{\partial m^z}{\partial x}(\mathbf{x}_t) & \frac{\partial m^z}{\partial y}(\mathbf{x}_t) & \frac{\partial m^z}{\partial z}(\mathbf{x}_t) \end{pmatrix} = \mathbf{I} + \Delta t \begin{pmatrix} -\sigma & \sigma & 0 \\ r - z_t & -1 & -x_t \\ y_t & x_t & -b \end{pmatrix}. \tag{5.3}$$

Observed data

The "true trajectory" of the state is generated by propagating the full model forward in time with a time step $\Delta t = 0.01$ following Lorenz (1963). We map to an arbitrary time scale of 15 min per Δt (i.e. 1h corresponds to 4 time steps) to mimic time scales typically found in the atmosphere. A total of 10000 points (104 days) are generated. Figure 5.1 shows the trajectory of the true state both in space (top row) and time (bottom row) for the first 3000 steps.

Observations are then obtained by taking corrupted measurements of the true state at regular intervals. The true state is observed directly, i.e. the observation operator h is the identity: $h = \mathbf{H} = \mathbf{I}$. A range of observation intervals is considered in order to measure the effect of observation frequency on the quality of the assimilation. The time intervals between observations are listed further in Section 5.1.3. Gaussian additive white noise (i.e. uncorrelated) is added to the observed "true state" to simulate noisy observations. The black dots on Figure 5.1 correspond to 6-hourly noisy observations with covariance 0.04 of the system's amplitude.

5.1.2 The Lorenz 96 system

The Lorenz 96 system was introduced by Lorenz (1996) and provides a relevant representation for some aspects of atmospheric dynamics (non-linearity, chaotic behaviour). It is a commonly used model for testing state-space models under non-linear conditions, in a data assimilation context.

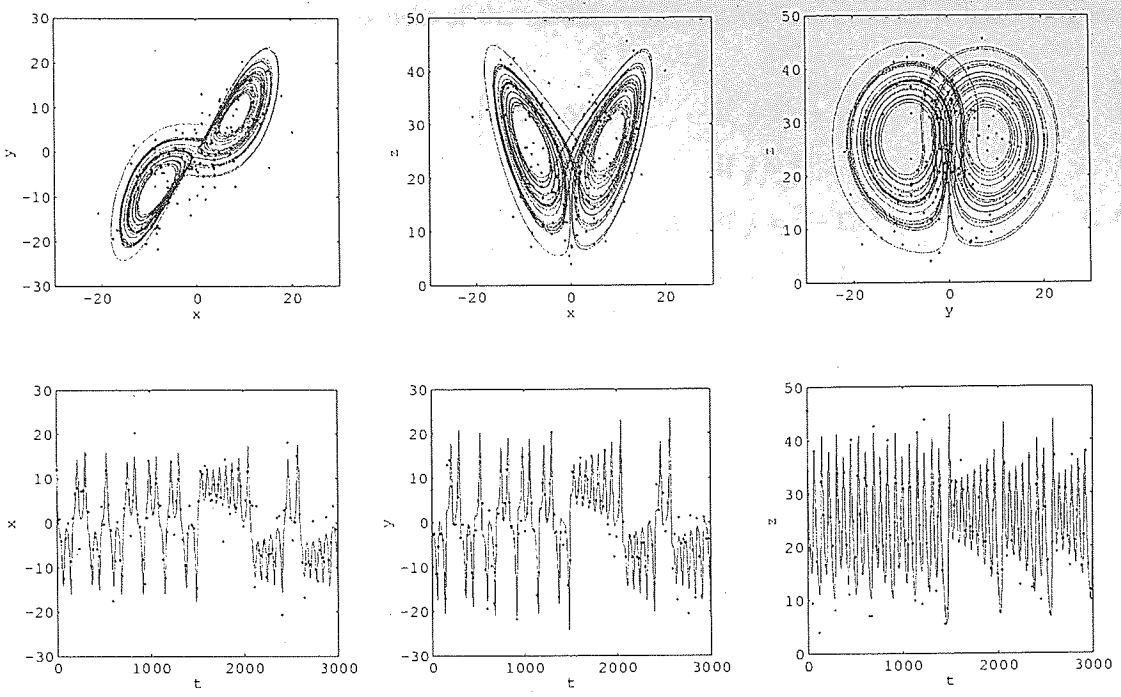


Figure 5.1: Lorenz 63 system: exact trajectory and observations (first 3000 time steps). Top row shows the spatial trajectory in each of the 3 planes. Bottom row shows the temporal evolution along each of the system's dimensions. Black dots represent the 6-hourly noisy observed values.

The Lorenz 96 system, in its original version, involves a set of N variables x_i, whose evolution is governed by N differential equations, as follows:

$$\frac{\mathrm{d}x_i}{\mathrm{dt}} = (x_{i+1} - x_{i-2})\, x_{i-1} - x_i + F. \tag{5.4}$$

The i index is cyclic, i.e. $x_{i+n} = x_{i-n} = x_i$. The number of variables is set to 40, so that $\mathbf{x} = (x_1, ..., x_{40})$. The x_i can be thought of as representing a weather variable at locations situated around the equator. The F term acts as a forcing term. A value of $F = 8.0$ is chosen, according to Lorenz and Emanuel (1998).

Tangent Linear Model

Like for the Lorenz 63 model, propagation of the state is achieved through a Runge-Kutta solver. The tangent linear model, required by 4D VAR and the EKF, is computed after application of Euler's method:

$$m_i(\mathbf{x}) = x_i + \Delta t\,[\,(x_{i+1} - x_{i-2})\, x_{i-1} - x_i + F\,], \tag{5.5}$$

where m_i denotes the i-th component of the model. The tangent linear model is band diagonal, with:

$$\hat{\mathbf{M}}_{i,j} = \frac{\partial m_i}{\partial x_j}(\mathbf{x}) = \begin{cases} x_{i-1}\,\Delta t & \text{if } j = i+1, \\ 1 - \Delta t & \text{if } j = i, \\ (x_{i+1} - x_{i-2})\,\Delta t & \text{if } j = i-1, \\ -x_{i-1}\,\Delta t & \text{if } j = i-2, \\ 0 & \text{otherwise.} \end{cases} \tag{5.6}$$

where the time index has been dropped to keep the notations clearer.

Observed data

The Lorenz 96 system was set-up as in Lorenz and Emanuel (1998). The time unit is assumed to be 5 days, and a short integration time-step of 1 hour ($\Delta t = 0.0083$) was used in order to minimise linearisation errors (for comparison, Lorenz and Emanuel (1998) use a time step of 0.05 (6h); Orrell (2003) uses a time step of 0.005 (36 minutes)).

Observations are generated in a similar fashion to the Lorenz 63 system, except the total duration of the experiment is 4800 time steps (200 days).

A preliminary study on the Lorenz 96 system involved running the different assimilation methods with (proportional) observation errors corresponding to low noise ($\sigma^2 = 0.01$), medium noise ($\sigma^2 = 0.05$), and strong noise ($\sigma^2 = 0.2$). Figure 5.2 shows the Root Mean Square Error (see Section 5.1.4) averaged over the assimilation period (200 days) for all selected methods. All these methods show a decrease in performance as observation noise increases, but all noise levels give similar results as far as comparative performance is concerned. As a consequence, a single noise value of 0.01 is considered, in order to keep the number of assimilation runs to a minimum.

5.1.3 Experiment set-up

The experiment looked at how the different parameters involved in the assimilation (noise level, assimilation frequency, method parameters) affect each method's performance. The parameters involved are summarised in Table 5.2. Details about the specific choices for each parameter are given below.

"Burn-in" phase

In order to obtain a coherent first guess estimate for the state's distribution and get rid of initial convergence issues, a "burn-in" phase is applied prior to the assimilation. During this "burn-in" phase, an EKF is run for a certain duration, chosen identical to that of the assimilation phase. The

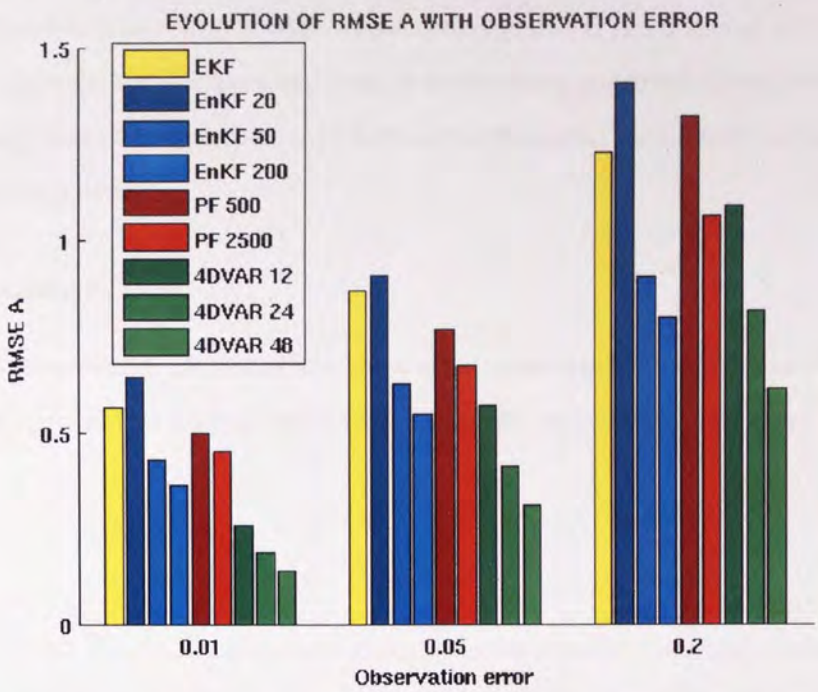


Figure 5.2: Preliminary study on the effect of observation noise on assimilation performance. The Root Mean Square Error (RMSE) between the model and the "true trajectory" is plotted for different observation noise levels (assimilation frequency: 1h).

last estimate of the state's probability density function is used as a first guess for the assimilation phase.

Assimilation phase

Each assimilation method is run on the Lorenz 63 system for a duration of 10000 time steps (104 days) and 4800 time steps (200 days) on the Lorenz 96 system.

Model error

Although a perfect model setting is used (i.e. the same model is used to generate the observation and conduct the assimilation), model error is added to reproduce standard assimilation conditions. Many examples in the literature use the perfect model and add model error. We follow this approach but would like to point out that this does not seem a consistent approach since, by definition, no error should be taken into account when using the exact model. A more consistent approach would involve using a perturbed model in the assimilation (e.g. using slightly different parameter values).

The model error is assumed constant, white, and Gaussian during the assimilation. The covariance matrix $\mathbf{Q}$ is chosen proportional to the amplitude of the "true trajectory" (computed as the covariance of the process over the whole experiment duration). The proportion is set to 0.01.

No correlation across dimensions is taken into account, so that $\mathbf{Q}$ is a diagonal matrix.

Note that no model error is taken into account in the strong constraint formulation of 4D VAR, which in consequence can be expected to perform better than other methods as long as linearisation errors remain negligible.

Assimilation frequency

The impact of assimilation frequency (i.e. how often observations are assimilated), is the most important parameter in this study. Chosen frequencies are: every 15min and every 1, 3, 6, 12, 24 and 48h.

Sample size

In the EnKF and the PF, the size parameter controls how accurately the state's probability density function is estimated. For the EnKF, retained sizes were: 3, 10, 20, and an additional 2000 for the higher-dimension Lorenz 96.

Particle Filters with 50, 200 and 2,000 particles were run on the Lorenz 63 system. The initial runs of the PF on the Lorenz 96 system showed that filter divergence would occur systematically for less than 500 particles. Since the PF is expensive to run with large number of particles, we restricted the study to 2,000 and 20,000 particles.

Time-window

In 4DVAR, the time window controls the method's behaviour. Fitting over a small time window will provide better results locally, whereas fitting over a large time window will provide better tracking of the system. In order to be able to measure the balance between those two opposite factors, several window lengths were considered: 1 hour (single observation), 6 hours, 12 hours, 24 hours and 48 hours.

Resampling rate

The Particle Filter uses a Systematic Resampling step. Resampling is triggered when the proportion of effective particles, as estimated by Equation (4.83), reaches below a certain threshold. Several thresholds were tested: 10%, 50%, 90% and 100% (systematic resampling) of the total number of particles but no clear pattern emerged which could have suggested an influence of this parameter on the filter's performance. As a consequence, all Particle Filters in the results discussed below were run with a resampling rate of 50%.

	Lorenz 63	Lorenz 96
Time step	0.01	0.0083
Time step interpretation	15min	1h
Burn-in phase	10000 steps (104 days)	4800 steps (200 days)
Assimilation phase	Same as burn-in phase	
Observation error	0.01 of the system's covariance	
Assimilation frequency	15min, 1h, 3h, 6h, 12h, 24h, 48h	
Ensemble size (EnKF)	3, 10, 20	20, 50, 200, 2000
Number of particles (PF)	50, 200, 2000	2000, 20000
Resampling rate (PF)	10%, 50%. 90% and 100% of the population	
Window length (4D VAR)	6h, 12h, 24h, 48h	

Table 5.2: Summary of parameters choice for the Lorenz 63 and Lorenz 96 systems

5.1.4 Lorenz 63 Results

Performance measure

As a measure of each method's performance for a given set of parameters, we use the Root Mean Square Error (RMSE) between the true state $\mathbf{x}^t$ and the estimated state $\mathbf{x}$:

$$\text{RMSE}(\mathbf{x}, \mathbf{x}^t) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(x_i - x_i^t)^2} \tag{5.7}$$

where N is the dimension of the state. The RMSE is averaged over the whole experiment and, for filters relying on sampling (EnKF, PF), over 5 runs of the filter, in order to minimise sampling error.

Although it was initially intended to compare the methods' forecasting skill (measured using the RMSE over a post-assimilation single forecast), it became clear that the quality of the forecast depended entirely on how close the state's estimate at the end of the assimilation phase was to the true state. A good measure of forecast skill would have needed averaging over many forecasts, however this wasn't done in the current experiment. Furthermore, it is well acknowledged that the RMSE isn't an optimal measure of forecast skill for probabilistic forecasts. Another more appropriate method, based on Receiver Operating Characteristics curves, will be introduced and applied later in this work (Section 7.3.2).

Figure 5.4 summarises the different methods' performance on the Lorenz 63 experiment. The RMSE is plotted against the assimilation frequency (in log-scale).

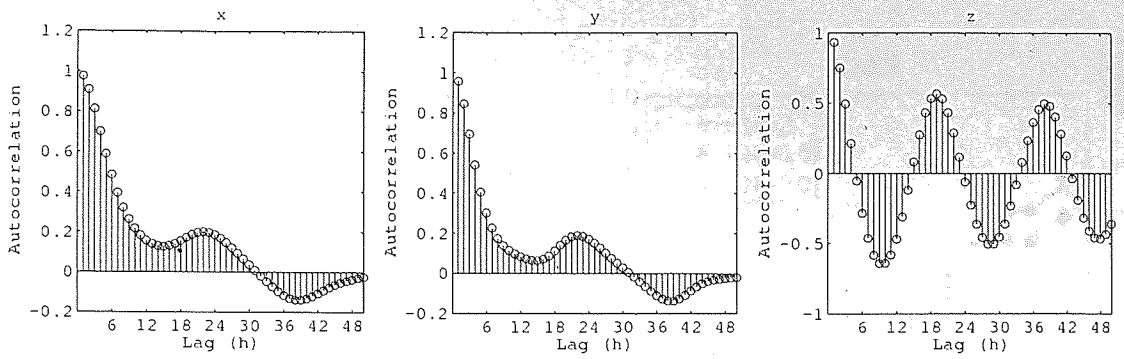


Figure 5.3: Autocorrelation of Lorenz 63 system

Assimilation performace

- **About model error** – The inclusion of model error in a perfect model setting is clearly unrealistic. The only source of error in this setting is the incorrect specification of the initial state. As a result, model error effectively acts as an additional forcing, and can be responsible for taking the state away from the true process, rather than providing the necessary flexibility to bring it closer to the process, as would be the case with an imperfect model. In consequence, some reserve must be shown with regard to the conclusions drawn from this experiment.

- **Effect of assimilation frequency** – The first general comment to be made when analysing the results shown on Figure 5.4 is that, as one would expect, the quality of the assimilation decreases as observations are assimilated at larger and larger time intervals, regardless of which assimilation method is used. This is due to the fact that, for a fixed duration, fewer assimilation steps mean fewer updates of the state to compensate for model divergence. All methods perform well when observations are assimilated frequently, with a slight, almost linear increase in the error up to 3h time intervals. In the range 3h to 12h, the error growth increases in a quadratic fashion (in log scale) but seems to go back to a more linear growth rate in the range 12h to 48h. We can expect the error to eventually converge to a maximum value for assimilation frequencies higher than 48h (since the process is bounded in space).

- **Linearity regimes** – Figure 5.3 shows the autocorrelation plots for each of the 3 dimensions. If we discard the z dimension, which shows a pseudo periodical regime, 3 regimes can be observed. Strong autocorrelation in the lag region 15min – 3 suggests a smooth, linear regime. A sharp decrease in autocorrelation for lags between 3h – 12h corresponds to an increasingly non-linear regime. For lags above 12h, the process reaches a non-linear regime in which the autocorrelation fluctuates about zero (no correlation). These 3 regimes explain the 3 phases observed on the error plots on Figure 5.4.

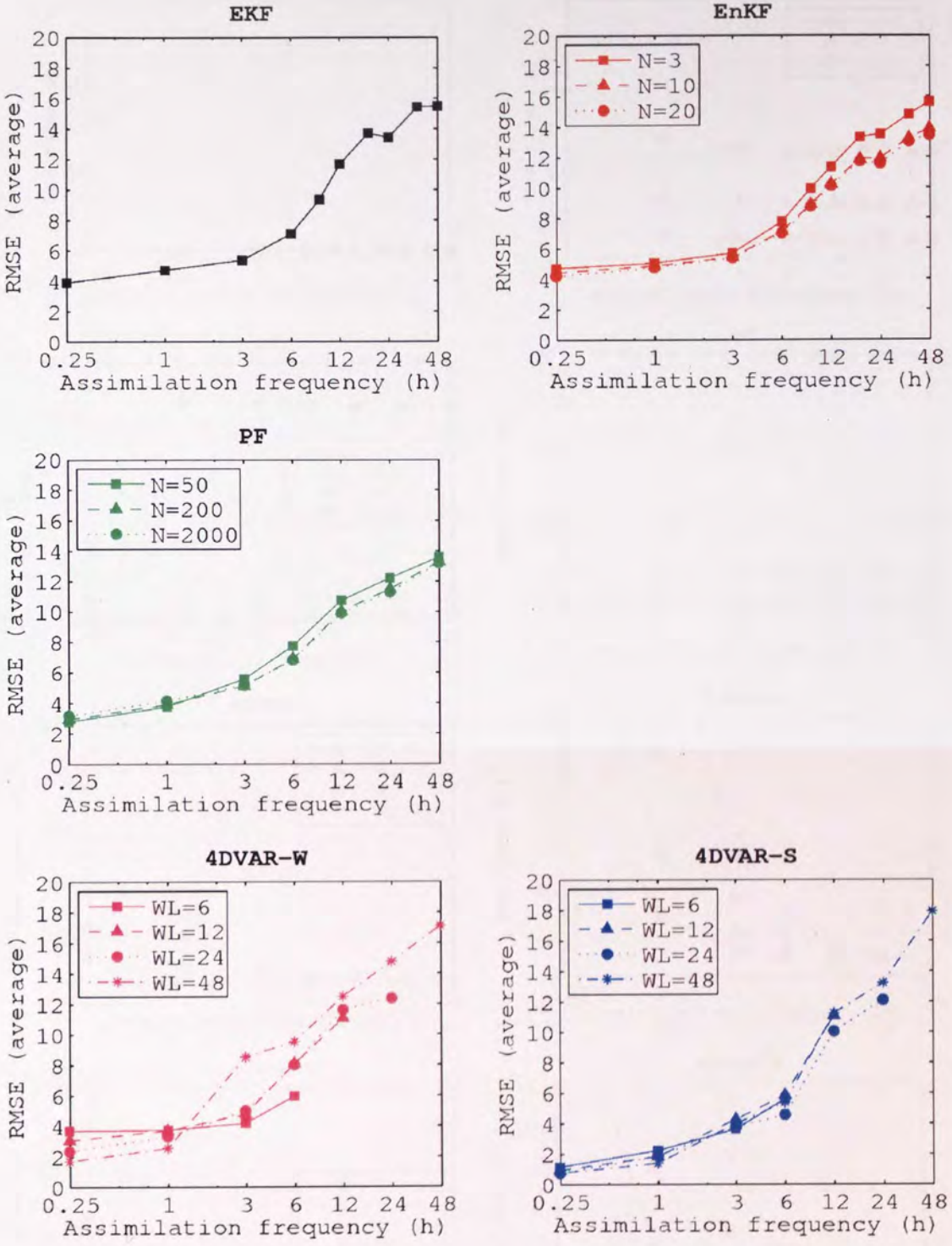


Figure 5.4: Lorenz 63 system – RMSE for different assimilation methods and parameters, as a function of assimilation frequency (log-scale). From top to bottom, left to right: Extended Kalman Filter (EKF); Ensemble Kalman Filter (EnKF) with 3, 10 and 20 ensemble members; Particle Filter (PF) with 50, 200, 2000 particles; 4D VAR strong constraint (4DVAR-S) with windows of length 6h, 12h, 24h and 48h; 4D VAR weak constraint (4DVAR-W) with windows of length 6h, 12h, 24h and 48h.

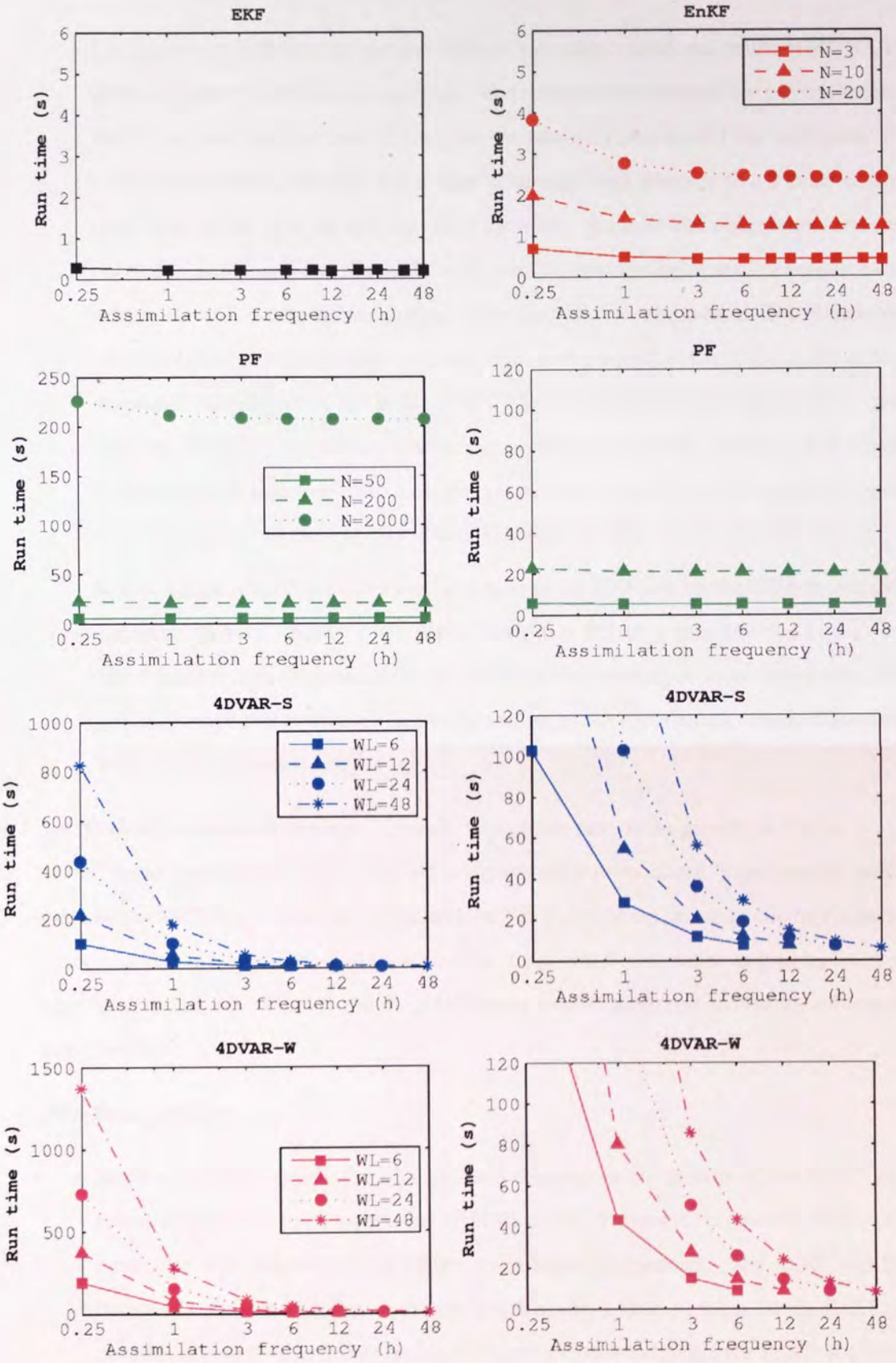


Figure 5.5: Lorenz 63 system – Run time (seconds) for different assimilation methods and parameters, as a function of assimilation frequency (log-scale). The legend is the same as on Figure 5.4. Results for the Particle Filter, 4D VAR strong and weak constraint are shown at 2 different scales.

- **Linear regime**
  - For lags below 3h, where the process behaves smoothly, variational methods (4D VAR) generally outperform filtering methods. The smooth behaviour of the process means that the approximations used to compute the tangent linear model are negligible. In such circumstances, 4D VAR has a clear advantage over filters in that it finds an optimal path rather than an optimal point estimate. Because the observation error is unbiased (zero mean) and the model is known, by considering several observations at a time, 4D VAR is able to "average out" the effect of observation error. The variational solution is thus expected to be very close to the true process, in particular in the strong constraint formulation of 4D VAR (4DVAR-S) in which no model error is taken into account. Although we did not consider smoothing assimilation methods, it is worth mentioning that since they also look for optimal paths, they could be expected to give results equivalent to those observed with 4D VAR.
  - In this regime, the PF provides similar results to the EKF and the EnKF, with a slight advantage at very frequent assimilation frequency (15min), possibly due to the fact that it handles non-Gaussianity in the distribution (however, at short time scales, the process is expected to behave linearly, hence the state's distribution, chosen Gaussian, is expected to remain such.)

- **Increasingly non-linear regime** – Overall, a quadratic rate of the growth in RMSE is observed for all methods. However, this rate is slightly more pronounced for variational methods and the EKF than it is for the EnKF and the PF. This is most certainly due to the linear approximation involved in both 4D VAR and the EKF, which only holds as long as the process remains smooth. The PF and the EnKF seem overall more resistant to the effects of non-linearity.

- **Non-linear regime**
  - In the non-linear regime, the EKF shows a decrease in the growth of the RMSE and seems to converge to a stable value (RMSE $\approx$ 16). However, this would need to be confirmed with experiments at higher assimilation frequencies. The EnKF and PF both reach a linear RMSE growth rate, which seems a little stronger for the EnKF. A possible reason for that is the Gaussian assumption on which relies the EnKF. A Monte Carlo experiment would show that samples part to orbit around one attractor or the other, eventually leading to a multimodal posterior distribution. Hence the Gaussian assumption is not realistic in the non-linear regime and the EnKF can be expected to

show more sensitivity to non-linearity than the PF.

- For assimilation every 12h and beyond, 4D VAR still shows a steep increase in RMSE, reaching the highest error of all methods (RMSE > 17). This is most probably due to the fact that the linearisation errors occurring in the computation of the tangent linear model add up as the length of the window increases. Thus, the optimisation of the cost function is performed using an incorrect gradient, leading to a solution which no longer approximates the minima of the cost function.

- **Effect of parameters**

  - **Ensemble Kalman Filter** – 3 ensemble sizes were considered: 3 (the dimension of the system), 10 and 20. A clear improvement can be observed with 10 ensemble members over 3 ensemble members, especially as the process becomes non-linear. However, there is little gain in taking the ensemble size to 20, which suggests that 10 ensemble members are sufficient to capture the first and second moments of the state's distribution.

  - **Particle Filter** – A similar observation can be made for the PF. The filter was tested with 50, 200 and 2000 particles. There is a slight improvement when using 200 particles over 50 particles, mostly in the transition regime between linear and non-linear, but no significant change is observed when taking the number of particles to 2000.

  - **4D VAR** – 4D VAR was run with 4 different lengths of time-window: 6h, 12h, 24h and 48h. In the linear regime, as one would expect, increasing the time-window improves the quality of the assimilation, as more observations are assimilated. However, this phenomenon is reverted when the process becomes non-linear. Longer time windows imply larger errors in the tangent linear model (since linearisation errors add up in time) and the best time-window in the linear regime become the worst in the non-linear regime. This is particularly noticeable for 4DVAR-W, for which the order is reversed between 1h and 3h assimilation frequencies. The change is less marked for 4DVAR-S, although for frequencies beyond 12h, the 48h time-window provides the highest RMSE.

Computation time

Figure 5.5 shows the experiment's computation time for each method/choice of parameter. Filters show almost constant run time which increases with the size of the ensemble/number of particles for EnKF/PF. This suggests that on this example, the assimilation phase has a negligible compu-

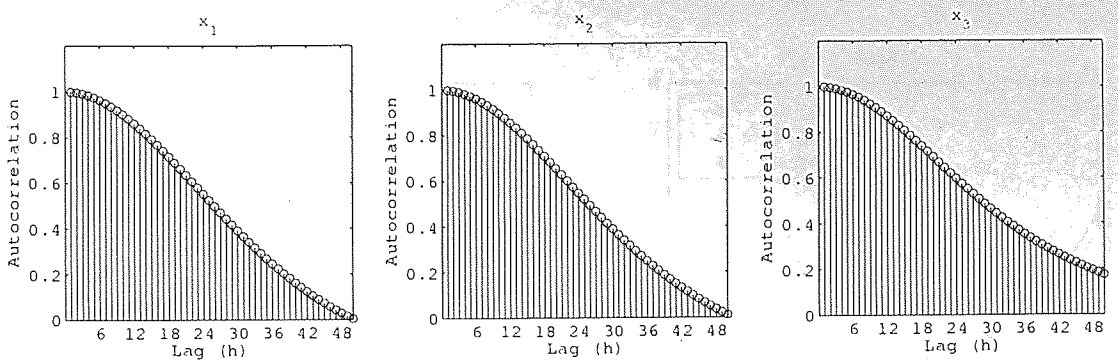


Figure 5.6: Autocorrelation of Lorenz 96 system (first 3 dimensions)

tational cost compared to the prediction phase. Only at assimilation frequencies below 3h can a slight rise in computation time be observed for the EnKF.

Variational methods, on the other hand, have their computation time strongly influenced by the number of observations assimilated, along with the size of the assimilation window. As expected, for a fixed assimilation frequency, longer assimilation windows mean more observations to be assimilated, with linear increase in the computation cost. For a fixed window length, an increase in assimilation frequency effectively means fewer observations to be fitted, resulting in a decrease of computation time. A quick evaluation suggests the relationship between computation time and assimilation frequency is inversely proportional but no rigorous analysis was performed to confirm this hypothesis.

5.1.5 Lorenz 96 Results

This section discusses the results of the experiment with the Lorenz 96 system. Figure 5.7 shows the assimilation performance of the different methods on the Lorenz 96 system, as measured by the RMSE to the true state. Figure 5.8 shows the corresponding computation times.

Assimilation performance

- **Linearity of the process** – Figure 5.6 shows autocorrelation plots for the first 3 dimensions of the Lorenz 96 system. Although a decrease in autocorrelation is observed, which can be related to an increase in non-linearity, the change is much slower at the time scales considered than it was for the Lorenz 63 system. Thus, the effects of non-linearity are expected to start being felt at the highest assimilation frequency only (every 48h).

- **Particle Filter issue** – The most noticeable feature here is the failure of the Particle Filter to keep track of the true system as soon as observations become too scarce. This is due to the dimension of the state (40) and the discrete nature of the PF, which in its original formulation

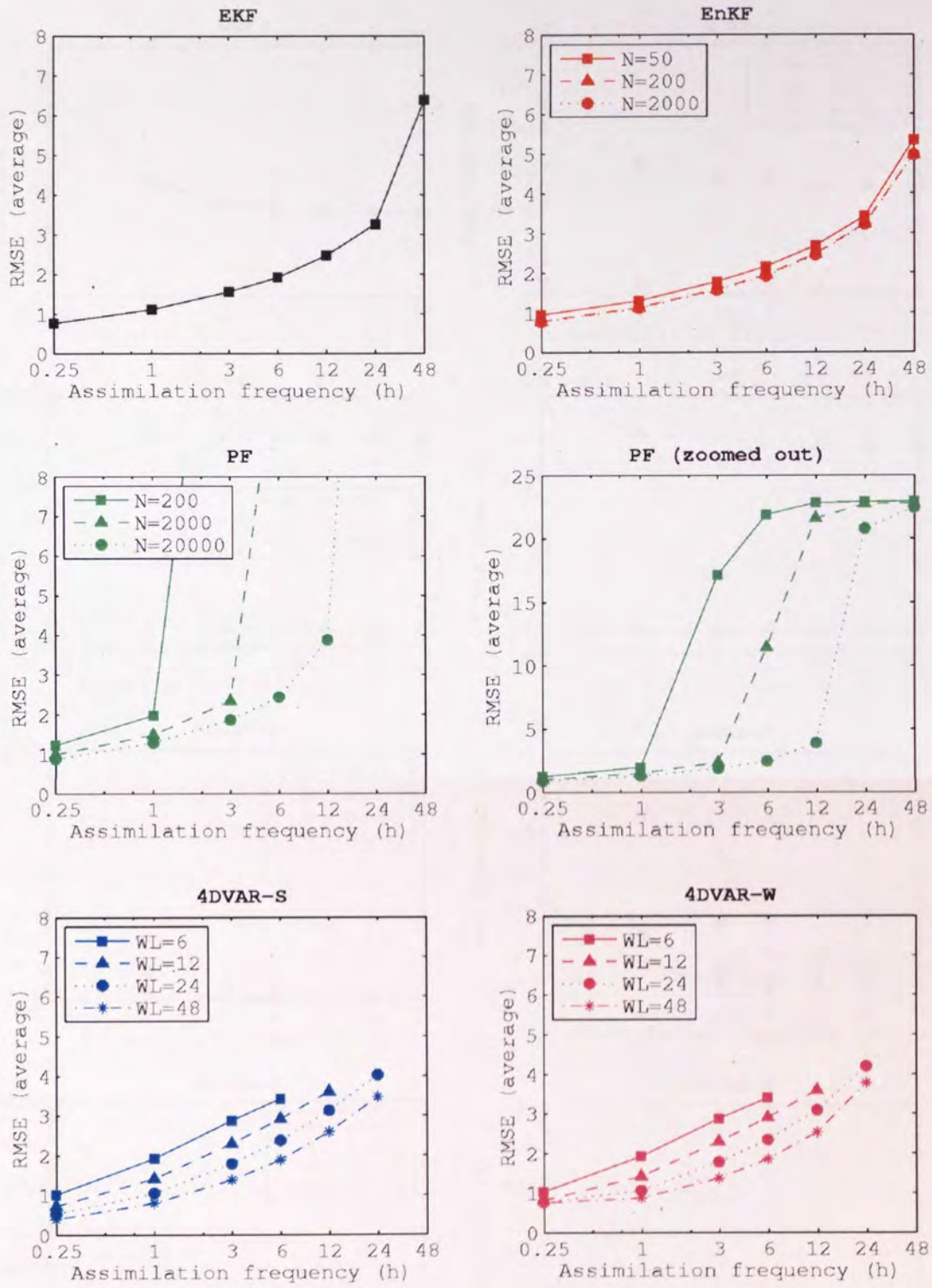


Figure 5.7: Lorenz 96 system – RMSE for different assimilation methods and parameters, as a function of assimilation frequency (log-scale). From top to bottom, left to right: Extended Kalman Filter (EKF); Ensemble Kalman Filter (EnKF) with 50, 200 and 2000 ensemble members; Particle Filter (PF) with 200, 2000, 20000 particles; Particle Filter (PF), zoomed out; 4D VAR strong constraint (4DVAR-S) with windows of length 6h, 12h, 24h and 48h; 4D VAR weak constraint (4DVAR-W) with windows of length 6h, 12h, 24h and 48h.

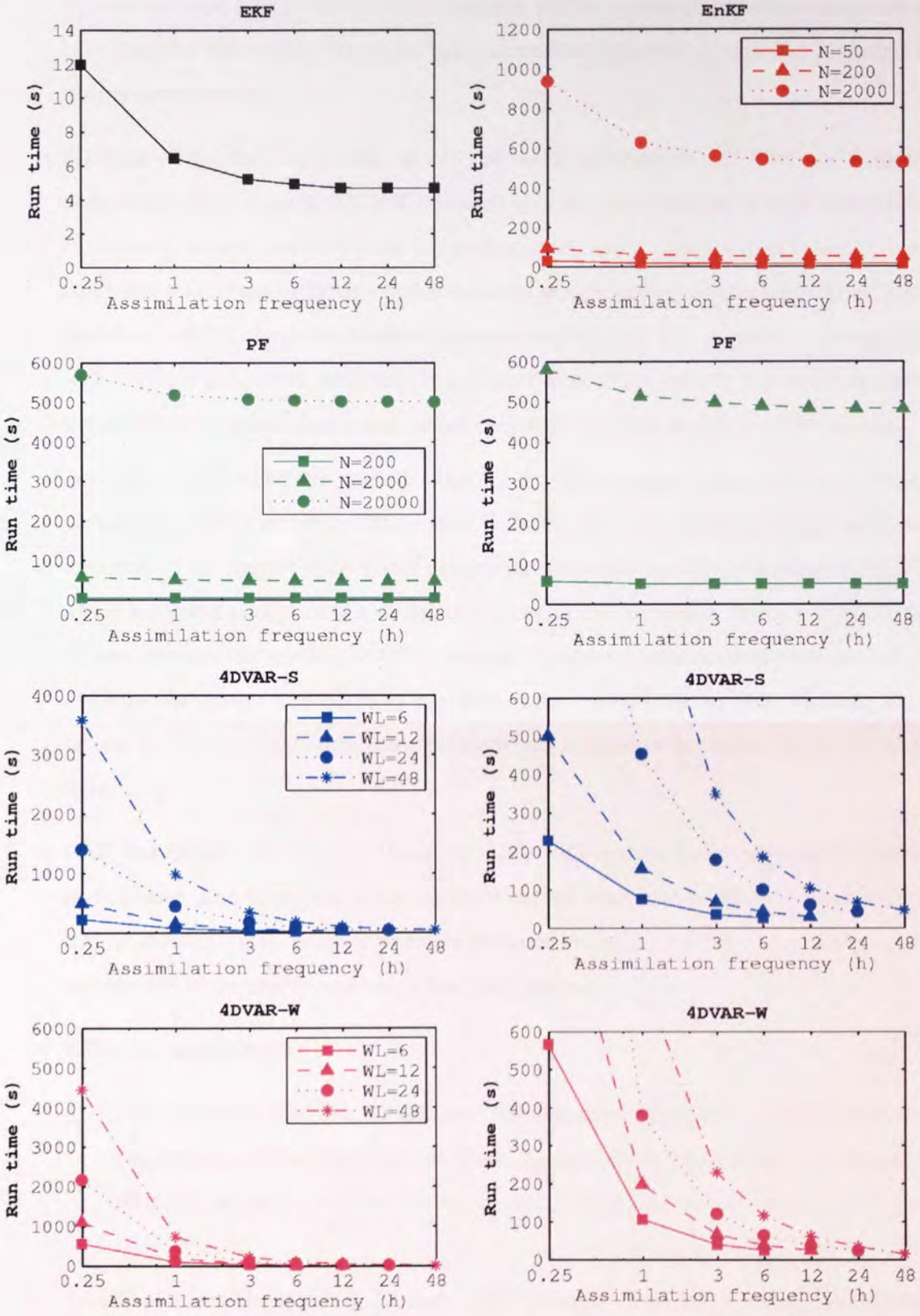


Figure 5.8: Lorenz 96 system – Run time (seconds) for different assimilation methods and parameters, as a function of assimilation frequency (log-scale). The legend is the same as on Figure 5.7. Results for the Particle Filter, 4D VAR strong and weak constraint are shown at 2 different scales.

doesn't have the ability to add particles in regions of high likelihood. Increasing the number of particles in the filter improves its performance, but the number of particles needed, which is exponential with respect to the state space dimension, becomes so large that computation time is unmanageable.

- **4D VAR** – The other filters (EKF, EnKF) and variational methods (4D VAR) show overall comparable performances. 4DVAR-S outperforms all other methods at high assimilation frequencies, but this is related to the use of the perfect model without added noise. Its weak constraint counterpart (4DVAR-W) shows similar though slightly worse results for all time-windows. 4D VAR methods also show a greater sensibility to the decrease in observations, with a steeper increase in error than the Kalman Filters (EKF, EnKF). This could suggest a greater effect of linearisation issues, which only seem to affect the EKF at 48h frequency.

  No result could be obtained with 4D VAR for the 48h frequency, due to numerical issues causing the method to break. This is most certainly due to the effect of non-linearity and the errors in the tangent linear model causing the minimisation of the cost function to fail. These numerical issues could be addressed, however, by improving the way in which the tangent linear model is computed. For instance, a midpoint Euler method could be used to discretise the system of differential equations rather than a standard Euler method. Also, second and higher order terms could be taken into account in the Taylor expansion when linearising.

- **EKF and EnKF** – In the range 15min – 24h, the EKF and the EnKF show almost similar performance, also equivalent to that observed for 4D VAR with a 48h time window. The RMSE shows a strong increase when observations are only assimilated every 48h, which corresponds to the process reaching a non-linear regime.

- **Effect of parameters**

  - The EnKF was run with 50, 200 and 200 ensemble members. A slight increase in performance, in the linear and non-linear regimes, can be observed when going from 50 to 200 ensemble members. Increasing that number further does not give any visible improvement.

  - The PF was run with 200, 2000 and 20000 particles. Although increasing the number of particles makes the filter more resistant to filter divergence, even 20000 particles are not sufficient to prevent filter divergence unless observations are assimilated frequently enough.

Computation time

As far as running time is concerned, the results are extremely similar to those obtained with the Lorenz 63 system, except for the scaling due to the higher dimension. Filters show almost constant run times, with a linear increase in the number of particles/ensemble members. 4D VAR computation times are affected by the number of observations taken into account in the cost function, which is proportional both to the length of the assimilation window and the assimilation frequency. Overall, the assimilation methods offering the best performance in terms of both accuracy and associated computation time seem to be the EnKF with 200 ensemble members and the EKF. However, as was mentionned earlier, the perfect model setting does not give a realistic account of the ability of data assimilation methods to capture a process in the presence of model error. The conclusions drawn here are thus to be taken with caution.

5.2 Implementation: a data-assimilation framework

As part of the work presented in this chapter, a data assimilation framework was developed in the C++ programming language. This section gives a brief description of the framework. More detail is provided in Appendix B.

5.2.1 Motivation

The data assimilation methods presented in this work were initially implemented in MATLAB®. However, this implementation suffered from two major drawbacks. First, although MATLAB® is particularly adapted to quick development and testing of algorithms (made easy by powerful visualisation tools), computation time does not scale well with problem complexity. Experiments involving large amounts of high-dimensional data can take up to weeks to run.

Second, there is very limited support in MATLAB® for Object Oriented Programming (OOP)[1]. OOP organises the code around meaningful entities (objects), putting together the data and tasks associated to it. For instance, in OOP, a probability density function object would consist in the distribution's parameters (e.g. mean, covariance if it Gaussian) and usual tasks performed with the distribution (setting the parameters to some value, sampling from the distribution...). OOP also benefits from the ability to have objects "extend" other objects by "inheriting" some of their features. For instance, consider one has written the code for a rectangle object. This includes two parameters (length and height) and a function to draw the rectangle on the screen. If one wanted to use a separate square object, and were to write it separately, much of the code would be duplicated.

[1] The version of MATLAB® used at the time of development was 6.5 Release 13. At the time of writing, version 7 has been released, which provides improved support for OOP.

Instead, one could implement the square as a particular case of the rectangle where the length and the height are equal. By treating the square as a specific rectangle, one would then automatically inherit from the ability to draw the square without having to rewrite it. Inheritance in OOP is a very powerful mechanism which reduces the amount of code to be written (and the associated risk of error), allows reusability, and facilitates maintenance of the code.

In data assimilation, most problems deal with long series of high-dimensional data. For example, the model discussed in Chapter 6 deals with series of about 700 radar images of dimension $100 \times 100 = 10^4$ pixels. Typical meteorological applications can easily require up to 10^6 inputs. For this reason, computation speed is a critical issue. The data assimilation problem is also particularly suited to the object-oriented approach, with clear distinct entities (systems, assimilation methods, probability density function...), some of which sharing very similar functionalities.

These considerations have motivated the translation (or, rather, rewriting) of the initial, MATLAB® based, data assimilation framework to the fast, object-oriented, C++ programming language.

5.2.2 Design considerations

The aim of the framework is to provide a set of tools to automate data assimilation experiments. Attention has been paid to reusability, and the framework was implemented in a modular fashion, so that end users can develop their own data assimilation methods as "plug-ins" and use them within the framework in a transparent way. Basically, the framework provides the skeleton (interface) for data assimilation experiments, and implements some of the data assimilation methods discussed in Chapter 4.

5.2.3 Features

The framework provides the basic interface and implementation for the following components:

- **Dynamical models:** Lorenz 63, Lorenz 96, autoregressive model.
- **Observations** include support for simulation, saving to and reading from files, accessing observations by index.
- **Data assimilation methods:** Extended Kalman Filter, Ensemble Kalman Filter, Particle Filter (with Systematic Resampling), 4D VAR strong constraint, 4D VAR weak constraint, and a couple of Kalman smoothers which were not discussed in this work.
- **Probability distributions:** Support for Gaussian and diagonal Gaussian distributions has been implemented. Further support for Gamma and Inverse-Gamma distribution was added later on.

- **Optimisation:** A single optimisation method has been implemented, namely the Scaled Conjugate Gradient algorithm. However, further support for additional optimisation methods would be an interesting extension.

The framework uses the Matpack library (Gammel, 2005) interfaced with the BLAS library for matrix computations. Further detail is provided in Appendix B for the interested reader.

5.3 Conclusions

Several data assimilation methods have been presented and implemented on two benchmark models: the 3-dimension Lorenz 63 and the 40-dimension Lorenz 96 systems. Results confirmed the intuition that frequent assimilation leads to better results, independent of what assimilation method is used.

At high assimilation frequency, variational methods perform slightly better than their filtering counterparts, because of their smoother-like approach (they look for an optimal trajectory of the state over a fixed time window rather than a single optimal state, like filters do). However, at lower assimilation frequencies, they suffer from linearisation issues due to the use of the model's tangent linear in the cost function. The Extended Kalman Filter is also sensitive to similar issues, though these seemed less pronounced on the models selected in this study. Although outperformed by variational methods at high assimilation frequencies, Ensemble Kalman Filters are the most robust at low assimilation frequencies and offer the best trade-off between accuracy and computation time.

6 Bayesian precipitation nowcasting: Theory

CONTENTS

6.1 Introducing precipitation nowcasting

6.1.1 Definition and motivation

Short term forecasts of precipitation fields (*nowcasts*) are of critical importance to hydrologists who rely on them to prevent floods (Sun et al., 2000; Moore et al., 2005; Smith et al., 2007) or manage sewage systems in real time (Reed et al., 2007; Pfister and Cassar, 1999; Vieux and Vieux, 2005). Forecasting models for flooding need to have sufficient resolution to resolve the local, fast developing processes involved in the development of convective storms. According to Golding (2000); Einfalt et al. (2004); Berne et al. (2004), precipitation run-off models could require forecasts down to 0.5-3 km spatial resolution and 1-5 min temporal resolution.

Traditional numerical weather prediction (NWP) models rely on complex systems of equations replicating the physics of the atmosphere and are usually run at resolutions too coarse to properly capture precipitation patterns. For instance, Golding (2000) reports that "the Met. Office mesoscale model" runs at "a minimal scale of 50 km". However, with the advances in computer power, resolution might not be an issue for long. The recent developments discussed in Lean et al. (2008) show that the UK Unified Model can already be run at resolutions down to 1-4km, with an improvement in the forecasting of precipitation events.

Yet, there are two more important issues encountered when trying apply NWP to short term precipitation forecasting. The first is the existence of a *spin up* phase during which the model's performance is generally poor, due to an incomplete representation of the state of the atmosphere at initial time, particularly at the scales of interest for precipitation nowcasting. It is thus difficult to obtain good NWP forecasts at the times that are most relevant to hydrological applications. A second issue with the application of NWP to the prediction of convective precipitation is our limited understanding, and hence formulation, of the physical processes involved in the development of convective showers. Efforts are being invested to better understand these processes (Browning et al., 2007) and will hopefully lead to a better formulation of precipitation processes in NWP models.

For the reasons mentionned above, it is usually well acknowledged that NWP models are not optimal for precipitation forecasting below 3h (Golding, 1998, 2000), where they are outperformed by simpler, data driven nowcasting methods (a review of such methods is given later). Figure 6.1 is taken from Golding (1998) and provides a qualitative illustration of the comparative performance of NWP models (dashed line) and nowcasting models (dotted line) as forecast lead time increases. The theoretical limit of predictability is indicated by a solid line. In the range 0-3h, nowcasting methods provide better forecasts that NWP models, but their forecasting skill decreases much faster with time. This fact is also illustrated for a particular example in a recent study by (Lin

et al., 2004), where the operational mesoscale Canadaion model GEM (CotÃl' et al., 1998) is compared with the radar nowcasting method from Germann and Zawadzki (2002). The authors show that the radar nowcast presented better forecasting skill for up to 7h forecasts.

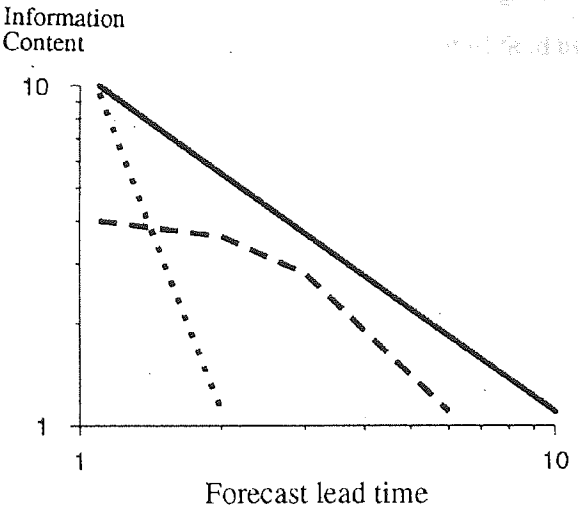


Figure 6.1: "Schematic representation of the loss of information content in forecasts as a function of lead time. The solid line represents the theoretical limit of predictability. The dashed line represents NWP models and the dotted line nowcasting methods." (Golding, 1998)

Therefore, alternative approaches have been developed over the last couple of decades. Constrained to run fast and at high resolution, such models cannot rely on physics in the same way NWP models do. The physical equations driving the processes of interest have to be approximated in some way. There are many different ways to model rainfall, ranging from image processing techniques to stochastic modelling approaches. A short review of some models found in the literature is given in Section 6.1.2.

An immediate consequence of the approximations performed in nowcasting models is their limited forecast skill. Precipitation nowcasts typically become very poor beyond lead times of order 30-60 min, when the characteristic features of the rainfall generating process take over. To prevent nowcasts models from diverging, as they would if used in a purely generative way, some control has to be performed to ensure they remain consistent with the real process. This is achieved by using observations of the rainfall field to recalibrate the model (i.e. assimilation of the state) in real time.

With the advent of weather radar, frequent measurements (every 5-15 minutes) with high spatial resolution (1-5 km) have become available to the forecasting community (Collinge, 1987). Such characteristics have quickly made radar data the favourite source of control for nowcasting models, greatly influencing the range of techniques developed. Although there are other sources of precipitation measurement available (rain gauges, airborne sensors, satellites), this chapter only

discusses radar-based precipitation nowcasting.

6.1.2 A review of radar nowcasting methods

There have been two main trends in the development of nowcasting methods. On the one hand, extrapolation methods predict the evolution of the observed rainfall field using object tracking and advection-based techniques, while on the other hand, storm generating methods focus on the birth, growth and dissipation of storms. Both approaches are equally valid, and often complementary. As a result, most recent nowcasting systems tend to operated a blend of both methods. We present a quick review of such methods. The reader is referred to Wilson et al. (1998), Burton and O'Connell (2003) and Pierce et al. (2004) for more extensive reviews of nowcasting models.

Extrapolation methods

Extrapolation-based nowcasting methods rely on the fact that at the prediction time of interest, the evolution of the rainfall field is largely governed by flow conservation laws. In other words, changes in the observed rainfall field can be attributed to motion only (internal dynamics are assumed negligible or following simple patterns). Motion fields are often inferred from consecutive radar scans using cross-correlation techniques. Additional wind-field forecasts from NWP models may be blended in for improved motion estimates. We proceed below to a short review of the most representative advection-based nowcasting systems in the literature.

NIMROD The NIMROD system in the UK (Golding, 1998, 2000) is a hybrid system merging a nowcasting model with a mesoscale numerical model. NIMROD provides a range of products including forecasts of precipitation, clouds and visibility. Composite radar data at 5km resolution is incorporated every 15 min after having undergone an automatic correction and enhancement procedure. Note that past records of this radar data have been made available by the British Atmospheric Data Centre and are used in the current work (see chapter 7).

Precipitation forecasts are based on advection of pre-identified rain objects. These rain objects are identified as series of contiguous rainy pixels of intensity above a given threshold. Motion vectors are updated for each object so that the correlation between the object and the resulting propagated estimate of the previous rainfall field is maximised. For longer term forecasts (> 1h), wind field estimates from the NWP model are also computed and compared with the motion vectors. The best estimate of the two is selected based on a maximum correlation criterion.

The propagation algorithm uses the updated motion vectors estimate to generate a trajectory for each pixel. A weighted average of the propagated pixels falling within the same grid location is

computed. The resulting nowcast is then corrected using the NWP corresponding forecast. Greater importance is given to the advection nowcast at shorter lead times through appropriate weighting.

TITAN The TITAN (Thunderstorm Identification, Tracking, Analysis, and Nowcasting) system (Dixon and Wiener, 1993) is an object tracking system in which precipitation objects are detected using 3-dimensional radar volume scans. Storms are identified as sets of adjacent rainy pixels and approximated by elliptic centroids. To match the centroids from one observation to the next, a cost function is devised as a weighted sum of the distance between current and previous storms and the difference in volume for the same cells. Combinatorial optimisation is used to find the best match. The TITAN system handles merging and splitting of cells based on overlapping of forecast previous storms and new estimates. If several small storms have forecasts lying within the boundaries of a newly identified larger storm, a merging is detected. Similarly, if several newly detected small storms lie within the boundaries of a larger forecast storm, then a splitting has occurred.

Forecasting storms is done in two ways. Storms that are newly detected and have no past trajectory are considered unchanged. Storms which have been tracked over several time steps have their parameters (position, volume, ...) propagated linearly according to the trend. The trend is identified using linear regression over the parameter's history. Further detail about the handling of mergers and splitters in the forecast are provided in (Dixon and Wiener, 1993).

GANDOLF Bringing further the concept of identifying physical components of the rain field, such models as Gandolf (Pierce et al., 2000; Golding, 2000) consider individual rain cells with associated life cycles. The rain objects grow and decay according to a conceptual life cycle model (Hand and Conway, 1995), while being advected by wind fields estimated by an NWP model. Nimrod, Titan, Gandolf and other operational nowcasting systems are discussed into more detail in the context of the Sydney 2000 forecasting project in Pierce et al. (2004).

TREC/COTREC The TREC (Tracking Radar Echoes by Correlation) algorithm (Rinehart and Garvey, 1978) uses correlation methods to determine motion vectors from two consecutive radar scans $\mathbf{y}_1$ and $\mathbf{y}_2$. Rather than identifying rain objects and matching them on both radar scans to compute their velocity, the TREC method splits each scan into a series of fixed size "boxes" and matches the boxes from $\mathbf{y}_1$ with the boxes from $\mathbf{y}_2$ according to a maximum correlation criterion. Li et al. (1995) reported issues with the original TREC method due to ground clutter or incorrect tracking of reflectivity patterns, resulting in unrealistic, non smooth velocity fields. To address this issue, they devised the COTREC (Continuity of TREC) algorithm which addresses the problem in two steps. First, unrealistic vectors (zero velocity or divergent direction) are replaced with

an average of the neighbouring vectors. Second, smoothness of the vector field is enforced by minimising the divergence between neighbouring vectors. The minimisation is achieved using a variational approach (minimisation of a cost function).

NCAR Auto-Nowcaster The NCAR Auto-Nowcast system is a complex nowcasting system incorporating a variety of datasets including: radar, lightning and satellite data, wind profiles and NWP model outputs (Mueller et al., 2003). The Auto-Nowcast system makes use of several algorithms to process these datasets into "predictor fields" (e.g. fields of reflectivity, storm growth/decay, accumulated precipitation...). Amongst the algorithms involved, the TREC method is applied to wind field retrieval from radar data and the TITAN algorithm is used to detect storms and estimate trends in their evolution. A boundary detection algorithm, coupled with a cloud detection system, allow for convergence lines, where storms are expected to occur, to be identified. A physically-based model using fuzzy logic algorithms merges the various predictor fields into a dimensionless likelihood field which is then post-processed to obtain a final prediction estimate.

Cascade and fractal-based models

In the 1980s, empirical studies have shown that rainfall fields exhibit statistical invariance with respect to the scale at which they are observed (Lovejoy and Mandelbrot, 1985; Gupta and Waymire, 1990) and, as such, could be modelled using random cascade models. In such models, the field at a given scale (i.e. spatial or temporal resolution) can be decomposed into a field at lower scale (higher resolution) by splitting unit regions into equal subregions. The intensity of the field at these subregions is determined through a random scaling which conserves the overall statistical characteristics of the field. Various options for the distribution of the scaling are given in Tessier et al. (1993).

The cascade methodology is naturally expressed using multifractal theory(Tessier et al., 1993; Lovejoy and Schertzer, 1995) and presents several advantages over object tracking methods: it models the rainfall field at different scales, allows seemless incorporation of data at various resolutions (fine for radar, coarser for NWP estimates) and in the universal multifractal framework described in Tessier et al. (1993), only requires a very small number of parameters. Dynamics in these models can be modelled by including time (along with space) in the cascade (Deidda, 2000; Bocchiola and Rosso, 2006) or using advection based propagation as in systems like S-PROG(Seed, 2003) and STEPS(Bowler et al., 2006).

However, a study by Veneziano et al. (2006) underlines the limitations of multifractal models and shows, using several datasets, that the common assumption of scale invariance (the characteristics of the rainfall field show similar properties at all observed scales) often leads to unrealistic

simulation results. The authors conclude that "rainfall does not behave like a multifractal process" and that "multifractal models (...) are inadequate", and suggest directions for a new generation of multifractal models.

Point process methods

Point process methods focus on reproducing the internal dynamics of storms and provide models for the occurrence and life of precipitation cells and storms.

LeCam (1961) introduced a formalism for precipitation models in which the rainfall field is treated as a probabilistic process. In this work and studies that followed, the occurrence in time of simple precipitating objects (cells) is governed by a point-process, namely a Poisson process. Rodriguez-Iturbe et al. (1987) discussed the characteristics of such a model in which each point generated is associated with a rectangular pulse of random duration and constant precipitation intensity (i.e. each precipitation object generates a constant amount of rain for a fixed duration). The total rainfall field at a given time is obtained by summing up the contribution of all active points at that time. Similarly, Cox and Isham (1988) discussed a 2-dimensional model in which storms are generated at locations following a temporal and spatial Poisson process. Each storm is represented by a circular area of fixed radius moving with constant random velocity. Each storm lives for a given duration during which it precipitates with constant intensity over the area it covers. Storms contributions are summed up at each location. Statistics can be computed analytically using these models and fitted to existing records of precipitation data in order to estimate the parameters of the different distributions (method of moments). A model similar to that of Cox and Isham (1988) can be found in Smith and Krajewski (1987).

The main limitation of such models is their inability to represent the different time scales involved in real precipitation processes. To overcome this issue, Rodriguez-Iturbe et al. (1987) replaced the single point process with a clustered point process in which several point processes are used to generate clusters of precipitation objects. A process is responsible for setting the storm origins, another process decides the number of cells for each cluster, and a third gives the origins of each cell within the storm. Two candidates for the cell generating process are suggested: the Neyman-Scott process and the Bartlett-Lewis process.

In the Neyman-Scott process, storm origins are generated from a Poisson process. Each storm is given a random number of precipitation cells. Cells are displaced from the storm's origin (in time) by a delay drawn from an exponential distribution. In the Bartlett-Lewis process, storm origins are also specified using a Poisson process, but a second Poisson process dictates, for each storm, the time at which cells are initiated. This cell-generating process is terminated after a sample duration drawn from an exponential distribution.

Point-process precipitation models have given rise to a large amount of research. For instance, Cox and Isham (1988) added some dynamics to their model by having storms propagated with constant random velocity shared by all their constituting cells. Cowpertwait et al. (1996) applied a Neyman-Scott process to single-site precipitation forecasting, Northrop (1997) took the model of Cox and Isham (1988) further by adding support for cells that are elliptic rather than circular. Two alternative methods are suggested to initialise cells locations. The first consists in moving cells away from the storm's centre by a displacement drawn from a bivariate zero-mean Gaussian distribution (similarly to the Neyman-Scott process approach). The second assigns random (uniform) locations to cells within an ellipse around the storm's centre. Onof and Wheater (1993); Koutsoyiannis and Onof (2001); Smithers et al. (2002) are few of the many applications of the Bartlett-Lewis process to rainfall modelling. A comparative review of some of these models can be found in Wheater et al. (2000).

In a recent report, Cowpertwait et al. (2007) introduced more flexibility in a point-process model by discarding the assumption that rain cells precipitate have constant precipitation intensity over their lifetime. In their model, during the life cycle of the cell, precipitation pulses occur according to another Poisson point-process.

Point-process models focus largely on the nature of the rainfall field and the characteristics of storms and thus are mostly used as descriptive models rather than prediction models.

Other approaches

Xu et al. (2005) apply the probabilistic framework of integro-differential equations developed by Wikle (2002) to precipitation nowcasting. In this framework, a redistribution kernel specifies how each pixel's intensity evolves in time as a linear combination of the previous pixels' intensities. The study is restricted to elliptic kernels. This method is able to model complicated features of rain such as translation (motion) and diffusion (growth/decay). Parameter estimation is performed using Markov Chain Monte Carlo, after projection onto the spectral domain (using Fourier decomposition), and dimension reduction (often, most of the high-frequency Fourier coefficients are negligible and can be discarded).

Neural networks are being applied increasingly to precipitation forecasting. Grecu and Krajewski (2000) modelled the evolution of the rain field using a back-propagation neural network trained on real data, but showed that this method had little advantage over the more traditional advection schemes. Liu et al. (2001) obtained good results using a Radial Basis Function network. Kuligowski and Barros (1998) use a neural network with a single hidden layer for nowcasting of point (i.e. unidimensional) precipitation. There are too many examples of such applications in the literature to give more than an overview here, but a descriptive review of 43 such examples can be

found in Maier and Dandy (2000).

6.2 A stochastic rainfall prediction model

The motivation for stochastic models in the context of precipitation forecasting arises from the fact that neither the data (radar) nor the models provide an exact representation of the true process. They are both subject to errors or approximations of various types.

Radar observations are effectively reflectivity measurements, i.e. they measure intensities of a reflected electro-magnetic beam of specified wavelength. Unfortunately, the beam is not reflected by precipitation only. Buildings, flying objects, as well as anomalous propagations, can corrupt the reflectivity field. Dense showers can shield further precipitation thus leaving them undetected. Rain can either form or disappear below the beam, leading to further incorrect measurements. Numerical approximation in the conversion from reflectivity fields to rainfall rates degrades the data further.

Nowcasting models often provide a crude representation of the rainfall field and its evolution process. As a consequence, the forecast itself can only be imperfect. It is thus important to be able not only to predict the evolution of precipitation events, but also to quantify how uncertain about this evolution we are. Smith and Austin (2000) indicate that "forecast products, particularly those for hydrological applications, need to be statistical in nature, giving (for example) a range and a likelihood for falling within that range rather than just a best estimate of the value." Further details on the motivation for probabilistic forecasts in hydrology can also be found in (Krzysztofowicz, 2001).

In the following sections, we introduce a radar-based rainfall model which provides a fully probabilistic framework for precipitation nowcasting and extends the work of Cornford (2004). Section 6.2 describes the model and discusses its application to data assimilation. Chapter 7 discusses results on synthetic and real data.

6.2.1 Nature of the data

The observed rainfall comes as a sequence of radar images, available in real-time. Each image is defined over a spatial domain Ω, in this case a grid with resolution 5×5 km, and total dimensions of about 100 km. Figure 6.2 shows one of these radar images. We denote M the number of pixels in Ω and $\mathbf{s}$ the input vector made of all pixel locations, so that $\mathbf{s} = (\mathbf{s}_1, ..., \mathbf{s}_j, ..., \mathbf{s}_M)$. We also denote by s_j^h and s_j^v respectively the horizontal and vertical components of the coordinate $\mathbf{s}_j$, so that $\mathbf{s}_j = (s_j^h, s_j^v)$. For a given observation $\mathbf{y}$, each pixel $\mathbf{s}_j$ has the associated observed rainfall intensity y_j, so that $\mathbf{y} = (y_1, ..., y_j, ..., y_M)$.

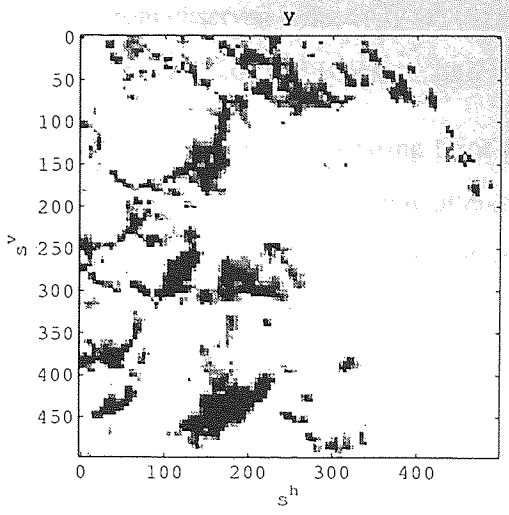


Figure 6.2: A sample radar image

Observation error

It has been stated that most measurements are imperfect, which motivates the use of a probabilistic assimilation framework. Radar data suffer from many sources of errors resulting, potentially, in an incorrect rainfall field estimate. The most common error sources usually associated with radar data are (Alberoni et al., 2003; Golz et al., 2005):

- Blocking and clutter: the radar beam can be obstructed by hills or mountains (blocking) or detect ground objects (buildings, trees...) or flying objects (insect clouds, migrating birds) which will be identified as precipitation
- Attenuation: the beam's intensity is weakened beyond stormy areas as these absorb much of the beam's energy.
- Overshooting: Due to the angle between the beam and the ground, measurements at a large distance from the radar sometimes capture precipitation aloft which does not reach the ground (evaporation) and sometimes fail to capture precipitation from low clouds (overshooting).
- Anomalous propagation: in some particular situations, the radar beam can be refracted towards the ground by a layer of air, resulting in precipitation being (wrongly) detected. This is most common at early hours, when a layer of warm air lies above a layer of cooler air.
- Bright band: snow melting into rain in the atmosphere has higher reflectivity than rain and can be misinterpreted as heavy rain.

- Incorrect conversion model from observed reflectivity ($\mathbf{z}$) to rain intensity ($\mathbf{y}$). Nimrod uses the Marshall-Palmer relationship $\mathbf{z} = 200\ \mathbf{y}^{1.6}$ (Gibson, 2001).

The NIMROD radar data undergoes intensive processing prior to being released, in order to correct for noise, clutter, occultation, anomalous propagation, attenuation, range, bright band and orographic enhancement (UK Meteorological Office, 2003). It is worth pointing out that, although the resulting products are much more accurate that the raw data, they are still prone to error (for instance, errors due to overshooting can hardly be estimated without the use of ground-based rain gages, which are not available at every location).

Preprocessing

To accommodate the smooth nature of the spatial model used (discussed in Section 6.2.2), a Gaussian filter (Gonzalez and Woods, 2008) is applied to the data. Each pixel in the radar image is replaced with a weighted average of the surrounding pixels (including itself). For a given pixel $\mathbf{s} = (s^h, s^v)$, the neighbouring pixels within a radius r are defined as $\Theta(\mathbf{s}, r) = \{\mathbf{s}_k = (s_k^h, s_k^v) \mid \max(|s_k^h - s^h|, |s_k^v - s^v|) \leqslant r\}$, e.g. with a radius of 1, the 1-nearest neighbours are:

$$\Theta(s,1) = \begin{array}{|c|c|c|} \hline s_1 & s_2 & s_3 \\ \hline s_4 & s & s_6 \\ \hline s_7 & s_8 & s_9 \\ \hline \end{array} \tag{6.1}$$

The rain intensity at pixel sis then given by the weighted sum of the surrounding pixels:

$$y(\mathbf{s}) = \sum_{\mathbf{s}_k \in \Theta(\mathbf{s},r)} w_k\, y(\mathbf{s}_k), \tag{6.2}$$

where the weights w_k are provided by a normalised discrete Gaussian kernel centred on $\mathbf{s}$. A radius of 2 pixels (i.e. 10 km) and a variance $\sigma^2 = 2.0$ are chosen for the kernel, giving the following weights:

$$\begin{pmatrix} 0.012 & 0.026 & 0.033 & 0.026 & 0.012 \\ 0.026 & 0.055 & 0.071 & 0.055 & 0.026 \\ 0.033 & 0.071 & 0.092 & 0.071 & 0.033 \\ 0.026 & 0.055 & 0.071 & 0.055 & 0.026 \\ 0.012 & 0.026 & 0.033 & 0.026 & 0.012 \end{pmatrix} \tag{6.3}$$

Figure 6.3 shows the original radar image and the resulting smooth precipitation field (right).

6.2.2 Spatial representation

This section describes the operator h used to map the 2-dimensional spatial domain Ω to the precipitation intensity space. In our case, Ω is a grid with resolution 5×5 km, and total dimensions

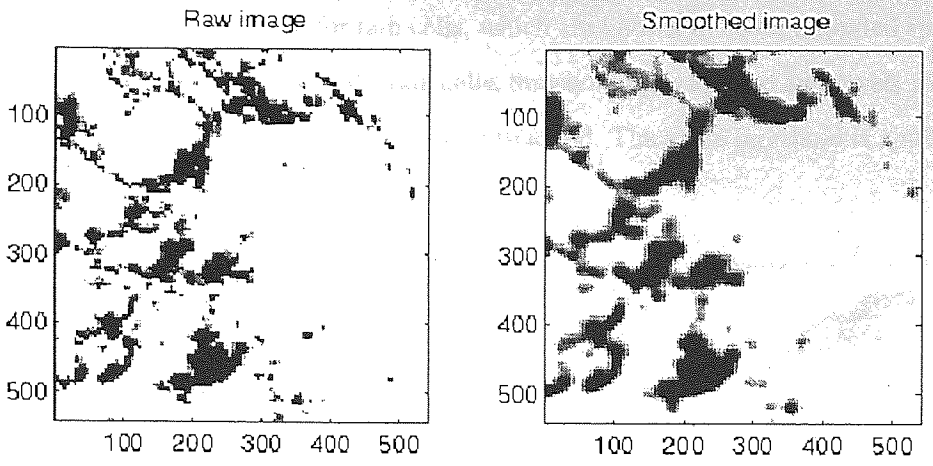


Figure 6.3: A sample radar image (left) and the resulting smoothed image (right). In this example, the domain is 480×480 km at 5×5 km resolution.

500×500 km^2 (100×100 square pixels).

The rainfall field is modelled by a set of K 2-dimensional unnormalised Gaussian basis functions with state vector $\mathbf{x}_k = (\mathbf{c}_k, w_k, r_k)$, where $\mathbf{c}_k$ is the k^{th} cell's centre, w_k its width and r_k the rainfall intensity at the centre.

Thus, the modelled rainfall intensity at pixel $\mathbf{s}_j$ for cell k is:

$$h(\mathbf{x}_k, \mathbf{s}_j) = r_k \exp\left(\frac{\| \mathbf{c}_k - \mathbf{s}_j \|^2}{2w_k}\right) \tag{6.4}$$

The total rainfall intensity at a given pixel is the sum of the intensities from all cells:

$$h(\mathbf{x}, \mathbf{s}_j) = \sum_{k=1}^{K} h(\mathbf{s}_j, \mathbf{x}_k), \tag{6.5}$$

where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_K)$ is the parameter vector for all cells.

We define similarly the rainfall field over the space domain Ω for a given cell:

$$h(\mathbf{x}_k, \mathbf{s}) = (h(\mathbf{x}_k, \mathbf{s}_1), \ldots, h(\mathbf{x}_k, \mathbf{s}_M)), \tag{6.6}$$

and the total rainfall field over the image is the vector:

$$h(\mathbf{x}, \mathbf{s}) = \sum_{k=1}^{K} h(\mathbf{x}_k, \mathbf{s}). \tag{6.7}$$

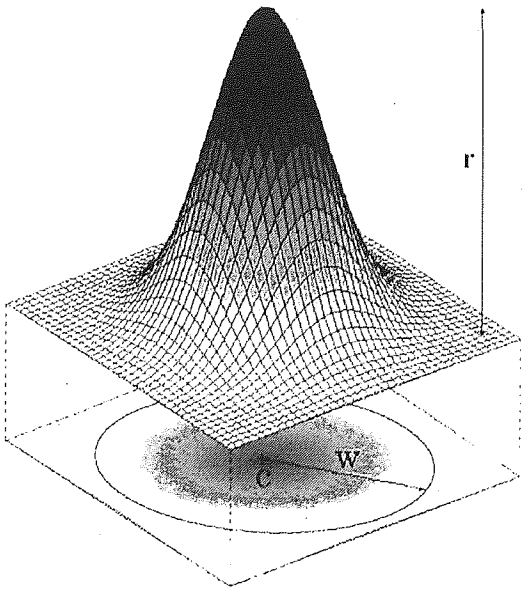


Figure 6.4: A Gaussian-shaped rain cell

The basis functions considered are continuous in space with infinite support, and thus differentiable. Although this makes gradient computations (and thus gradient-based optimisation)

possible, it is an unrealistic feature for rain cells, which are known to cover a limited spatial area only. In order to localise the intensity of a rain cells, thresholding is applied and pixels with a total rain rate h below a certain limit ($0.5\,\mathrm{mm.h}^{-1}$) are discarded. The same threshold is applied to the the observations for consistency.

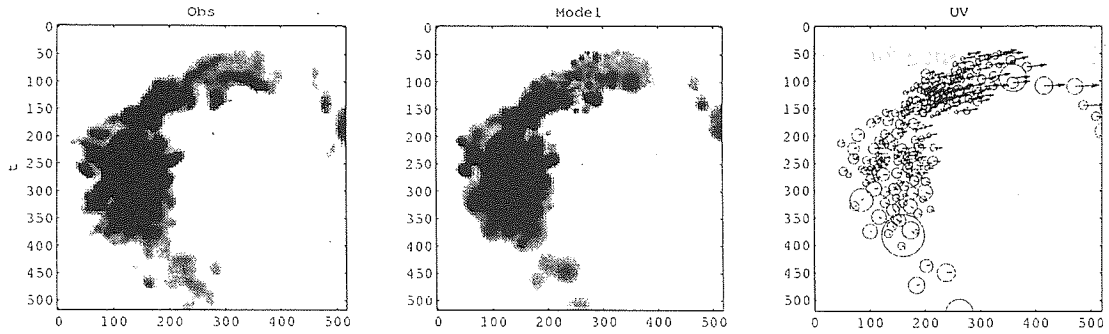


Figure 6.5: A radar observation of the rainfall field (left) and the model's corresponding representation (centre), with $K = 200$ cells. The principal cells' contours and advection vectors are plotted on the right.

The model described above provides a visually satisfying static approximation to the observed rain field (see Figure 6.5).

6.2.3 Dynamics

The rain field is evolved in time and space according to the advection equation:

$$\frac{\mathrm{d}h}{\mathrm{dt}} + \mathbf{u}.\nabla h \approx 0, \tag{6.8}$$

where $\mathbf{u}$ denotes the *advection* (i.e. velocity) of the rain field. The equation relates the evolution of the rainfall in time and space as follows: assuming the rain field remains fixed while being advected, we can write its conservation equation.

The advection is specified at each cell's location (see Figure 6.5, right-hand plot), so that $\mathbf{u} = (\mathbf{u}_1, ..., \mathbf{u}_K)$ with spatial vector $\mathbf{u}_k$ denoting the advection at cell k. We will also use the notation $\mathbf{u} = (u, v)$ when it is necessary to distinguish the (multivariate) spatial components u (horizontal) and v (vertical). u_k and v_k then denote the similar scalar components for the k^{th}cell .

6.2.4 Overview of the data assimilation process

The adaptation of the Bayesian formalism discussed in Section 2.3.2 to the Basis Functions model is introduced in this section and discussed in the following.

The data assimilation scheme involves the following sequence of steps:

1. Initialisation of the model

   Using the first two observations, the cell's centres, widths and intensities are estimated. From the two resulting sets of model parameters, the advection is initialised.

2. Propagation (forecast)

   The cell's centres are propagated using the current estimate of the advection. The widths, intensities and advection components are assumed constant. Model error is added to all the distributions to account for the model uncertainties.

3. Assimilation of new observation

   (a) The rain cells that have collapsed/disappeared are removed from the model

   (b) The posterior is computed, given the predicted distribution (step 2) and the observations

   (c) New cells are detected and added to the model

   (d) The advection is updated given the cells new locations

Once the model has been initialised (step 1), the propagation of the model (step 2) and assimilation of new observations (step 3) are repeated for each time increment. Details about these steps are presented in Sections 6.3 and 6.4.

6.2.5 Priors and likelihood

In order to be able to perform Bayesian assimilation in this model, suitable priors on the parameters and observations (likelihood) are chosen. This section describes the choice of priors for the model.

Likelihood of the data

The observation's likelihood is chosen to have a Gaussian distribution:

$$\mathrm{p}(\mathbf{y}|\mathbf{x}) = \frac{1}{(2\pi)^{\frac{M}{2}}\,|\mathbf{R}|^{\frac{1}{2}}}\, e^{-\frac{1}{2}(\mathbf{y}-h(\mathbf{x}))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-h(\mathbf{x}))} \tag{6.9}$$

where $\mathbf{y}$ is the observed rainfall field, $h(\mathbf{x})$ is the predicted rainfall field given the state $\mathbf{x}$, and $\mathbf{R}$ is the covariance matrix for the observation error (as specified in Section 2.2.3).

Prior over the rainfall field's parameters

There exist some families of prior distributions which are *conjugate* for a particular likelihood, i.e. the posterior distribution belongs to the same family of distributions as the prior. Very often in the

Bayesian paradigm, the normalising constant $p(\mathbf{Y}_t)$ cannot be evaluated and one has to resort to approximations. Conjugate priors then ensure the nature of the posterior is known and one only needs to estimate its parameters.

The priors over the parameters have been chosen as follows. The widths are defined to have an Inverse-Gamma distribution:

$$p(w_k) = \mathrm{IGa}(\alpha_k, \beta_k), \tag{6.10}$$

the centres follow a Bivariate Normal distribution conditioned on the widths, up to a scaling factor ξ_k (so that centres and widths jointly follow a Normal-Gamma distribution):

$$p(\mathbf{c}_k|w_k) = \mathcal{N}(\bar{\mathbf{c}}_k, \xi_k w_k), \tag{6.11}$$

and the intensities are Gamma distributed:

$$p(r_k) = \mathrm{Ga}(\gamma_k, \delta_k). \tag{6.12}$$

This choice of prior is motivated by the following considerations:

- The priors over the intensities and widths should ideally have strictly positive support,
- The larger the cell, the less confident one is about its exact centre's position, hence the conditioning on the width,
- For the Gaussian likelihood, the Normal-Gamma prior on the centres and inverse-widths (which is equivalent to a Normal-Inverse Gamma prior on the centres and widths) is conjugate (Bernardo and Smith, 1994).

The distributions are assumed uncorrelated between cells and groups of parameters, so that the state's distribution factorises as:

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(\mathbf{c}_k|w_k)\, p(w_k)\, p(r_k) \tag{6.13}$$

Advection prior

The advection $\mathbf{u}$ follows a bivariate Gaussian process with polynomial exponential correlation function. The Gaussian process has been designed to include both a divergent and a rotational component, in order to display the type of behaviour one would expect from an advection field. Realisations from this Gaussian process are smooth and rotational in character. The advection is estimated at each cell's centre (see Figure 6.5, right-hand plot).

The following paragraphs reproduce the work carried out by Cornford et al. in the two technical reports (Cornford, 1998b,a) and the paper (Cornford et al., 2002). This work is largely based

on Daley (1991) and leads to the particular choice of covariance function used for the advection prior.

Daley (1991) showed that, using the Helmoltz theorem, the vector flow field for the advection can be related to a rotational potential Ψ and a divergent potential Φ. These relate to the vector components as follows:

$$u = -\frac{\partial \Psi}{\partial s^v} + \frac{\partial \Phi}{\partial s^h}$$
$$v = \frac{\partial \Psi}{\partial s^h} + \frac{\partial \Phi}{\partial s^v} \tag{6.14}$$

where u and v denote the usual horizontal and vertical component of the vector field and (s^h, s^v) are the spatial coordinates over the domain. The following demonstration shows how the specification of an appropriate correlation function for Ψ and Φ leads to a convenient covariance function for $\mathbf{u}$ which only depends on the distance between cells. This covariance function will be denoted as $K(\mathbf{c}, \mathbf{c}')$, so that $\Sigma_{\mathbf{u}} = K(\mathbf{c}, \mathbf{c})$.

One can convert from (u, v) to (Ψ, Φ) and back as detailed below. Given two advection vectors $\mathbf{u}_1 = (u_1, v_1)$ and $\mathbf{u}_2 = (u_2, v_2)$ at locations $\mathbf{c}_1$ and $\mathbf{c}_2$, we define $r = \mathbf{c}_2 - \mathbf{c}_1$. The longitudinal component (along the direction r) and the transverse component (across r) of the advection can then be expressed as:

$$l = u\cos(\phi) + v\sin(\phi)$$
$$t = -u\sin(\phi) + v\cos(\phi) \tag{6.15}$$

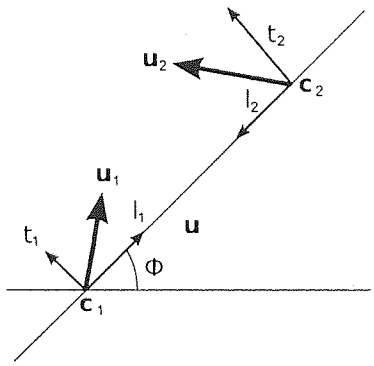


Figure 6.6: Conversion from $\mathbf{u} = (u, v)$ to (l, t) coordinates

where ϕ is the angle between the x-axis and the vector r (Figure 6.6, after Cornford (1998a)). Following from (6.15), the covariances can be expressed:

$$\begin{pmatrix} C_{ll} & C_{lt} \\ C_{tl} & C_{tt} \end{pmatrix} = \begin{pmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix} \begin{pmatrix} C_{uu} & C_{uv} \\ C_{vu} & C_{vv} \end{pmatrix} \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix}. \tag{6.16}$$

These can be related to the covariances for the stream function Ψ and velocity potential Φ in radial coordinates:

$$C_{ll}(r) = -\frac{1}{r}\frac{\partial}{\partial r} C_{\Psi\Psi}(r) - \frac{\partial^2}{\partial r^2} C_{\Phi\Phi}(r) \tag{6.17}$$
$$C_{tt}(r) = -\frac{\partial^2}{\partial r^2} C_{\Psi\Psi}(r) - \frac{1}{r}\frac{\partial}{\partial r} C_{\Phi\Phi}(r) \tag{6.18}$$
$$C_{lt}(r) = C_{tl}(r) = 0 \tag{6.19}$$

Thus, given some suitably defined covariances $C_{\Psi\Psi}$ and $C_{\Phi\Phi}$, it is possible to compute the covariances C_{uu}, C_{uv}, C_{vu} and C_{vv} in a way which guarantees the joint covariance of u and v is positive

on Daley (1991) and leads to the particular choice of covariance function used for the advection prior.

Daley (1991) showed that, using the Helmoltz theorem, the vector flow field for the advection can be related to a rotational potential Ψ and a divergent potential Φ. These relate to the vector components as follows:

$$\begin{aligned} u &= -\frac{\partial \Psi}{\partial s^v} + \frac{\partial \Phi}{\partial s^h} \\ v &= \frac{\partial \Psi}{\partial s^h} + \frac{\partial \Phi}{\partial s^v} \end{aligned} \tag{6.14}$$

where u and v denote the usual horizontal and vertical component of the vector field and (s^h, s^v) are the spatial coordinates over the domain. The following demonstration shows how the specification of an appropriate correlation function for Ψ and Φ leads to a convenient covariance function for $\mathbf{u}$ which only depends on the distance between cells. This covariance function will be denoted as $\mathrm{K}(\mathbf{c}, \mathbf{c}')$, so that $\Sigma_{\mathbf{u}} = \mathrm{K}(\mathbf{c}, \mathbf{c})$.

One can convert from (u, v) to (Ψ, Φ) and back as detailed below. Given two advection vectors $\mathbf{u}_1 = (u_1, v_1)$ and $\mathbf{u}_2 = (u_2, v_2)$ at locations $\mathbf{c}_1$ and $\mathbf{c}_2$, we define $r = \mathbf{c}_2 - \mathbf{c}_1$. The longitudinal component (along the direction r) and the transverse component (across r) of the advection can then be expressed as:

$$\begin{aligned} l &= u\cos(\phi) + v\sin(\phi) \\ t &= -u\sin(\phi) + v\cos(\phi) \end{aligned} \tag{6.15}$$

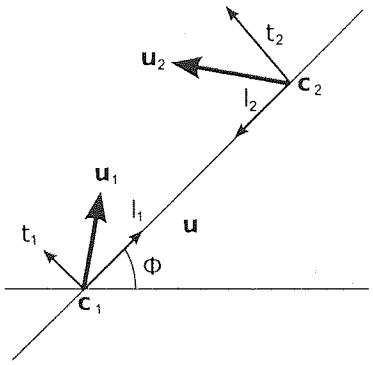


Figure 6.6: Conversion from $\mathbf{u} = (u, v)$ to (l, t) coordinates

where ϕ is the angle between the x-axis and the vector r (Figure 6.6, after Cornford (1998a)). Following from (6.15), the covariances can be expressed:

$$\begin{pmatrix} C_{ll} & C_{lt} \\ C_{tl} & C_{tt} \end{pmatrix} = \begin{pmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix} \begin{pmatrix} C_{uu} & C_{uv} \\ C_{vu} & C_{vv} \end{pmatrix} \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix}. \tag{6.16}$$

These can be related to the covariances for the stream function Ψ and velocity potential Φ in radial coordinates:

$$C_{ll}(r) = -\frac{1}{r}\frac{\partial}{\partial r} C_{\Psi\Psi}(r) - \frac{\partial^2}{\partial r^2} C_{\Phi\Phi}(r) \tag{6.17}$$

$$C_{tt}(r) = -\frac{\partial^2}{\partial r^2} C_{\Psi\Psi}(r) - \frac{1}{r}\frac{\partial}{\partial r} C_{\Phi\Phi}(r) \tag{6.18}$$

$$C_{lt}(r) = C_{tl}(r) = 0 \tag{6.19}$$

Thus, given some suitably defined covariances $C_{\Psi\Psi}$ and $C_{\Phi\Phi}$, it is possible to compute the covariances C_{uu}, C_{uv}, C_{vu} and C_{vv} in a way which guarantees the joint covariance of u and v is positive

definite:

$$\begin{aligned}\Sigma_{\mathbf{u}} &= \mathrm{K}(\mathbf{c},\mathbf{c}) \\ &= \begin{pmatrix} C_{uu} & C_{uv} \\ C_{vu} & C_{vv} \end{pmatrix} \\ &= \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \begin{pmatrix} C_{ll} & C_{lt} \\ C_{tl} & C_{tt} \end{pmatrix} \begin{pmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix}\end{aligned} \tag{6.20}$$

The covariance functions $C_{\Psi\Psi}$ and $C_{\Phi\Phi}$ are usually expressed using correlation functions ρ, the conversion from one to the other being given by the identity: $C_{\Psi\Psi} = E_{\Psi}^2 \rho_{\Psi}$, where E_{Ψ}^2 is the variance of the rotational component Ψ (i.e. its covariance at $r = 0$). Equation (6.19) can then be rewritten:

$$\begin{aligned} C_{ll}(r) &= -E_{\Psi}^2 L_{*\Psi}^2 \frac{1}{r}\frac{\partial}{\partial r}\rho_{\Psi\Psi} - E_{\Phi}^2 L_{*\Phi}^2 \frac{\partial^2}{\partial r^2}\rho_{\Phi\Phi} \\ C_{tt}(r) &= -E_{\Phi}^2 L_{*\Phi}^2 \frac{\partial^2}{\partial r^2}\rho_{\Psi\Psi} - E_{\Psi}^2 L_{*\Psi}^2 \frac{1}{r}\frac{\partial}{\partial r}\rho_{\Phi\Phi} \end{aligned} \tag{6.21}$$

This formulation provides a flexible way to balance the influences of the divergent and rotational fields in $\mathbf{u}$, by choosing the ratio between E_{Φ}^2 and E_{Ψ}^2.

The L_* terms result from the change of coordinate system and are called the *effective length scales* of the parameters (Ψ and Φ). They are defined as:

$$L_* = \left.\frac{2\rho}{\nabla^2\rho}\right|_{r\to 0} \tag{6.22}$$

with

$$\nabla^2 = \frac{1}{r}\frac{\partial}{\partial r} r \frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \phi^2}. \tag{6.23}$$

The same correlation function is chosen in this work for both the rotational and divergent component, albeit with different parameters. It is derived from a Matérn covariance function (Rasmussen and Williams, 2006) with a smoothness parameter $\nu = 2.5$, which then simplifies to a polynomial-exponential covariance function of the form:

$$\rho(r) = \left(1 + \frac{r}{L} + \frac{r^2}{3L^2}\right)\exp\left(-\frac{r}{L}\right) \tag{6.24}$$

where L denotes the length-scale of the process. The value of ν has been determined using several years of wind field data as detailed in Cornford et al. (2002).

It is straightforward to show that ρ accepts the following derivatives:

$$\begin{aligned} \frac{\partial}{\partial r}\rho(r) &= -\frac{1}{3L^2}\, r \left(1 + \frac{r}{L}\right)\exp\left(-\frac{r}{L}\right) \\ \frac{\partial^2}{\partial r^2}\rho(r) &= -\frac{1}{3L^2}\left(1 + \frac{r}{L} - \frac{r^2}{L^2}\right)\exp\left(-\frac{r}{L}\right) \end{aligned} \tag{6.25}$$

It is easily proved using (6.25) and (6.23) that, for this choice of $\rho(r)$, the effective length scales evaluates for $r \to 0$ to:

$$L_* = -\frac{1}{3L^2}. \tag{6.26}$$

(6.25) and (6.26) can then be plugged back into (6.21) to compute C_{ll} and C_{tt}. Substituting them in turn into (6.20) gives the joint covariance for u and v.

6.3 Initialisation of the model

The first step in the data assimilation scheme consists in providing an initial estimate of the parameters for the priors discussed in section 6.2.5. The initialisation of the model is two-fold: first, two rainfall field estimates are computed using the first and second observation fields available, then the advection is initialised from these two consecutive rainfall fields.

6.3.1 Initialisation of the rainfall field

The rainfall field is initialised by sequentially adding new cells until a goodness of fit criterion is achieved. This criterion involves minimising the relative error between the observed field and the estimated field:

$$E(\mathbf{x},\mathbf{y}) = \frac{\| h(\mathbf{x}) - \mathbf{y} \|}{\| \mathbf{y} \|} \tag{6.27}$$

The overall algorithm is described in Figure 6.7. The algorithm starts with an empty rainfall field estimate, meaning the misfit $\mathbf{y}'$ is equal to the observed rainfall field $\mathbf{y}$. The pixel $\mathbf{s}$ with highest intensity $\mathbf{y}'(\mathbf{s})$ is located. $\mathbf{s}$ and $\mathbf{y}'(\mathbf{s})$ will be used as initial estimates for the new cell's centre and intensity respectively.

By analysing the gradient of the intensity along the vertical and horizontal directions, an estimate of the cell's radius d is computed (starting from $d = 1$, d is increased by 1 pixel until the gradient becomes positive along one of the directions). A new cell with parameters $\mathbf{x}_k = (\mathbf{c}_k, w_k, r_k) = (\mathbf{s}, \mathbf{y}'(\mathbf{s}), d)$ is added to the estimate of the rainfall field.

The updated rainfall field is then optimised by minimising its negative log-likelihood with respect to the observation. Given a Gaussian likelihood for the observation with fixed covariance matrix $\mathbf{R}$, the negative log-likelihood is given by:

$$-\ln(\mathrm{p}(\mathbf{x}|\mathbf{y})) \propto \frac{1}{2}(\mathbf{y} - h(\mathbf{x}))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y} - h(\mathbf{x})) \tag{6.28}$$

The new misfit is computed by subtracting the estimated rainfall field from the observed rainfall field, giving $\mathbf{y}' = \mathbf{y} - h(\mathbf{x}_k)$. The procedure is repeated until the error reaches a predefined minimum value, unless the maximum number of cells is reached before.

- Compute initial misfit: $\mathbf{y}' = \mathbf{y}$
- While $E(\mathbf{x}, \mathbf{y}) < E_{min}$
  - Find location $\mathbf{s}$ with maximal intensity $\mathbf{y}'(\mathbf{s})$
  - Set cell's centre: $\mathbf{c} = \mathbf{s}$
  - Set cell's height: $r = \mathbf{y}'(\mathbf{s})$
  - Find maximal radius d for which the slope along the vertical and horizontal directions remains negative
  - Set cell's width: $w = 2 \times d \times \Delta\mathbf{s}$, where $\Delta\mathbf{s}$ is the spatial resolution in km
  - Add cell to rainfall field (increment state): $\mathbf{x} = (\mathbf{x}, \mathbf{c}, w, r)$
  - Optimise fit to data by minimising 6.28
  - Compute new misfit: $\mathbf{y}' = \mathbf{y} - h(\mathbf{x})$

Figure 6.7: Initialisation of the rainfall model

Covariances The covariances for the cells parameters are set to some user-defined default value. In the experiment setting, all covariances are assumed diagonal initially.

6.3.2 Initialisation of the advection field

Two rainfall field objects are initialised using the first and second observations by applying the steps described in section 6.3.1. It is not possible at this stage to match the cells from the first rainfall field with these from the second rainfall field, as these have been initialised independently. Thus, the initial advection cannot be computed at each cell's location and one has to resort to finding a "best" overall initial estimate.

Going back to the advection equation (6.8), the $\frac{dh}{dt}$ term can be obtained from the two initial rainfall fields:

$$\frac{dh}{dt} \approx \frac{h(t_1) - h(t_0)}{t_1 - t_0} \tag{6.29}$$

while the gradient term ∇h is derived from the first rainfall field (equation 6.7), so that for each pixel j:

$$(\nabla h)_j = \sum_k \begin{pmatrix} \frac{c_k^h - s_j^h}{w_k} \\ \frac{c_k^v - s_j^v}{w_k} \end{pmatrix} h(\mathbf{x}_k, \mathbf{s}_j) \tag{6.30}$$

A pseudo-inversion is then applied to equation (6.8) to find the best estimate $\mathbf{u}_0$ (in a least-square sense), which is then used to set the initial advection of all the cells in the first rainfall field:

$$\mathbf{u}_0 = -\frac{\mathrm{d}h}{\mathrm{dt}}.\nabla h^+ \tag{6.31}$$

where ∇h^+ is the Moore-Penrose pseudo-inverse of ∇h.

Covariance The covariance matrix for the Gaussian process is computed by applying the covariance function described in 6.2.5 over the set of cells locations. The rotational component is characterised by a variance $E_\Psi^2 = 40\,\mathrm{m.s}^{-1}$ and a length-scale $L_\Psi = 200\,\mathrm{km}$. The divergent component uses $E_\Phi^2 = 0.5\,\mathrm{m.s}^{-1}$ and $L_\Phi = 200\,\mathrm{km}$. These values have been chosen according to expert judgement following meteorologists advice.

6.4 Data assimilation in the BF model

Once initialised, the rainfall field model can be run in an assimilation setting by simply alternating between the prediction (propagation to the next time step) and the assimilation (update given a new observation) steps. Changes in the cells population due to birth of new cells and death of existing ones are handled with simple detection/deletion schemes. These steps are described in the following section.

6.4.1 Propagation of the rainfall field

The rainfall field is propagated forward in time in a linear fashion, following the advection equation (6.8). This translates effectively to each cell being advected by the corresponding estimate of the advection.

The cell centres are evolved according to their associated advection:

$$\bar{\mathbf{c}}^f(\mathrm{t}+1) = \bar{\mathbf{c}}^a(\mathrm{t}) + \Delta\mathrm{t}\,\bar{\mathbf{u}}, \tag{6.32}$$

$$\Sigma_{\mathbf{c}}^f(\mathrm{t}+1) = \Sigma_{\mathbf{c}}^a(\mathrm{t}) + (\Delta\mathrm{t})^2\,\Sigma_{\mathbf{u}}(\mathrm{t}) + \Delta\mathrm{t}\,\mathbf{Q}_{\mathbf{c}}(\mathrm{t}), \tag{6.33}$$

where the f and a indexes have been used for the predicted and assimilated values respectively. Note that in the experimental settings used here, the model error covariance matrix $\mathbf{Q}_{\mathbf{c}}$ is assumed diagonal and time-invariant.

The widths and heights are assumed constant during prediction, which is a reasonable assumption at the time scales this model is targetting (15 min ahead prediction). This gives the following

predicted estimates for the widths:

$$\bar{w}^f(t+1) = \bar{w}^a(t), \tag{6.34}$$

$$\Sigma_w^f(t+1) = \Sigma_w^a(t) + \Delta t\, \mathbf{Q}_w(t), \tag{6.35}$$

and similarly, for the cells' intensities:

$$\bar{r}^f(t+1) = \bar{r}^a(t), \tag{6.36}$$

$$\Sigma_r^f(t+1) = \Sigma_r^a(t) + \Delta t\, \mathbf{Q}_r(t). \tag{6.37}$$

Model error is added to the predicted parameters to reflect the increased uncertainty due to the model's imperfections. Model error is assumed Gaussian with zero-mean for all parameters (i.e. there is no bias).

In the model, the cells are assumed to have no dynamics other than motion. This is an unrealistic feature, as we know that rain cells also undergo modifications due to internal dynamics, resulting in growth and decay phenomena. However, Wilson et al. (1998) conducted an experiment in which they compared the performace of the TITAN nowcasting system with and without capturing (and forecasting) the evolution of cells size and intensity. They found no significant difference between the performance of the two methods. This confirms a similar study by Tsonis and Austin (1981) and the assumption that at short scales, motion is the main observable cause of change for rainfall fields.

The reader will notice that the model's noise is assumed Gaussian while the widths and intensities have Gamma distributions. It is thus necessary to convert the effect of the propagation from (mean, variance) to (α, β) space. Because for Gamma distributions the mode (most probable value) is often more meaningful than the mean (for a skewed Gamma, the mode can be fairly distant from the mean), the mode is used to set the new parameters, rather than the mean.

For a Gamma distribution with parameters (α, β), the mode m and variance σ^2 are related to the parameters according to:

$$m = \frac{\alpha - 1}{\beta} \tag{6.38}$$

$$\sigma^2 = \frac{\alpha}{\beta^2} \tag{6.39}$$

Expressing α and β as functions of m and σ^2 merely requires solving a quadratic equation in one of the parameters. Solving for instance the equation in β, and noting that only one of the roots is positive, the values for α and β are given by:

$$\beta = \frac{m + \sqrt{m^2 + 4\sigma^2}}{\sigma^2}, \tag{6.40}$$

$$\alpha = 1 + \beta m. \tag{6.41}$$

A similar conversion can be done with an Inverse-Gamma distribution, however, the relationships between the parameters are more complicated:

$$m = \frac{\beta}{\alpha + 1}, \tag{6.42}$$

$$\sigma^2 = \frac{\beta^2}{(\alpha-1)^2(\alpha-2)}, \tag{6.43}$$

and require solving a cubic equation in the parameters:

$$\alpha^3 - (4+s)\alpha^2 + (5-2s)\alpha - (2+s) = 0 \tag{6.44}$$

where $s = \frac{m^2}{\sigma^2}$.

It has been observed numerically that there is a single value of s for which equation (6.44) accepts two real roots (Figure 6.8). This value, which lies in the range $[0.5, 1.5]$, has however never been met in experimental settings, for which equation (6.44) has always shown a single real root. This root is computed numerically in practice.

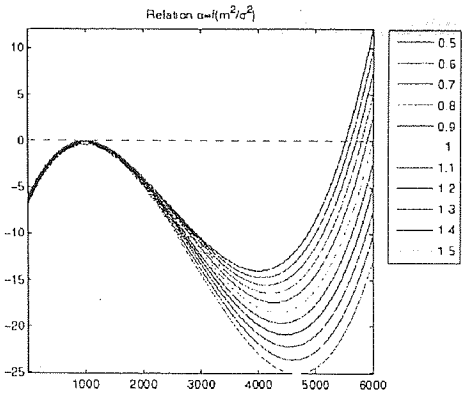


Figure 6.8: Plot of polynomial in (6.44) for different values of σ^2

6.4.2 Propagation of the advection field

The advection field is assumed constant at the propagation time-scales considered. To account for model error, a fraction $\sigma_{\mathbf{u}}^2$ of the initial covariance matrix is added to the predicted estimate, while the mean remains unchanged:

$$\bar{\mathbf{u}}^f(t+1) = \bar{\mathbf{u}}^a(t), \tag{6.45}$$

$$\Sigma_{\mathbf{u}}^f(t+1) = \Sigma_{\mathbf{u}}^a(t) + \Delta t\, \sigma_{\mathbf{u}}^2\, \Sigma_{\mathbf{u}}(0) \tag{6.46}$$

6.4.3 Removal of obsolete cells

The disappearance of cells is a phenomenon that needs to be taken into account in the model. The causes for this phenomenon are similar to those for the creation of new cells: existing cells are advected outside the observable area, and cells dissipate as a results of internal dynamics (they "rain-out" through precipitation, or merge with existing cells). Collapsing cells are automatically deleted when their width or intensity become lower than a certain threshold. This threshold is set to 10 km for the width (twice the spatial resolution) and 0.8 mm.h^{-1} for the intensity.

It is known that radar observations are usually good at detecting the location of precipitation fields. It is thus further assumed in the model that cells having little or no corresponding support

(i.e. rainy pixels) in the observed rainfall field can be discarded. As a consequence, cells for which the supported precipitating mass is below a certain threshold are removed from the model. The supported precipitating mass is obtained by summing up the cell's effect (intensities) at observed rainy pixels only. The proportion of precipitation obtained (with respect to the cell's total precipitation) is compared against a threshold and the cell is discarded if the proportion is below that threshold. Several thresholds were tested in the range 30 to 80%. The best results (i.e. those providing the most visually satisfactory approximation to the observed rainfall field) are obtained for the 80% threshold.

6.4.4 Assimilation of the rainfall field

When a new observation becomes available, the state can be updated given the new information, in the so called assimilation step. This is done within the Bayesian framework discussed in Section 2.3.2.

Recall that the posterior distribution over the parameters is obtained, in theory, by applying Bayes rule:

$$p(\mathbf{x}_{t+1}|\mathbf{Y}_{t+1}) = \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})\, p(\mathbf{x}_{t+1}|\mathbf{Y}_t)}{p(\mathbf{y}_{t+1}|\mathbf{Y}_t)}. \tag{6.47}$$

Unfortunately, most of the time, the normalising constant $p(\mathbf{y}_{t+1}|\mathbf{Y}_t)$ cannot be evaluated. We thus need to resort to approximation methods.

Previous work (Cornford, 2004) had focused on a maximum a posteriori approach, whereby the optimal mean parameters were computed through minimisation of the negative log-likelihood with respect to the observation. A Laplace approximation was then applied about the optimum in order to obtain an estimate of the covariance. However, in order to obtain a reasonable estimate of the covariance, the mean optimum had to be computed with great accuracy, resulting in a computationally expensive optimisation phase.

One of the aims of the present work was to improve the assimilation scheme, in particular by providing a more reliable – and possibly faster – estimation of the state's second moments. An attractive framework to address this problem is the one introduced by Hinton and van Camp (1993), in which the posterior distribution is approximated by another, tractable distribution q. The optimal parameters for this q distribution are chosen such that they minimise the Kullback-Leibler (KL) divergence to the posterior:

$$\mathrm{KL}(p \,\|\, q) = -\int q(\mathbf{x}_{t+1}|\mathbf{Y}_{t+1}) \ln \frac{p(\mathbf{x}_{t+1}|\mathbf{Y}_{t+1})}{q(\mathbf{x}_{t+1}|\mathbf{Y}_{t+1})}\, d\mathbf{x}_{t+1} \tag{6.48}$$

Because conjugate distributions were chosen for the priors, we know that the posterior has a similar structure. This allows us to derive the expression of the KL divergence in close form. This derivation is presented in Appendix A. The outline of the computation is given below.

The q distribution is chosen to have the same structure as the prior, to exploit the conjugate nature of the priors chosen. Given the prior (Equation 6.13) and the likelihood (Equation 6.9), the expression for $p(\mathbf{x}_{t+1}|\mathbf{Y}_{t+1})$ can be obtained, up to a constant, by applying Bayes' rule (Eq. 2.11), leading to:

$$\mathrm{KL}(p \parallel q) \propto -\int q(\mathbf{x}_{t+1}|\mathbf{Y}_{t+1}) \ln \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{Y}_t)}{q(\mathbf{x}_{t+1}|\mathbf{Y}_{t+1})} \, d\mathbf{x}_{t+1} \tag{6.49}$$

Expanding this yields:

$$\mathrm{KL}(p \parallel q) \propto -\left\langle \ln p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) \right\rangle_{q(\mathbf{x}_{t+1}|\mathbf{Y}_{t+1})} \tag{6.50}$$

where the cross-entropy of p with respect to q is defined as:

$$\left\langle \ln p(\mathbf{x}) \right\rangle_{q(\mathbf{x})} = -\int q(\mathbf{x}) \ln p(\mathbf{x}) \, d\mathbf{x} \tag{6.51}$$

The likelihood term arising from the new radar image being assimilated expands, after computation, and omitting the time index for readability, to:

$$-\left\langle p(\mathbf{y}|\mathbf{x}) \right\rangle_{q(\mathbf{x}|\mathbf{Y})} = \frac{1}{2\sigma^2} \sum_{j=1}^{M} \left[\left(\sum_{k=1}^{N} \mathrm{E}_{k,j} - y_j \right)^2 - \sum_{k=1}^{N} (\mathrm{E}_{k,j})^2 + \sum_{k=1}^{N} \mathrm{F}_{k,j} \right] \tag{6.52}$$

with

$$\mathrm{E}_{k,j} = \frac{\gamma_k}{\delta_k(1+\xi_k)} \left(1 + \frac{(\bar{\mathbf{c}}_k - \mathbf{s}_j)^{\mathrm{T}}(\bar{\mathbf{c}}_k - \mathbf{s}_j)}{2\beta_k(1+\xi_k)} \right)^{-\alpha_k} \tag{6.53}$$

and

$$\mathrm{F}_{k,j} = \frac{\gamma_k(\gamma_k+1)}{\delta_k^2(1+2\xi_k)} \left(1 + \frac{(\bar{\mathbf{c}}_k - \mathbf{s}_j)^{\mathrm{T}}(\bar{\mathbf{c}}_k - \mathbf{s}_j)}{\beta_k(1+2\xi_k)} \right)^{-\alpha_k} \tag{6.54}$$

Further computation leads to the following results for the prior term:

$$\begin{aligned} -\left\langle \ln \frac{p(\mathbf{x}|\mathbf{Y})}{q(\mathbf{x}|\mathbf{Y})} \right\rangle_{q(\mathbf{x}|\mathbf{Y})} = \sum_{k=1}^{N} \Bigg[& \gamma_k \ln \delta_k - \gamma'_k \ln \delta'_k - \ln \frac{\Gamma(\gamma_k)}{\Gamma(\gamma'_k)} \\ & + (\gamma_k - \gamma'_k)\left[\Psi(\gamma_k) - \ln \delta_k\right] - \gamma_k \left(1 - \frac{\delta'_k}{\delta_k}\right) \\ & + \alpha_k \ln \beta_k - \alpha'_k \ln \beta'_k - \ln \frac{\Gamma(\alpha_k)}{\Gamma(\alpha'_k)} \\ & + (\alpha_k - \alpha'_k)\left[\Psi(\alpha_k) - \ln \beta_k\right] - \alpha_k \left(1 - \frac{\beta'_k}{\beta_k}\right) \\ & + \ln \frac{\xi'_k}{\xi_k} + \frac{\xi_k}{\xi'_k} + \frac{1}{2\xi'_k} \frac{\alpha_k}{\beta_k} (\bar{\mathbf{c}}_k - \bar{\mathbf{c}}'_k)^{\mathrm{T}}(\bar{\mathbf{c}}_k - \bar{\mathbf{c}}'_k) \\ & \Bigg] - \frac{N}{2} \end{aligned} \tag{6.55}$$

in which the prime is used to distinguish the prior parameters from the equivalent parameters in the posterior.

The full derivation of these results, which requires a series of non-trivial integrals, is provided in Appendix A. The main point to note is that the use of the variational Bayes framework allows

us to repose the estimation of the posterior distribution by a minimisation of Equation 6.48. The minimisation is achieved using a scaled conjugate gradient algorithm (Bishop, 1996; Nabney, 2001). Since the q distribution is an approximation to the full posterior distribution (rather than just the mean), we hoped that fewer optimisation steps would be required to obtain a good overall estimate of the true posterior. However, this assumption proved to be wrong, most probably due to the complexity of the error function and the large number of parameters to optimise. In experiment setting, 200 optimisation steps are necessary to reach a good estimate of the state.

6.4.5 Detection of new cells

When the new observations become available, the model must be able to adjust the presence of new cells. These new cells result mainly from two sources: the spatial motion of the rainfall field while the observed area remains static, which means new cells get advected into the spatial window, and the creation of new cells as a result of the rainfall field's internal dynamics (condensation, splitting of existing cells...).

In order to detect these new cells, an algorithm similar to the one described in Section 6.3.1 is applied, whereby new cells are added until either the misfit is reduced below a threshold or the maximum number of cells is reached. Note that only the new cells are optimised, keeping all other cells parameters fixed.

6.4.6 Assimilation of the advection field

Update of previously existing cells

The advection is updated using the displacement between the updated centre $\mathbf{c}_{t+1}$ and its previous estimate $\mathbf{c}_t$ as a pseudo-observation, having mean and covariance:

$$\Delta\bar{\mathbf{c}}(t+1) = \bar{\mathbf{c}}(t+1) - \bar{\mathbf{c}}(t) \tag{6.56}$$

$$\Sigma_{\Delta c}(t+1) = \Sigma_c(t+1) + \Sigma_c(t) \tag{6.57}$$

Since both the predicted estimate of the advection and the pseudo-observation follow Gaussian distributions, the posterior (i.e. updated) advection estimate is also Gaussian-distributed, with mean and covariance:

$$\Sigma_u(t+1)^{-1} = \Sigma_u^f(t+1)^{-1} + \Delta t^2 \times \Sigma_{\Delta c}(t+1)^{-1} \tag{6.58}$$

$$\bar{\mathbf{u}}(t+1) = \Sigma_u(t+1)\left[\Delta t\, \Sigma_{\Delta c}(t+1)^{-1}\, \Delta\bar{\mathbf{c}}(t+1) + \Sigma_u^f(t+1)^{-1}\, \mathbf{u}(t)\right] \tag{6.59}$$

The dependencies between the different parameters during the propagation/assimilation step are illustrated on Figure 6.9.

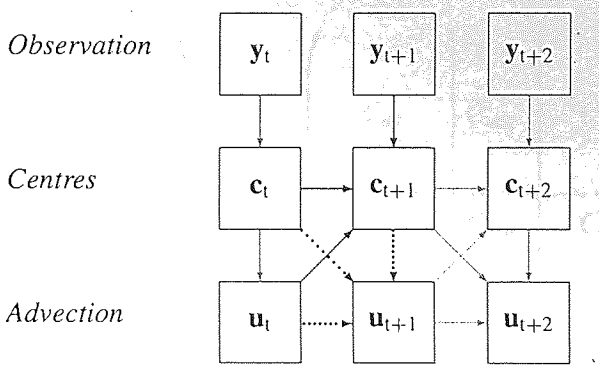


Figure 6.9: Dependencies between the advection and centres during the assimilation phase. The updated centres (blue, solid arrows) depend on their previous location and advection (through the predicted state) and on the new observation (through the assimilation) as illustrated by the blue arrows. The updated advection vectors (red, dotted arrows) depend on their previous estimates and the assimilated centres at the previous and current time step (the temporal variation between these two acting as an "observed" advection vector) as illustrated by the red arrows

Absence of cells

It might happen, particularly during summer, that no precipitation is observed over the spatial domain. This results in an empty rainfall field (no cells left in the model). The last estimate of the mean advection is kept so that, when new precipitation is observed, it can be used as a prior for the mean advection of new cells. The advection's covariance is reset to its original prior.

Advection of new cells

When new cells are detected, their advection's mean and covariance need to be specified. The predicted advection estimate at the new cells locations is computed using the Gaussian process.

Considering a new cell with advection $\mathbf{u}_* = (u_*, v_*)$, we can write the augmented vector $\mathbf{u}_+$ and its associated mean and covariance:

$$\mathbf{u}_+ = \left(\begin{array}{c} u \\ \hline u_* \\ \hline v \\ \hline v_* \end{array}\right) ; \bar{\mathbf{u}}_+ = \left(\begin{array}{c} \bar{u} \\ \hline \bar{u}_* \\ \hline \bar{v} \\ \hline \bar{v}_* \end{array}\right) , \Sigma_{\mathbf{u}_+} = \left(\begin{array}{c|c|c|c} C_{uu} & C_{uu_*} & C_{uv} & C_{uv_*} \\ \hline C_{uu_*}^{\mathrm{T}} & C_{u_*u_*} & C_{vu_*}^{\mathrm{T}} & C_{u_*v_*} \\ \hline C_{uv}^{\mathrm{T}} & C_{vu_*} & C_{vv} & C_{vv_*} \\ \hline C_{uv_*}^{\mathrm{T}} & C_{v_*u_*} & C_{vv_*}^{\mathrm{T}} & C_{v_*v_*} \end{array}\right)$$

where u and v denote the horizontal and vertical components of the previous advection estimate. For computational purposes, it is however more convenient to reorganise these by blocks, so that:

$$\mathbf{u}_+ = \left(\begin{array}{c} \mathbf{u} \\ \hline \mathbf{u}_* \end{array}\right) ; \bar{\mathbf{u}}_+ = \left(\begin{array}{c} \bar{\mathbf{u}} \\ \hline \bar{\mathbf{u}}_* \end{array}\right) , \Sigma_{\mathbf{u}_+} = \left(\begin{array}{c|c} \Sigma_{\mathbf{u}} & \mathbf{k}_{\mathbf{u}\mathbf{u}_*} \\ \hline \mathbf{k}_{\mathbf{u}\mathbf{u}_*}^{\mathrm{T}} & \Sigma_{\mathbf{u}_*} \end{array}\right)$$

with $\mathbf{u} = (u, v)$, $\mathbf{u}_* = (u_*, v_*)$ and:

$$\Sigma_{\mathbf{u}} = \begin{pmatrix} C_{uu} & C_{uv} \\ C_{uv}^{\mathrm{T}} & C_{vv} \end{pmatrix} , \Sigma_{\mathbf{u}_*} = \begin{pmatrix} C_{u_*u_*} & C_{u_*v_*} \\ C_{u_*v_*}^{\mathrm{T}} & C_{v_*v_*} \end{pmatrix} \tag{6.60}$$

In a static context, the covariances would be determined using the covariance function K and the distances between cells:

$$\Sigma_{\mathbf{u}} = \mathrm{K}(\mathbf{c}, \mathbf{c}), \ \Sigma_{\mathbf{u}_*} = \mathrm{K}(\mathbf{c}_*, \mathbf{c}_*), \ \mathbf{k}_{\mathbf{u}\mathbf{u}_*} = \mathrm{K}(\mathbf{c}, \mathbf{c}_*) \tag{6.61}$$

However, this cannot be done here, as the first equality does not hold anymore in the case of a Gaussian process which has been sequentially updated: $\Sigma_{\mathbf{u}} \neq \mathrm{K}(\mathbf{c}, \mathbf{c})$. This is due to the fact that, although $\Sigma_{\mathbf{u}}$ was initialised as $\mathrm{K}(\mathbf{c}, \mathbf{c})$, it has then been modified in time as the model got updated. Typically, the prior covariance is chosen to have a large magnitude to reflect our initial uncertainty about the advection. However, as the model gets confronted to observations and learns from them, our uncertainty decreases, resulting in a decrease in magnitude of $\Sigma_{\mathbf{u}}$.

Using the prior covariance function K to initialise the new cells covariance results in the updated matrix $\Sigma_{\mathbf{u}_+}$ becoming singular. This is most probably due to the variances of the new/old cells having different orders of magnitude.

To overcome this issue, a scaled normalised covariance function $\hat{\mathrm{K}}$ is introduced, which is similar to K but provides a variance of order 1. This is achieved by replacing in equation (6.21)

the variance terms E_Φ and E_Ψ by their normalised counterparts $\frac{E_\Phi}{E_\Phi+E_\Psi}$ and $\frac{E_\Psi}{E_\Phi+E_\Psi}$, so as to keep the same ratio between the variances for the divergent and rotational components. The variance for $\mathbf{u}_*$ is then scaled by the average variance of $\mathbf{u}$ (i.e. the average of the diagonal terms in $\Sigma_\mathbf{u}$).

The (marginal) covariance for the new cells is thus set to:

$$\Sigma_{\mathbf{u}_*} = \frac{1}{N} \mathrm{Tr}(\Sigma_\mathbf{u})\, \hat{\mathrm{K}}(\mathbf{c}_*, \mathbf{c}_*) \tag{6.62}$$

To improve numerical stability, the correlations between the new and the old cells are ignored, i.e. $\mathbf{k}_{\mathbf{u}\mathbf{u}_*} = 0$. This might seem a rather crude approximation, however, one should note that these correlations will be induced by the model within only a few assimilation steps.

As for the mean of the new cells advection, it is set to:

$$\bar{\mathbf{u}}_* = \mathrm{K}(\mathbf{c}, \mathbf{c}_*)^\mathrm{T} \mathrm{K}(\mathbf{c}, \mathbf{c})^{-1} \bar{\mathbf{u}} \tag{6.63}$$

which corresponds to the standard mean prediction for a static Gaussian process.

6.5 Forecast

Once data has been assimilated and the model has converged to a steady state, it can be used to generate forecasts. The forecasting scheme used is fairly crude and simply consists in propagating the cells forward in a linear manner, as described in Sections 6.4.1 and 6.4.2. Each cell is propagated independently, according to its advection at initial time. This advection is assumed to remain constant over the duration of the forecast. It is clear that this assumption is only relevant for forecasts at very short lead times.

The main drawback of this method is that it does not maintain the smoothness of the advection field. Cells might become neighbours as a result of their respective propagation but have very different advection estimates. Figure 6.10 illustrates this problem: two cells with initial advection from a smooth field (left) have been propagated forward in time, resulting in an unrealistic, non-smooth, advection field (right)

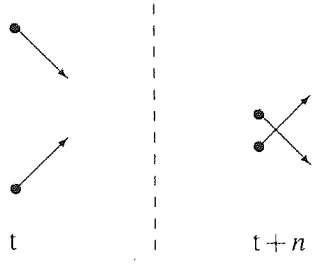


Figure 6.10: Forecast abnormality illustrated

A better forecasting scheme would involve projecting the advection field onto a fixed grid and propagate the cells advection to the estimated advection at their location. This would maintain the smoothness of the advection field in time. Although better than a linear forecast, this method also has its limitation in the fact that the field is still assumed constant through time.

Probabilistic forecast

Because the model keeps track of the uncertainty in the parameters, a probabilistic forecast can easily be generated. This can be done in two ways. The first option consists in propagating the full stochastic model forward in time. One then obtains a full probabilistic forecast. The problem with this method is that it is difficult in practice to correctly model the propagation of the variance, especially as forecast lead time increases.

An alternative option is to make use of Monte Carlo approximations. One can generate a sample from the parameters distributions at time t and forecast each sample to time $t+n$ using deterministic dynamics. An ensemble of forecasts is thus obtained, which provides an approximation to the actual forecast distribution. First and second moments of the forecast distribution can then be estimated from the sample.

In this particular problem, the dynamics are effectively linear, so both methods could be applied. Cornford (2004) chose to apply the first method. In this work, we prefer to apply the second method as it is the one used operationally in most hydrological forecasting systems.

6.6 Discussion

A model has been introduced in this chapter which is able to reproduce the main characteristic of a dynamic rainfall field. The model is decomposed as a set of Gaussian-shaped rain cells with parameters for the cell's location, its width and height (intensity at the centre). Suitable priors have been chosen for these parameters. As seen from Figure 6.5, the model provides a reasonable approximation to the real observed rainfall field.

Dynamics are modelled by an advection field. A Gaussian process prior on this advection field ensures it is spatially smooth and has a realistic behaviour. The advection is specified at each cell's centre and used to propagate the cells linearly.

Variational inference is used to compute the posterior distribution for the cells parameters : hyperparameters are determined which minimise the KL-divergence between the true posterior distribution and a suitable approximating distribution. This results in an updated rainfall field, from which obsolete cells (i.e. cells that are not matched in the observed rainfall field) are removed.

From the updated cells locations, the cells displacement is computed and used to update the advection field. If new cells appear on the observation, they are detected and incorporated into the model. A suitable prior is used to initialise their advection while ensuring the model remains numerically stable.

In the following chapter, this model is run on synthetic data for validation before being applied

to real data. Details about these experiments are provided and the results are analysed using state of the art verification methods.

7 Bayesian precipitation nowcasting: Results

CONTENTS

7.1 Preliminary experiment: synthetic data

A stochastic rainfall prediction model has been introduced in Chapter 6. The current chapter presents results obtained with this model. Simple validation experiments are considered first, in which the model is tested on synthetic data involving a single cell and several cells with linear advection. The model is then tested on real radar data in two contexts: a frontal event and a convective event. Validation methods are applied to the results and results discussed.

7.1.1 Single cell experiment

As a sanity check, we first test the ability of the model to correctly locate a single precipitation cell, on simulated data. The data is generated by propagating a single precipitation cell with constant, horizontal advection of 6 $m.s^{-1}$ (Figure 7.1). The cell is 42 km wide and its central intensity is 33 mm.h^{-1}.

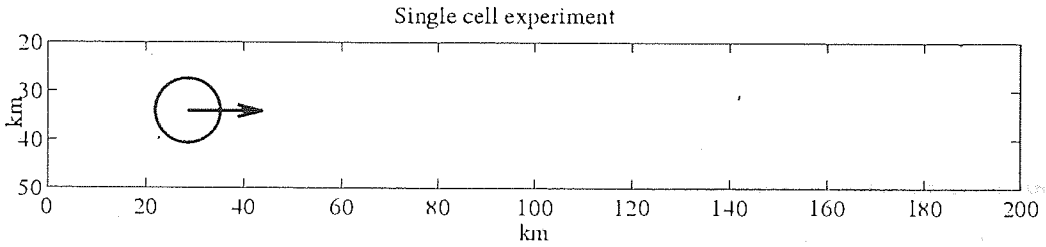


Figure 7.1: Single cell experiment: experiment setting

Noise models Observations are taken every 900 s and corrupted with additive Gaussian white noise with variance 4.0 $\text{mm}^2.\text{h}^{-2}$. The noise is applied locally, i.e. only to those pixels where the precipitation rate exceeds a threshold $\mathbf{y}_{min} = 0.8$ mm.h^{-1}.

$$\mathbf{y}_{noisy} = \begin{cases} \mathbf{y}_{true} + \varepsilon & \text{if} \quad \mathbf{y}_{true} \geq \mathbf{y}_{min} \\ \mathbf{y}_{true} & \text{if} \quad \mathbf{y}_{true} < \mathbf{y}_{min} \end{cases} \tag{7.1}$$

This is believed to be a realistic noise model for simulated radar-like data, since it is empirically observed that radar is usually well able to identify geographical regions without precipitation, with the errors occurring mainly in the regions where precipitation is detected (Meischner, 2004).

The following other two noise models were also tested. Global additive noise (applied to the whole radar field):

$$\mathbf{y}_{noisy} = \mathbf{y}_{true} + \varepsilon \tag{7.2}$$

and global multiplicative noise (also applied to the whole radar field):

$$\mathbf{y}_{noisy} = \mathbf{y}_{true} \times (1 + \varepsilon) \tag{7.3}$$

These did not lead to any significant difference in the results, so the first noise model (local additive) is retained for this synthetic experiments.

The cell parameters are estimated over 180 15-minute time steps (equating to 45 hours), and then forecast over 12 steps (3 hours). The initial variances over the centres, widths and heights are set respectively to 10.0 km^2h^{-1}, 10.0 $km^2h^{-1}$and 10 mm^2h^{-3}, representing weak knowledge of the initial state at the start of the assimilation. The propagation error variance (also known as the model error component) is set to 25 $km^2h^{-1}$for the centres, 0.1 $mm^2h^{-3}$for the central intensity, 0.1 $km^2h^{-1}$for the cell widths and 1.6 $m^2s^{-2}h^{-1}$for the advection field. These priors are chosen to be characteristic of the typical model errors we might expect for real data, and have been derived by analysis of radar image sequences.

Figure 7.2 shows the predicted, observed and assimilated cells from the idealised model at 3 different times. At the end of the assimilation period, the cell is propagated forward in time using the last estimate of the parameters. There is no noticeable difference between the cell's characteristics at the end of the forecast and the observed cell, which confirms the model managed to track the "true" cell on this very simple validation example.

Figure 7.3 shows the evolution of the parameter error during the experiment (i.e. the difference between the estimated parameter and the true value). The top row plots the error for the cell's coordinates (horizontal on the left, vertical on the right). The 3rd row shows the error for the advection components (horizontal on the left, vertical on the right). It is clear that these 4 parameters converge within a few assimilation steps and then oscillate about the true value. The 2nd row displays the error for the width (w) and rain intensity (r), which are more difficult to learn, which seems to be related to the noise added to the observation and discretisation issues. On the bottom row, the ξ scale factor is plotted (left), showing a stable behaviour during the assimilation phase and growth during the forecast phase (which is to be related to the increase in uncertainty about the centres). The Root Mean Square Error (RMSE) is plotted in the bottom right corner (Section 7.3.1 provides details on the computation of the RMSE). The RMSE is kept down to a very small value on average during the whole assimilation phase, and grows almost linearly during the forecast due to small errors in the parameters (especially advection) leading to the cells trajectory diverging slightly from the truth.

7.1.2 Multiple cells experiment

A similar experiment to the one described previously is carried out, this time with a set of several cells. The advection is set to 6 $m.s^{-1}$ in the horizontal direction for all cells, but is perturbed with Gaussian noise so that cells have slightly different velocities. Observations are corrupted with

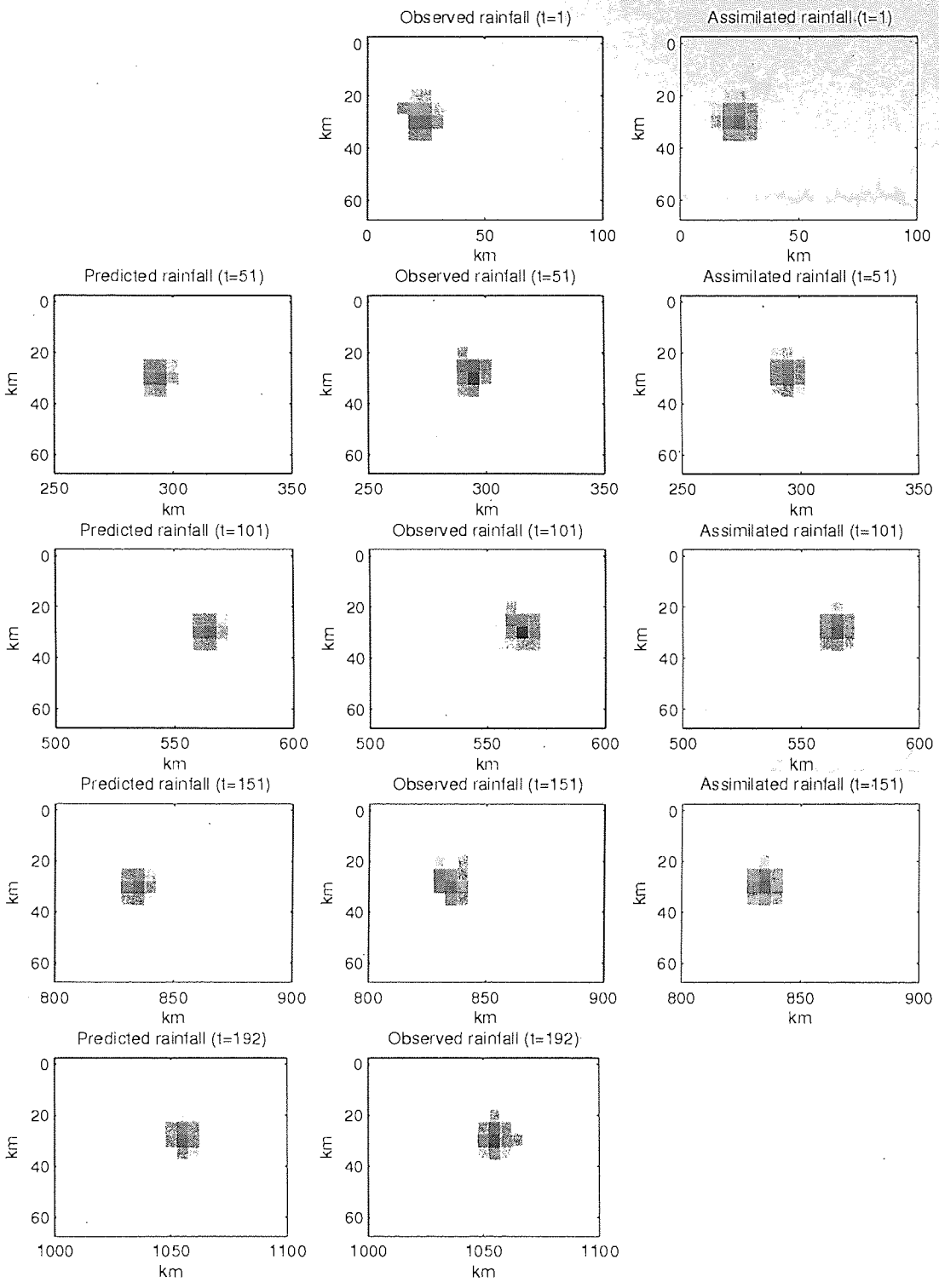


Figure 7.2: Single cell experiment: assimilation and prediction results. This plot shows the predicted precipitation cell (left), the observed precipitation cell (centre) and the assimilated precipitation cell (right) at different time intervals. The initial guess is displayed on the top row, followed by three assimilation steps on rows 2-4, and the end of the forecast is shown on the bottom row. Note that the x-axis is translated to follow the cell.

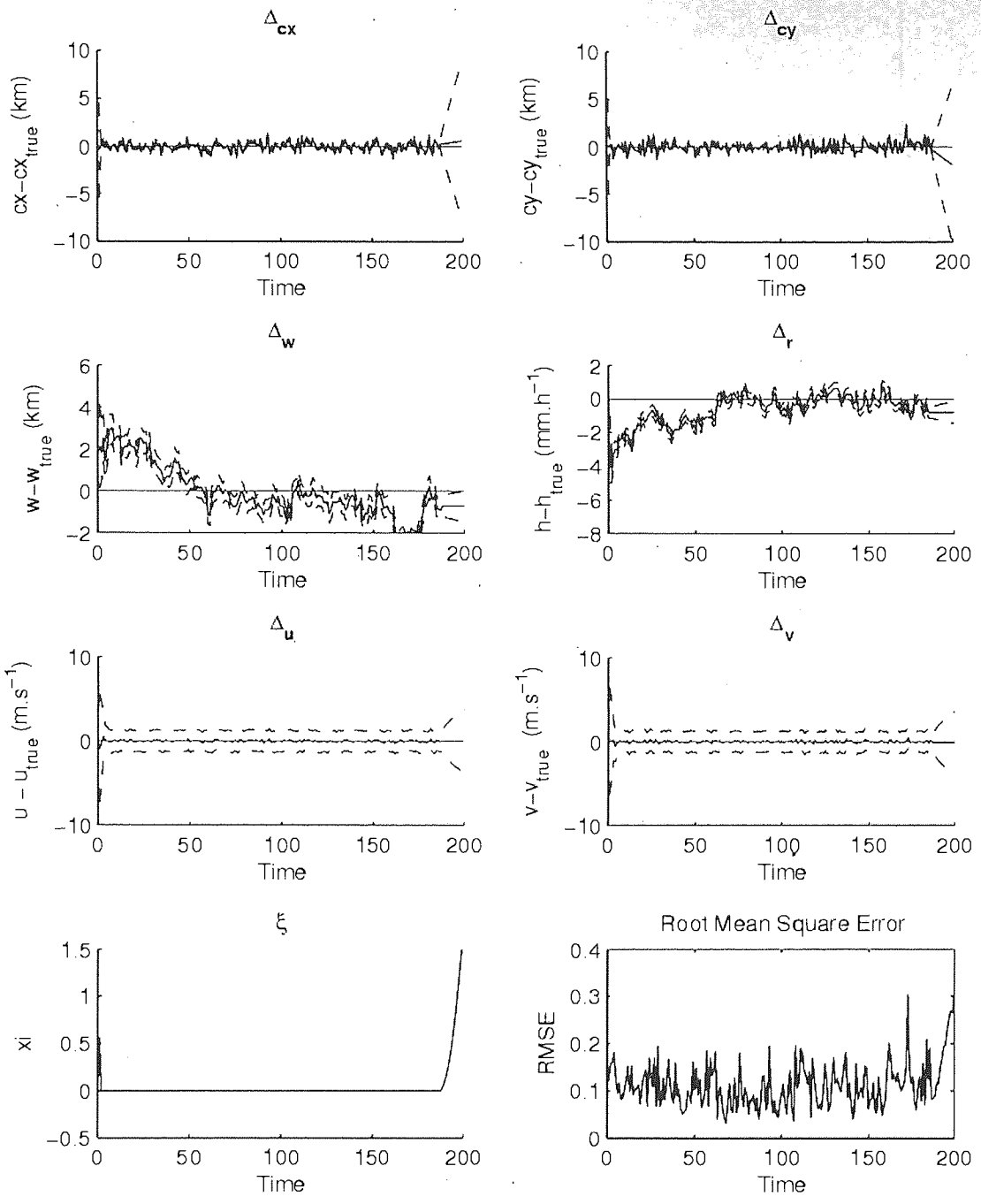


Figure 7.3: Single cell experiment: parameter estimation. This figure shows the error for the parameters, with from left to right, starting from the top: centre error, horizontal (Δ_{cx}) and vertical (Δ_{cy}); width (Δ_w); central rain rate (Δ_r); advection component, horizontal (Δ_u) and vertical (Δ_v); width/centre correlation scale factor (ξ); Root Mean Square Error (RMSE) between true cell and model's estimate/prediction.

local additive Gaussian white noise with variance 4.0 $mm^2.h^{-2}$.

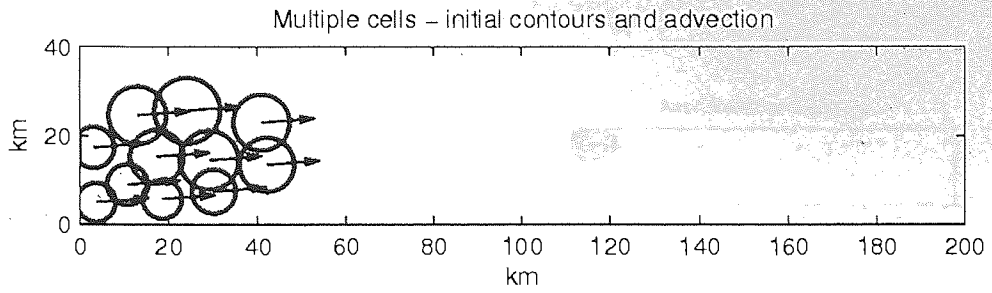


Figure 7.4: Multiple cells experiment: experiment setting

The true cells are propagated 100 time steps. The model's parameters are determined using the first 88 steps and forecast during the 12 remaining steps. Figure 7.5 shows, in the upper part, the evolution of the synthetic observed rainfall field (left) and that of the corresponding estimated rainfall field (right). In the lower part, the root mean square error between the observed and estimated/predicted rainfall field is plotted. The model is able to capture the rainfall field and track it throughout the experiment. It is also able provide a reasonable forecast of the rainfall field, which has become a set of separated cells due to the different velocities of the cells.

These two trivial experiments on synthetic data have confirmed the ability of the model to track very simple rainfall fields. The next challenge is to test the model on real radar data.

7.2 Real data experiment

Having validated the model on two very simple examples, we move on to applying it to the realistic problem of real precipitation nowcasting.

There are two main types of precipitation than can be observed in the UK: convective (or convectional) and frontal. *Convective precipitation* results from the evaporation of ambient air due to heat at ground level. This creates a current of warm rising air, which cools down as it travels through higher, colder air masses. Condensation occurs, resulting in clouds and showers. Convective precipitation in temperate areas is typically characterised by local, heavy showers of relatively short duration occurring during the summer (Barry and Chorley, 2003; Jennings, 2005).

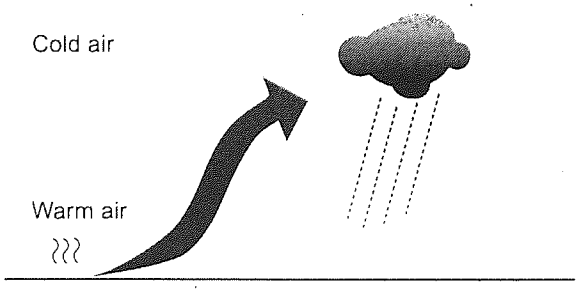


Figure 7.6: Convective precipitation

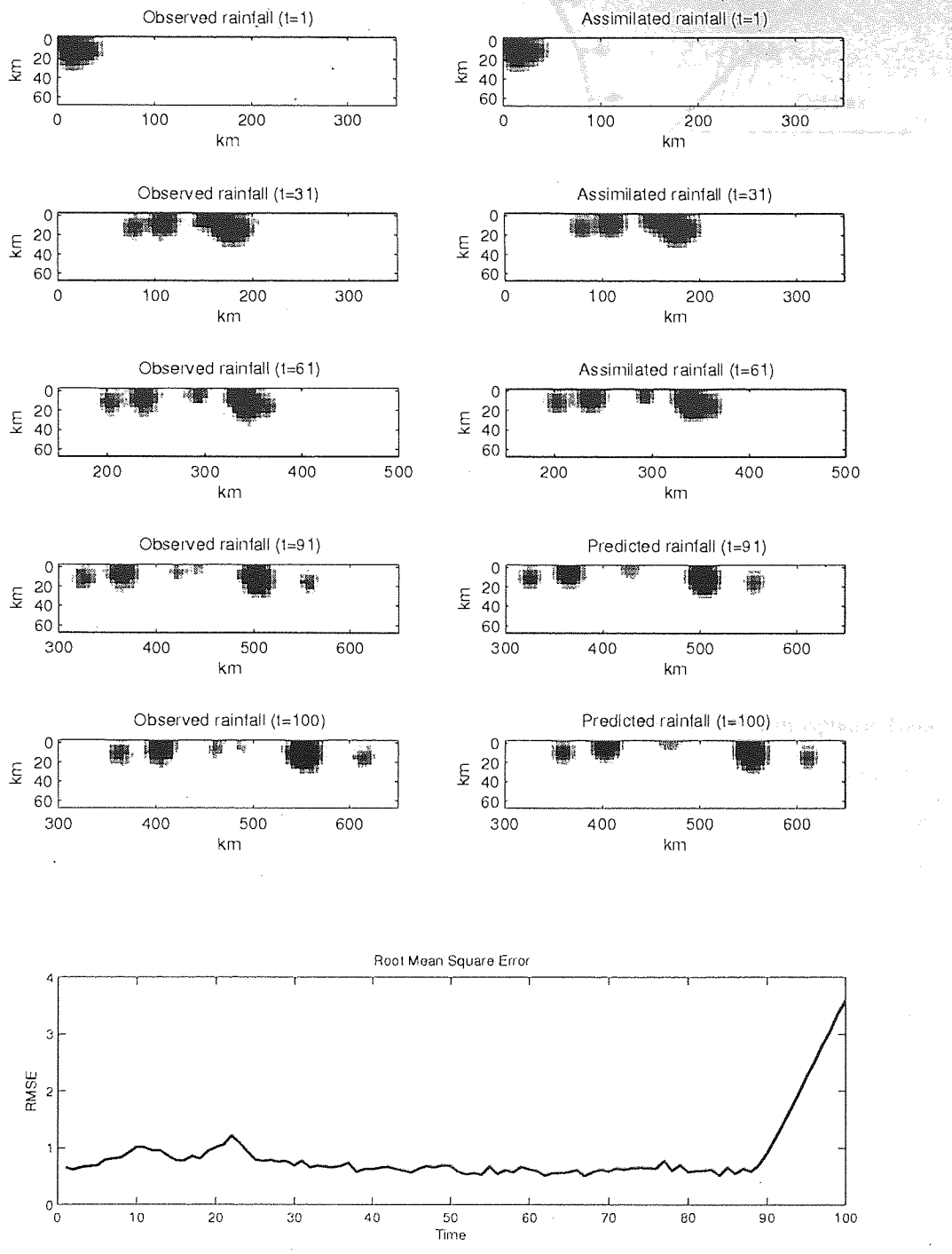


Figure 7.5: Multiple cells experiment: assimilation and prediction. The observed precipitation field (left) and the assimilated / predicted precipitation field (right) at different time intervals (note the change of location on the x-axis). The initial guess is displayed on the top row, followed by two assimilation steps on rows 2-3 and one prediction step on row 4. The end of the forecast is shown on row 5. The bottom plot shows the Root Mean Square Error.

Frontal precipitation, also referred to as stratiform or cyclonic precipitation, happens when a mass of cold air encounters a mass of warm air, causing condensation and precipitation to happen. The meeting boundary of the two air masses is called a *front* and is identified as a *cold front* if the cold mass of air pushes into the warm mass, and *warm front* in the opposite case. Frontal rain is typically lighter and longer lasting than convective precipitation and much more spread spatially.

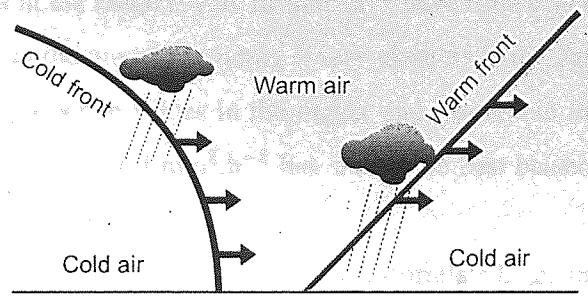


Figure 7.7: Frontal precipitation

Two datasets have been selected to test the performance of the model on both types of precipitation. The first dataset consists in a weeks worth of radar observations for a convective event in July 2006, while the second dataset looks at a frontal event over the same duration in January 2005. Details on the experimental design are provided in the following paragraphs.

7.2.1 Experimental design

Nature of the data

The data used in this experiment has been provided free of charge by the British Atmospheric Data Centre (UK Meteorological Office, 2003). The data consists in composite radar scans obtained from the NIMROD radar network (Golding, 1998). These scans have a 5×5 km^2 grid resolution, for a total size of 1725×2175 km^2 (345×435 pixels) and the time between two consecutive scans is 15 minutes. The observations were pre-processed using a Gaussian filter with radius 10 km to improve the estimation of the model precipitation field, as detailed in Section 6.2.1.

In the following experiments, we restrict ourselves to an area of 500×500 km^2, i.e. 100×100 pixels, which is wide enough for the rainfall field to present interesting structure, but small enough for the computations to be handled by a standard single core desktop computer (3GHz CPU with 1GB RAM).

The assimilation is carried out over a sequence of 672 observations, corresponding to a weeks work of data (168 hours).

Observation error

Estimating observation error is a difficult task, mostly due the variety of factors responsible for the inaccuracies in such measurements.

A very simple error model is used in these experiments : observation error is assumed to

be Gaussian with zero mean. Variance values in the range 0.1 to 10 $mm^2.h^{-2}$ have been tested. Values in the lower end of the range resulted in the model providing a very close fit to the data but less smoothness in the parameters evolution, while values in the higher end resulted in the observations being discarded by the model. A value of 4 $mm^2.h^{-2}$ has shown the best balance between the observations and the model.

It is convenient to further assume that the observation error is spatially uncorrelated, i.e. the covariance matrix $\mathbf{R}$ is diagonal, for reduced computational complexity. However, this is an unrealistic assumption, as the causes of radar error are likely to induce spatial correlations. For instance, ground clutter and insect clouds are responsible for spatially structured errors. Errors in the formulation of the observation operator h are also space-dependent and likely to introduce spatially correlated errors. The specification of radar observation covariance matrix is still very much an active area of research (see Keeler and Ellis (2000) and references therein).

Number of cells

In order to specify the (maximum) number of cells to be used in the model, a quick experiment is set in which a sample observation is fitted with an increasing number of cells. The fitting is done according to the procedure described in Section 6.3.1, with a minimum error $E_{min} = 0.03$. Figure 7.8 shows how the Root Mean Square Error (dashed line) decreases as more and more cells are allowed into the model. However, this comes at the cost of computation time (solid line), which increases almost linearly with the number of cells. A limit of 250 cells is sufficient to discard more than 90% of the misfit while keeping initialisation time below 30 seconds. This is the limit retained in the following experiments.

The sample rainfall field for this experiment has been chosen complex enough to provide an estimate of the required number of cells likely to remain valid for most observed rainfall fields. Figure 7.9 shows how the estimated rainfall field evolves as the number of cells increases. A reasonable fit is obtain for about 200 cells, with the main features being accurately reproduced. However, it takes up to 400 cells for the lower precipitation areas to be included in the estimate. This is a consequence of the model putting priority on accurately tracking high precipitation areas: additional cells are used to improve the fit in such areas rather than to improve the quality of the overall spatial field.

The initialisation process takes 480 cells to reach the minimal error chosen (3%). A closer fit to the data could be obtained by simply lowering this minimal error convergence criteria, at the expense of even more cells. However, the computation cost implied makes very large numbers of cells prohibitive in practice.

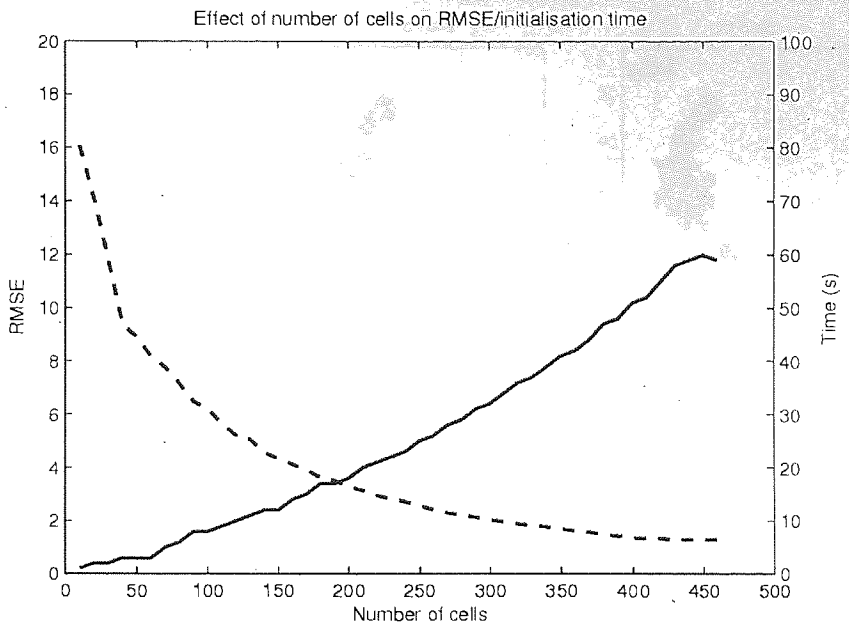


Figure 7.8: Trade off between number of cells and optimisation speed. This plot shows the effect of the number of cells used on the quality of the fit and computation time. The dashed curve (left y-axis) is the root mean square error between the observation and the model's estimated rainfall field. The solid curve (left y-axis) is the initialisation time in seconds.

7.2.2 A convective event: July 2006

The data for the convective event spans from July 6, 2006 at 3:15 am UTC to July 13, 2006 same time (672 observations).

Figure 7.10 shows the state of the model for the convective event, after about 67 hours of data (268 observations) have been assimilated. 4 hourly estimates are shown. This corresponds to a peak in the complexity of the precipitation field, and thus to a peak in the Root Mean Square Error with respect to the mean of the posterior distribution, as shown in Figure 7.12. Reading from left to right, Figure 7.10 shows: the radar observations of precipitation, the corresponding estimated precipitation after assimilation of the observed radar, the cell contours along with their advection vectors (only the major cells are displayed for clarity) and the KL convergence curves for each of the 4 time steps considered.

Note that the choice of showing the model 67 hours into the assimilation process is based on no particular reason, in particular the model does not need to assimilate that much data to provide a good estimate of the rainfall field. In fact, a good estimate is normally reached within 5 time steps of the initial guess.

Figure 7.10 shows that after seeing 67 hours of data the model is able to assimilate the radar data to estimate the precipitation field while also jointly estimate advection vectors for the precip-

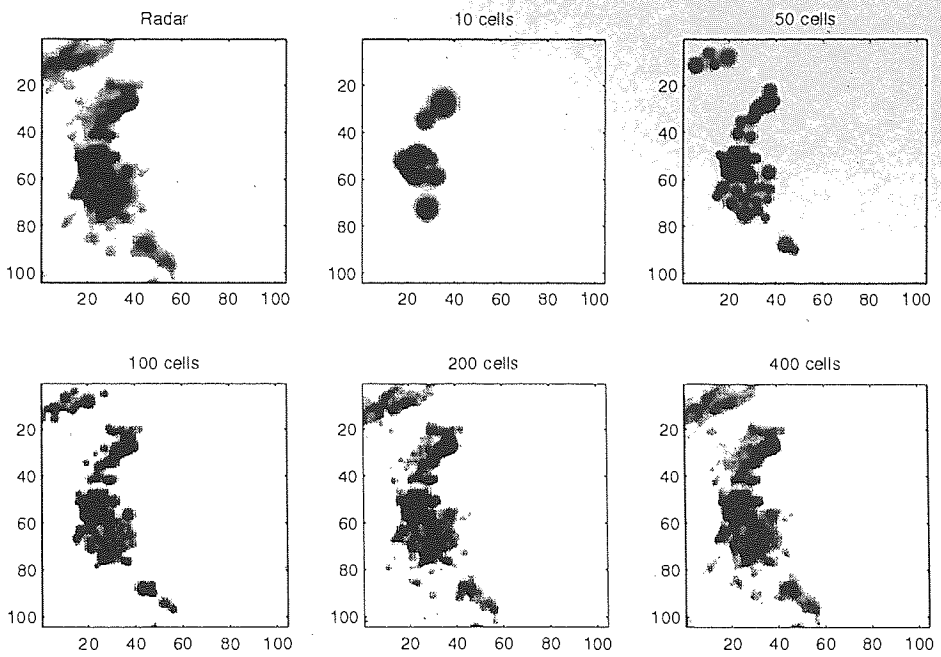


Figure 7.9: Effect of increasing the number of cells on the quality of the modelled rainfall field. This plot shows the model's realisation as the number of cells increases. The actual observation is shown in the top left corner.

itation field. It can be noticed that the image appears somewhat "grainy" in some parts. This is possibly related to the partial optimisation of the KL-divergence; as can be seen from the bottom line of the figure, the KL-divergence has not converged fully in the optimisation, much like the 3D VAR cost function is only partially optimised in classical data assimilation. This might also be related to the convective nature of the event, with the birth / death processes of the 'precipitation cells' making it very difficult for the model to track specific precipitation features, and resulting in cells being frequently added and removed. The advection vectors show a clear storm motion from south-west to north-east, but there are small variations in the advection over space, which are likely to reflect differential development and possibly steering of the precipitation field.

7.2.3 A frontal event: January 2005

The data for the frontal event spans from January 4, 2005 10:00 am UTC to January 11, 2005 same time (672 observations).

Figure 7.11, shows the same information as Figure 7.10 but for the January 2005 frontal rainfall event, starting this time after 60 hours of data (240 observations) have been assimilated. We again see a good fit of the assimilated precipitation field, but again note a problem with some rather "grainy" behaviour in the assimilated estimate of precipitation. This problem appears to be most

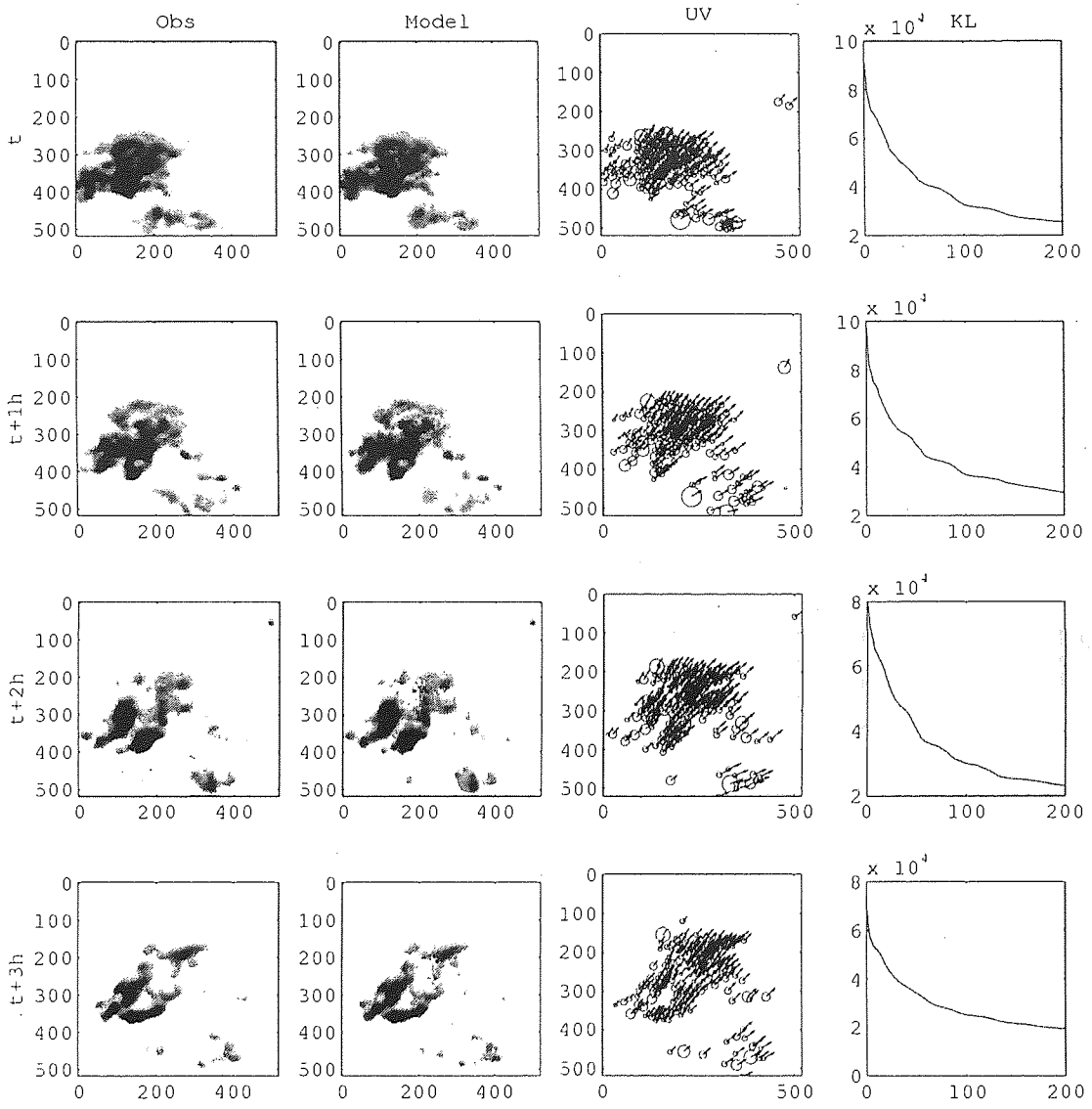


Figure 7.10: Estimation of convective precipitation (July 2006). This plot shows, from top to bottom, 4 consecutive hourly snapshots of the frontal rainfall field. Columns show, from left to right: the observed rainfall field, the modeled rainfall field, the principal cells with corresponding advection, the optimisation of the KL divergence over 200 optimisation steps.

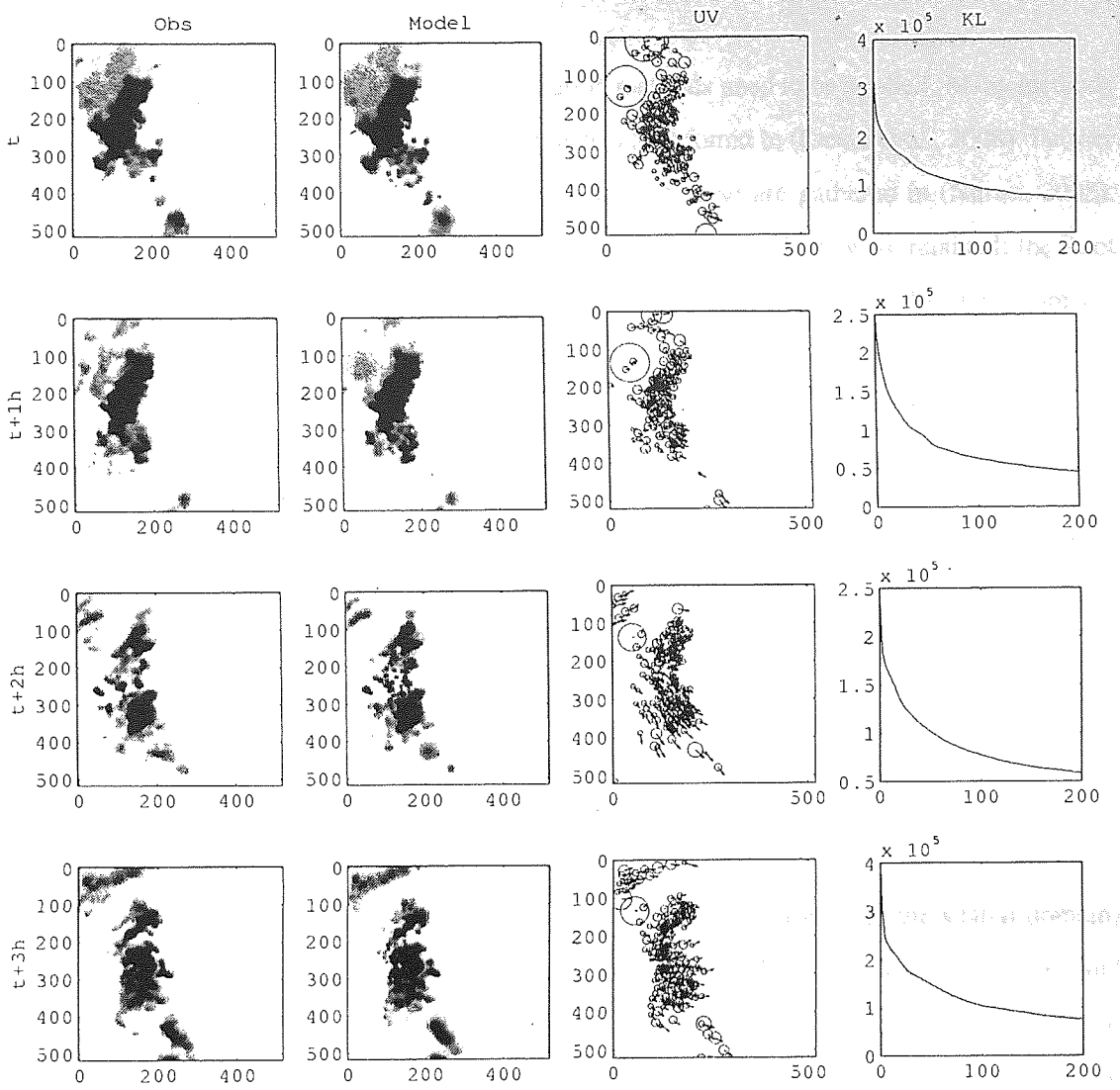


Figure 7.11: Estimation of frontal precipitation (January 2005). This plot shows, from top to bottom, 4 consecutive hourly snapshots of the frontal rainfall field. Columns show, from left to right: the observed rainfall field, the modeled rainfall field, the principal cells with corresponding advection, the optimisation of the KL divergence over 200 optimisation steps.

severe in region with strong dynamic changes to the advected precipitation field. The advection vectors from this example show rather complex structure. Initially this was felt to be a problem with the model, however it appears that there is strong apparent differential advection in this storm, probably related to embedded precipitation elements within the frontal zone, particularly early in the time window show here. At the later times the advection seems more uniform across the domain considered. We note that for computational reasons we truncated the optimisation of the KL-divergence at 200 iterations, however it is clear that the system has not fully converged.

7.3 Validation

In order to assess the quality of the model, verification methods need to be applied. Many methods have been developed to that effect, a review of which can be found in (Casati et al., 2008). Further details on the actual computation of the most common of these are gathered in (Nurmi, 2003). For the purpose of validating the current model, three validation methods were retained: the Root Mean Square Error, ROC curves and variograms. These are discussed in the following sections.

7.3.1 Root Mean Square Error

A common measure of quality for a deterministic model is given by the Root Mean Square Error between the model's estimate and the "true" value. Most of the time, this true value is not available and one has to resort to using the observations as the best estimate to the truth.

If $\mathbf{z}$ and $\mathbf{y}$ denote the estimate and the truth respectively, then the RMSE is given by:

$$\mathrm{RMSE}(\mathbf{z},\mathbf{y}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(z_i - y_i)^2} \tag{7.4}$$

It should be noted that the RMSE is particularly sensible to large errors due to the square in the difference (Nurmi, 2003).

For precipitation fields, the RMSE measures the average distance (over the spatial domain) between the model's estimate and the "true" field as measured by the radar. Figures 7.12 (top) and 7.14 (top) show the evolution of the RMSE in time over the assimilation period. The variability of the RMSE is to be related to that of the rainfall field's complexity. Total precipitation is plotted on Figures 7.12 (bottom) and 7.14 (bottom) for comparison.

The assimilation appears to give better results for the convective event (Figure 7.12) than for the frontal event (Figure 7.14). This is to be related to the overall spatial complexity of the precipitation field, which greater in the winter event, probably due to the higher overall rain rates, and the greater part of the domain covered by precipitation compared to the summer event.

Figures 7.13 and 7.15 show scatter plots of the RMSE against the total observed precipitation (summed over the spatial domain) for the convective and frontal cases respectively. This confirms, in both cases, the correlation between the complexity of the precipitation field and the quality of the corresponding estimate.

For both cases the same, limited, number of cells (250) was used which is probably not overly realistic. In practice it would be very desirable to be able to estimate an appropriate complexity for the model, which could adapt to the complexity of the observations. This was not implemented in this version of the code but we believe that it might be possible to incorporate a 'sparsity' prior in the model, in a similar manner to the treatment of the relevance vector machine (Tipping, 2001).

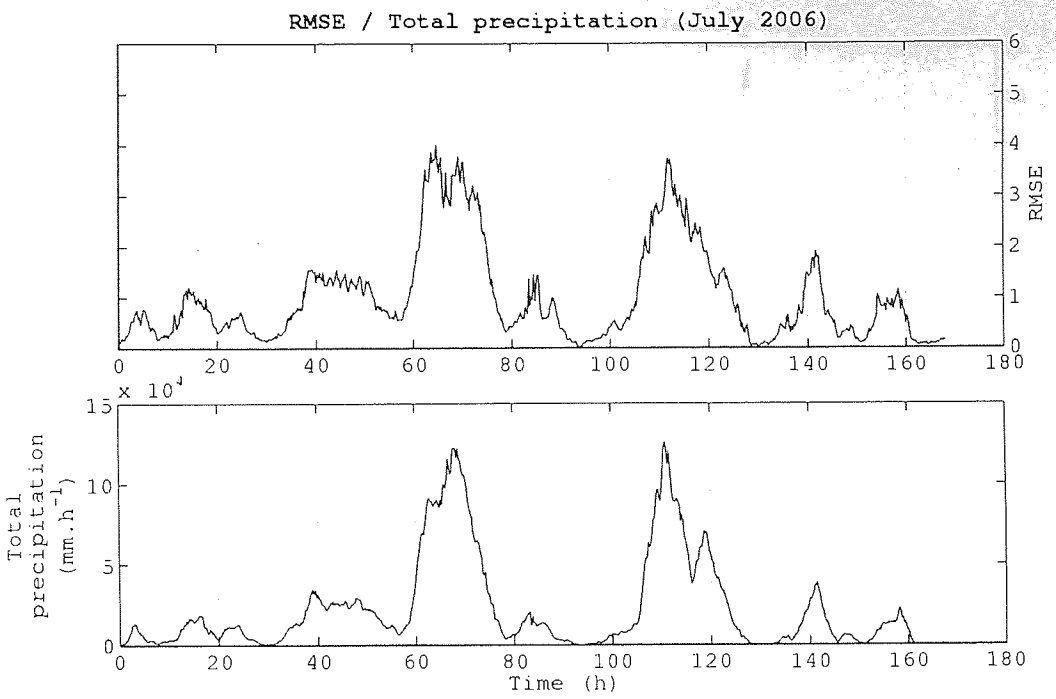


Figure 7.12: RMSE and total precipitation for a convective event (July 2006)

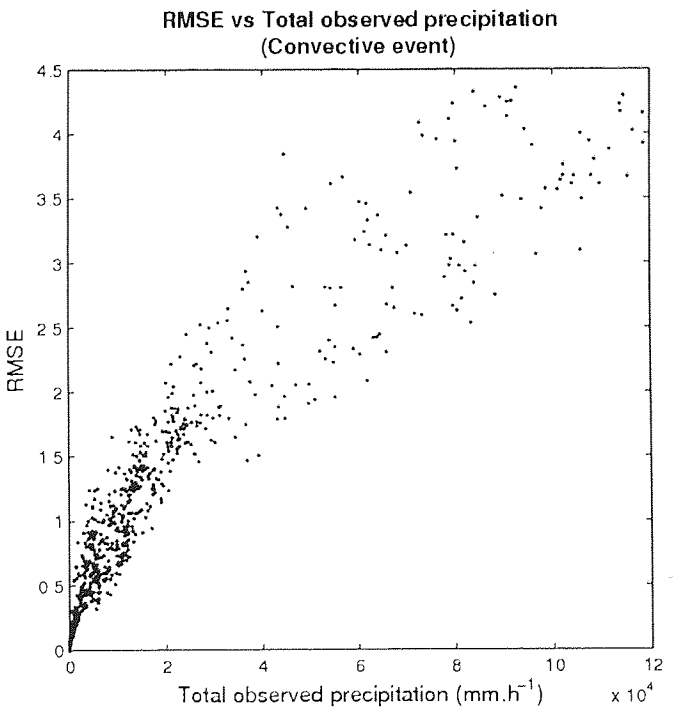


Figure 7.13: Scatter plot RMSE / Total precipitation for a convective event (July 2006)

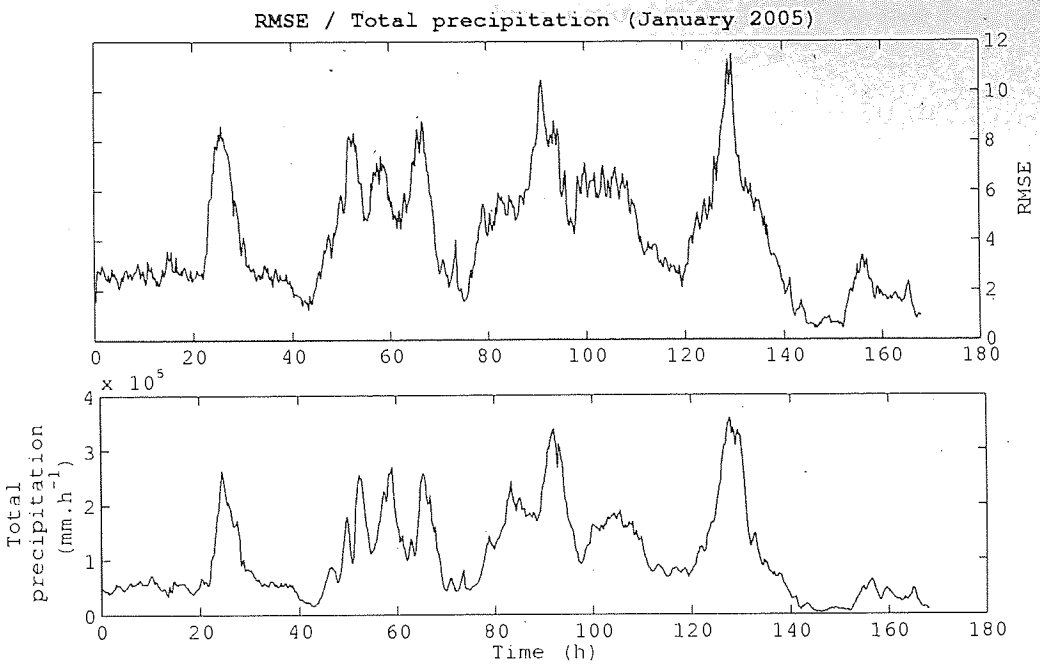


Figure 7.14: RMSE and total precipitation for a frontal event (January 2005)

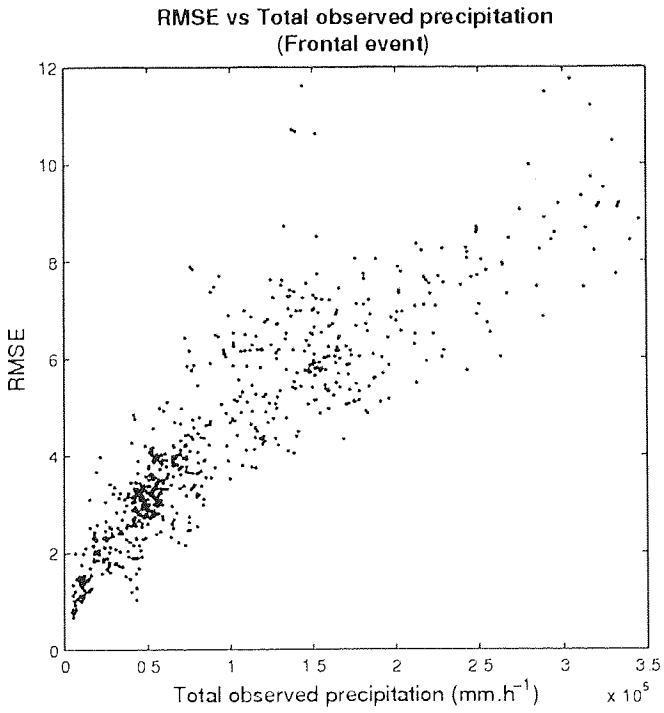


Figure 7.15: Scatter plot RMSE / Total precipitation for a frontal event (January 2005)

The model is thus better able to represent the simple, *localised* precipitation patterns which make up most of the convective data compared to the complex, diffuse precipitation patterns from the winter data. This also explains the considerable variations in the RMSE for each event, where quiet, dry(er) periods alternate with stormy phases.

7.3.2 Receiver Operating Characteristic (ROC) curves

The RMSE presented above provides a good first estimate of the quality of the assimilation, but doesn't take into account the probabilistic nature of the estimate. ROC curves, a validation method for probabilistic forecasts traditionally used in signal detection, medical and psychological applications, is now commonly applied to weather forecasting too. Examples of its application can be found in Harvey et al. (1992); Zhang and Casey (1999); Buizza et al. (1999).

ROC curves provide a way to estimate the "skill" of a model, i.e. its ability to correctly detect whether an event occurs or not in a dataset (a disease amongst a group of patients, rainy pixels on a radar image...). A clear and thorough introduction to the ROC curve is provided by (Fawcett, 2005). We will now detail the basics of ROC curve estimation using a simple example taken from an hypothetical medical domain situation, before moving on to their application to precipitation forecasts.

Introduction using an example

Let us assume we want to assess the quality of a medical model for detecting a disease X. We have at our disposition a set of patients presenting various blood concentrations in a given protein we think is linked to the presence of the disease. We assume this concentration varies between 0 and 1 for the purpose of this example. A simple model is devised by assuming that there is a given threshold τ of that concentration above which a patient is X-positive (i.e. carries the disease). We can then classify patients in two sets: "X-positive" and "X-negative", which we will simply denote by "positive" and "negative". Table 7.1 illustrates the result of such classification on a set of 1000 patients.

	Number of patients
Positive	30
Negative	970

Table 7.1: Contaminated patients (predicted)

Let us assume we can then perform a second medical test and determine accurately whether

patients are actually infected or not. We can now validate our prediction against the test, and split our positive and negative categories each into two sub-categories, depending whether they have been classified correctly or not.

Patients that were predicted "positive" are called "true positives" (TP) if the test also detected them positive, "false positive" (FP) otherwise. Similarly, patients detected "negative" by both the model and the test are called "true negative" (TN) while those incorrectly detected "negative" are called "false negatives" (FN). Table 7.2 summarises these denominations.

		TEST	
		Positive	Negative
PREDICTION	Positive	TP	FP
	Negative	FN	TN

Table 7.2: Classification of correct prediction against test

Assuming 27 of our patients were incorrectly detective positive and 3 incorrectly negative, we obtain the classification presented in Table 7.3.

		TEST	
		Positive	Negative
PREDICTION	Positive	27	3
	Negative	199	771

Table 7.3: Classification of prediction results against test

In order for these values to be plotted on a graph, two additional measures are computed:

- the *Hit Rate* (or true prediction rate):

$$HR = \frac{TP}{TP+FN}, \tag{7.5}$$

- and the *False Alarm Rate* (or false prediction rate):

$$FAR = \frac{FP}{FP+TN}. \tag{7.6}$$

Points on the ROC curve are obtained by plotting the value of HR against that of FAR. For our example, $HR = 0.1195$ and $FAR = 0.0039$. Figure 7.16 shows the ROC curve obtained for this example. The arrow indicate the location of our point.

The other points on the curve are obtained by varying the decision criterion. In our case, the threshold τ controls whether patients are detected positive or not. Different values of τ give

different points. Table 7.4 shows how the values of FAR and HR evolve as τ is increased and Figure 7.16 shows the resulting curve.

	τ	TP	FN	TN	FP	FAR	HR
	0.00	965	35	0	0	1.00	1.00
	0.10	866	53	57	24	0.30	0.94
	0.20	756	75	148	21	0.12	0.91
	0.30	623	115	242	20	0.08	0.84
	0.40	476	142	362	20	0.05	0.77
	0.50	306	201	482	11	0.02	0.60
	0.60	131	238	624	7	0.01	0.36
→	0.70	27	199	771	3	0.00	0.12
	0.80	3	62	935	0	0.00	0.05
	0.90	0	0	1000	0	0.00	0.00
	1.00	0	0	1000	0	0.00	0.00

Table 7.4: Computation of the points on the ROC curve

Note that a threshold concentration $\tau = 0$ corresponds to the extreme case where 100% of the patients are detected positive, while $\tau = 1$ corresponds to the opposite case where all patients are tested negative.

The strong dashed line on Figure 7.16 corresponds to a random classifier, i.e. a model which diagnoses patients positive and negative with equal probability. As τ increases, the number of true positives decreases the points slides up the diagonal line (Fawcett, 2005). Figure 7.17 shows the ROC curve for a random classifier applied to the same problem. A perfect model (i.e. a model in perfect agreement with the test), on the other hand, would have a ROC curve corresponding to the upper left triangle with, for all values of τ, $(FAR, HR) = (0,1)$. The ROC curve for a reasonable classifier lies between the diagonal (random model) and the upper left triangle (perfect model). A measure of the skill, determined from the ROC curve, is the area under the curve (AUC). The closer that area is to 1, the more skillful the model.

Application to probabilistic forecasts

This section details the application of ROC curves to the forecast scheme described in 6.5, following (Atger, 2001). In order to obtain a probabilistic forecast, an ensemble of model states are generated from the distribution at an initial time t and propagated forward to the lead time of interest. We restricted ourselves to an ensemble size of a 100 forecasts. For a chosen precipitation

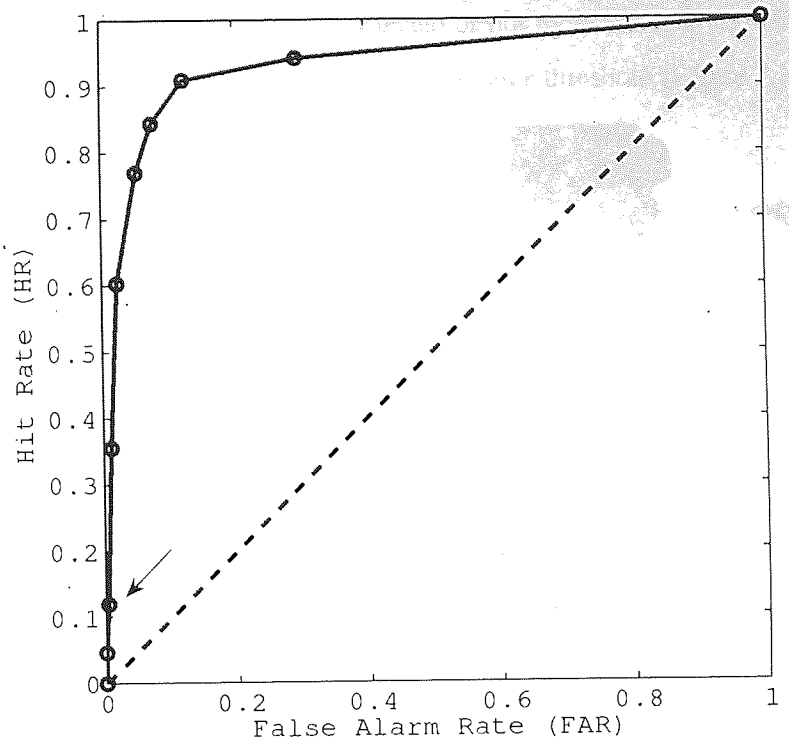


Figure 7.16: ROC curve for disease detection model

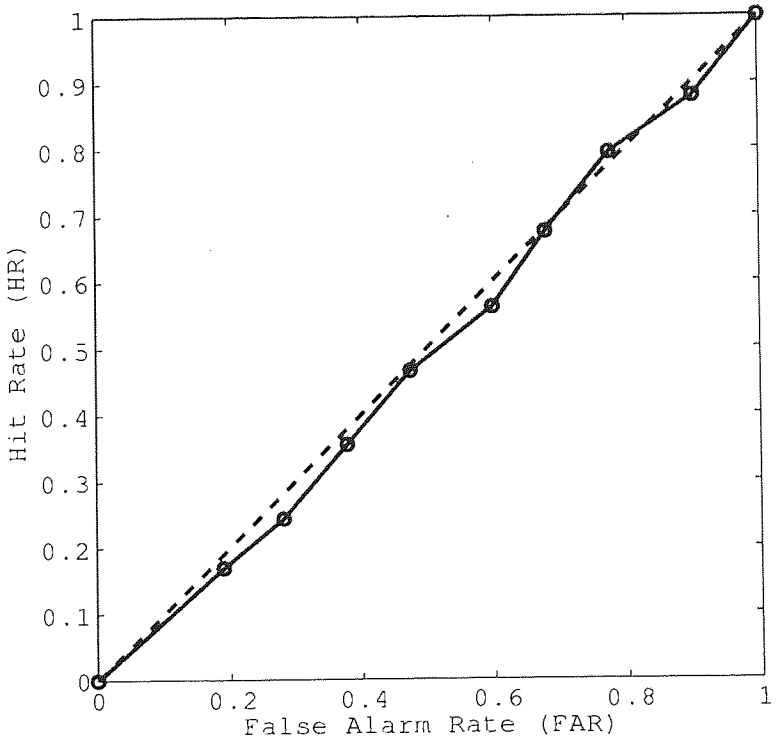


Figure 7.17: ROC curve for a random model

threshold R_{min} (e.g. 5 mm.h^{-1}), each forecast is converted into a series of positive and negative pixels, indicating whether precipitation is detected or not by that forecast at each location. The observation field is also converted to binary using the same threshold (Figure 7.18).

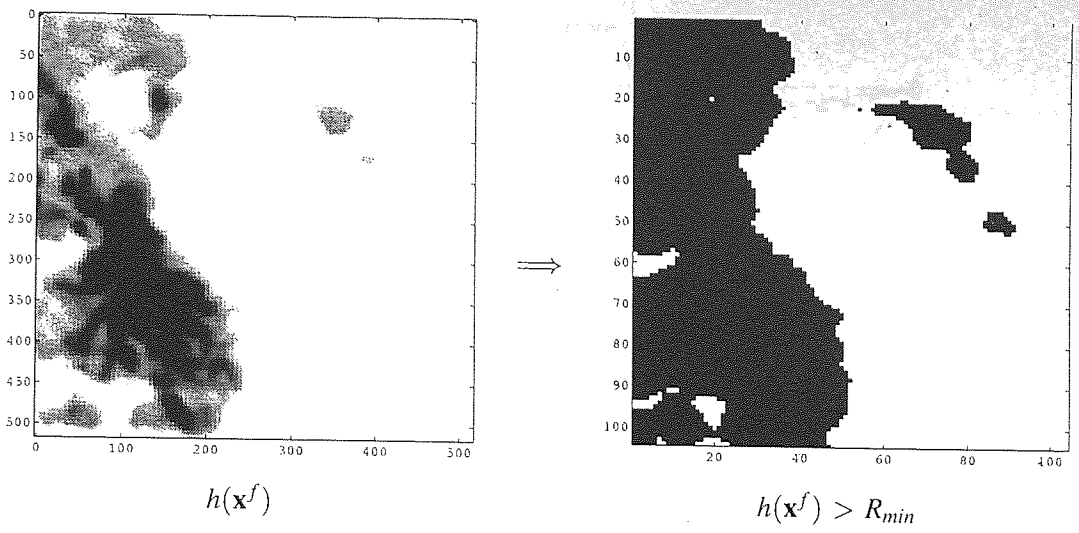


Figure 7.18: Rainfall field converted to a binary field

Given the ensemble of binary forecasts, a unique composite forecast is computed by assigning each pixel a positive value if at least $\tau = N$ of the ensemble forecasts detected rain at this pixel, and a negative value otherwise (Figure 7.19).

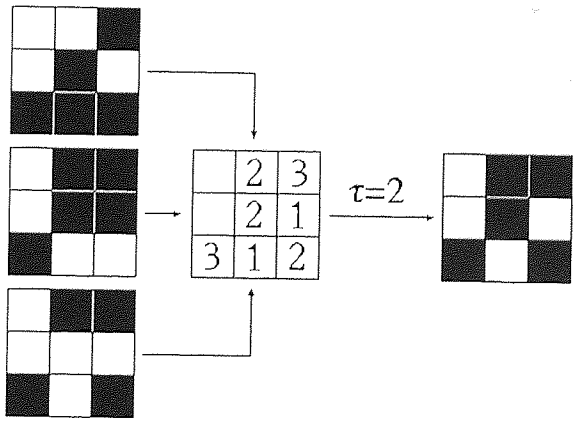


Figure 7.19: Detection of rainy pixels based on ensemble forecast: rainy pixels are selected based on the number of ensemble forecasts detecting them having greater precipitation intensity than τ ($\tau = 2$ in this example)

The composite binary forecast is matched to the observed binary rainfall field to determine the hit rate and false prediction rate. The number τ plays a similar role as in the disease detection example, and the ROC curve for the probabilistic forecast is obtained by having it vary between 0 (all pixels positive) and N+1 (all pixels negative). An example of such a ROC curve is shown on

Figure 7.20. The 3 curves correspond, from top to bottom, to 3 forecast lead times: 30 minutes, 60 minutes and 180 minutes. It is easy to see that the model forecasting skill decreases as the lead time increases, as would be expected.

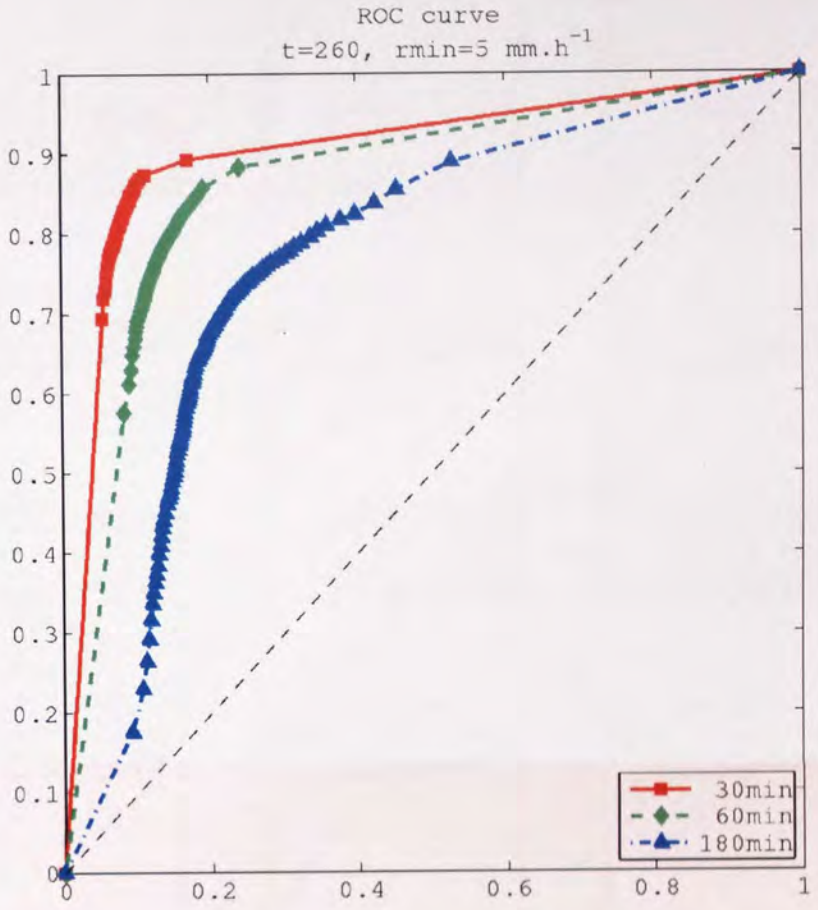


Figure 7.20: Real example of ROC curve based on forecasts for a frontal event (January 2005, t = 260, rain threshold = 5mm.h^{-1})

Of course, a single ROC curve does not tell much about the skill of the model, as it depends strongly on the complexity of the rainfall field and the accuracy of the estimated parameters at the particular instant chosen. In order to obtain a better idea of the overall model's skill, one needs to look at the distribution of such ROC curves over a sufficient number of observations.

To that effect, at each of the 672 15-minute time step, 100 realisations of the stochastic model were generated from the parameters posterior distribution and then propagated forward in time, to provide a Monte Carlo (or large ensemble) estimate of the probability distribution of the precipitation rates over the region at times from t+15 min, t+60 min and t+180 min. 3 precipitation thresholds were considered for rain detection: 1mm.h^{-1} (light rain), 5mm.h^{-1} (medium rain), 10mm.h^{-1} (heavy rain) and the AUC was computed for each.

Figures 7.21 and 7.22 show examples of the ROC curves computed for the convective and frontal event respectively. From top to bottom, ROC curves corresponding to the same 4 consec-

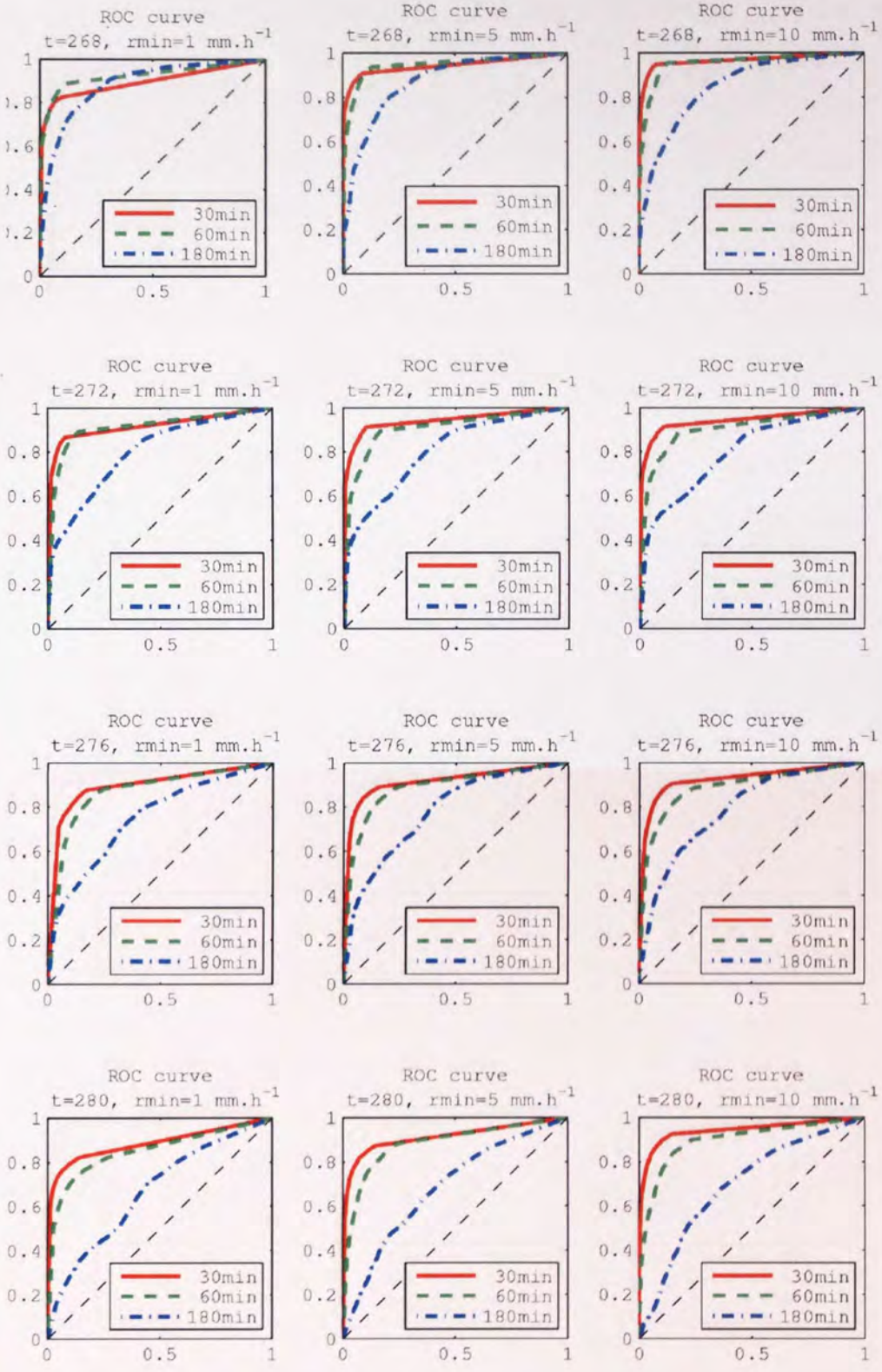


Figure 7.21: ROC curves for convective precipitation forecasts (July 2006). This plot shows, from top to bottom, ROC curves for the same 4 consecutive hourly estimates of the rainfield as in Figure 7.11. Three rain thresholds are considered, from left to right: 1, 5, and 10 mm.h^{-1}. Each plot displays the ROC curve for 3 forecast lead times: 30 min (solid red), 60 min (dashed green) and 180 min (dash-dotted blue).

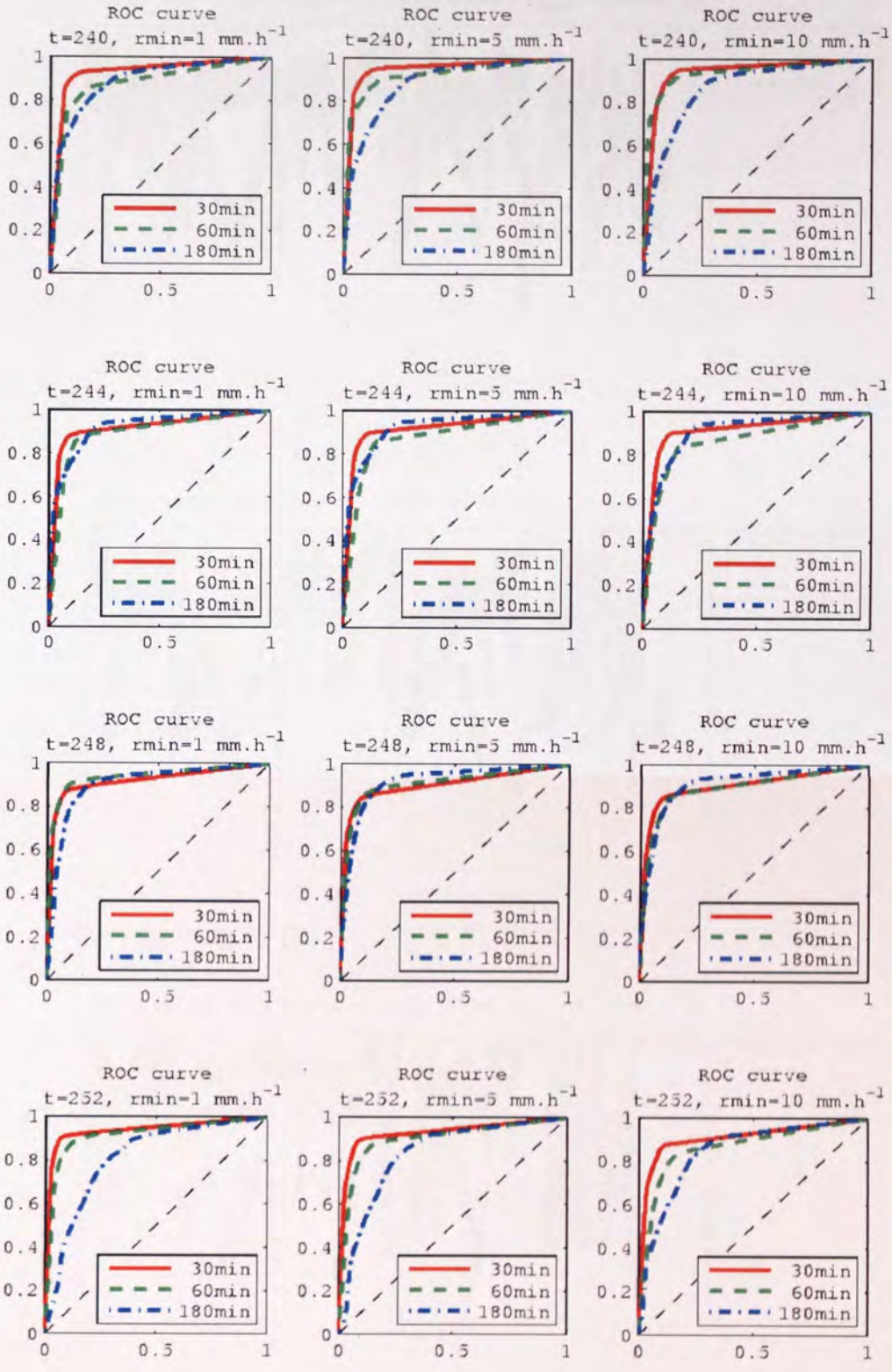


Figure 7.22: ROC curves for frontal precipitation forecasts (January 2005). This plot shows, from top to bottom, ROC curves for the same 4 consecutive hourly estimates of the rainfield as in Figure 7.11. Three rain thresholds are considered, from left to right: 1, 5, and 10 mm.h^{-1}. Each plot displays the ROC curve for 3 forecast lead times: 30 min (solid red), 60 min (dashed green) and 180 min (dash-dotted blue).

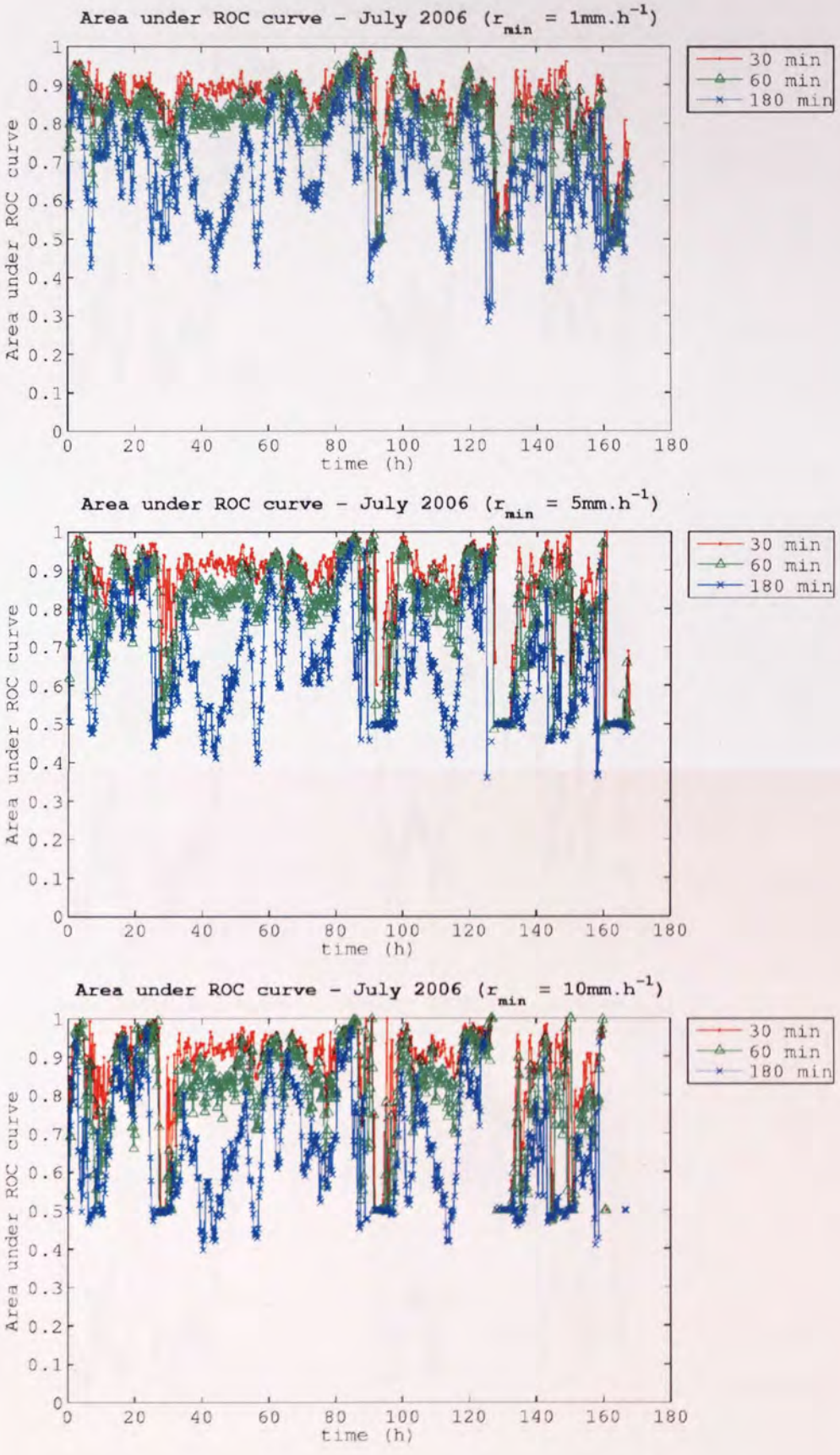


Figure 7.23: Evolution of the area under the ROC curve for a convective event (July 2006)

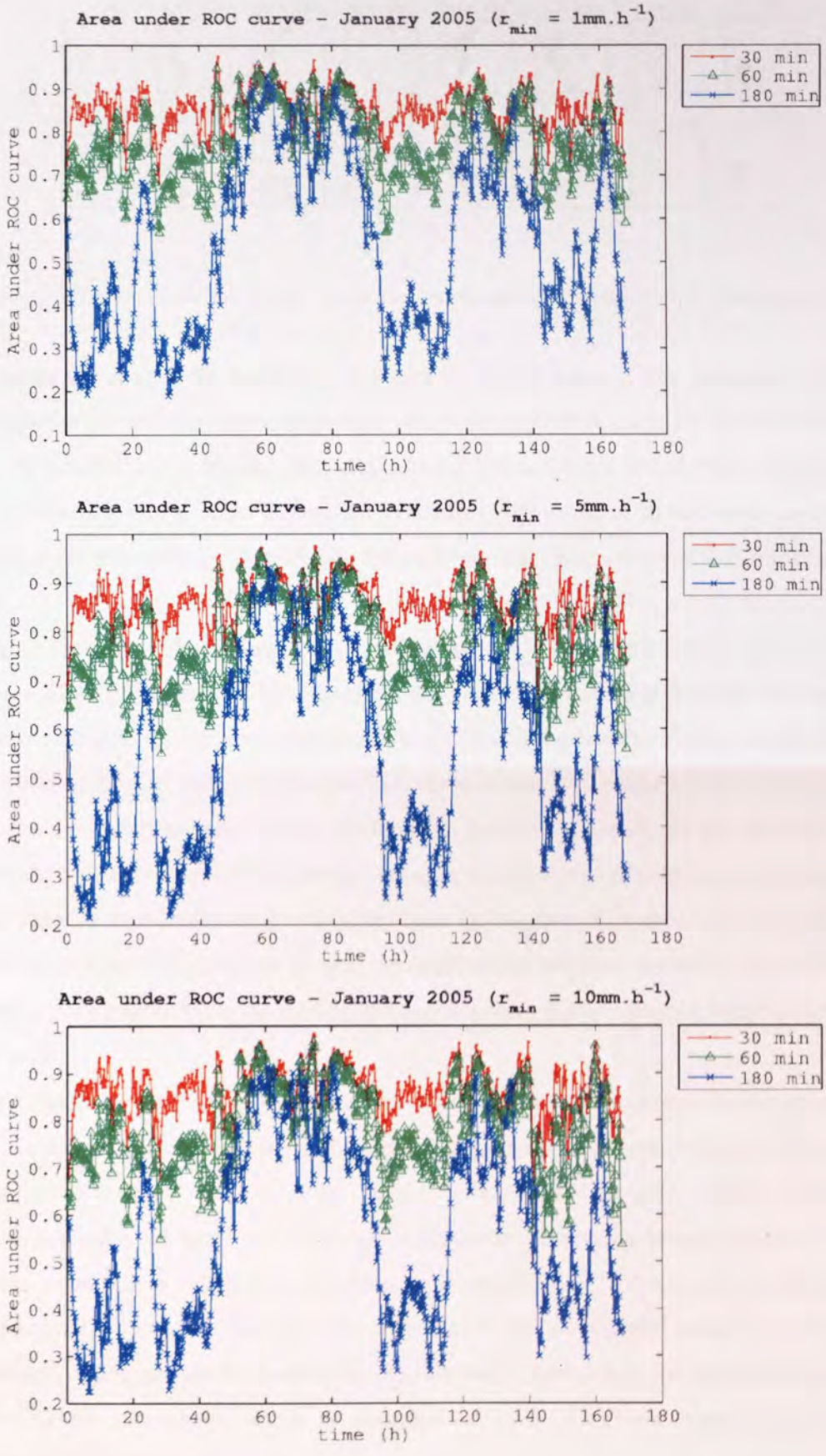


Figure 7.24: Evolution of the area under the ROC curve for a frontal event (January 2005)

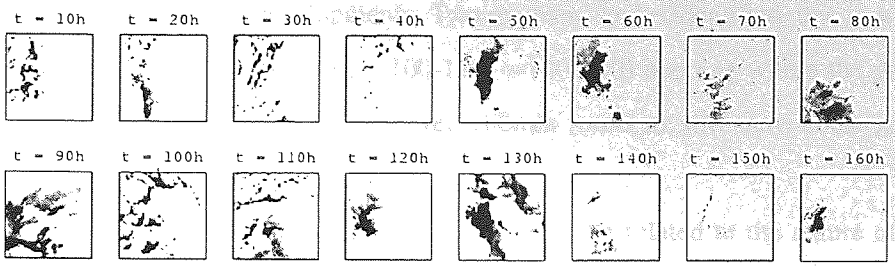


Figure 7.25: Evolution of the rainfall field during the frontal event (January 2005, 10h snapshots).

utive hourly estimates of the rainfield as in Figure 7.11 are displayed. The three rain thresholds considered are plotted from left to right. Each plot displays the ROC curve for the 3 forecast lead times: 30 min (solid red), 60 min (dashed green) and 180 min (dash-dotted blue). In the frontal case, the model is able to retain respectable forecasting skill for up to 3h lead time, as might be expected in the more strongly dynamically forced, larger scale processes typical in frontal precipitation.

Figure 7.23 shows the evolution of the area under the ROC curve for the convective event, computed for the 3 different thresholds (from top to bottom) and three different forecast lead times (for each plot, the top curve corresponds to the 30 min forecast, the middle curve to 60 min and the bottom curve to 180 min). The variability in prediction skill can be related to the nature of the rainfall field which undergoes rapid developments (storms). However, the area under the ROC curve rarely goes below 0.5, even at 180min lead time, which is believed to be a consequence of the simpler structure in the rainfall field (localised storms). However, it is important to mention that the ROC skill score also takes into account the ability of the model to detect dry areas. Because such areas are usually larger in the summer, the model is more likely to perform better with respect to this score.

Note that the absence of rain (or heavy rain) during some periods can result in erroneous ROC curves. If no rain is observed, the Hit Rate cannot be computed (division by zero). This results in missing points on the curve as can be seen on the bottom plot around t = 130h. In the case where no rain cell is left in the model (as a result from a dry observation being assimilated), all of the model's realisations will predict a dry forecast, hence all points will coincide with the bottom left corner ($HR = FAR = 0$), resulting in a curve aligned with the diagonal and an area of exactly 0.5. Several such cases can be observed in the two lower plots, where the detection thresholds is higher. These cases should ideally be discarded as they do not give a correct account of the model's prediction skill.

Figure 7.24 shows similar information to Figure 7.23 for the frontal event. It can be observed that the prediction skill varies more smoothly that in the convective case, due to the larger scale

and slower nature of precipitative developments. Two regimes can be identified, one in which the prediction skill decreases quickly (t=0-40, t=100-120, t=140-160) and one where the prediction skill is retained much longer, to the point that even 180min forecasts still show some good skill (t=60-90, t=120-140).

Qualitative analysis as shown that these two regimes can be related to the nature of the observed rainfall field. Figure 7.25 displays the observed rainfall field at 10h intervals. Heavily clustered fields of high precipitation intensity seem to result in good forecasts while sparse, localised precipitation fronts correspond to poor forecasts. This could be due to the linear forecasting scheme, which is likely to perform better when the rain cells are clustered as this ensures their advection field is smooth. Another possible explanation is the assumption, in advection-based forecasts, that motion is the primary factor of change, and that internal development (growth/decay) can be neglected. It is clear that dissipation phenomena are less noticeable, in proportion, for large areas of intense precipitation than for smaller isolated cells.

Figures 7.26 and 7.27 summarise the distributions, for the whole experiment, of the areas under the ROC curve for the 3 forecast lead times and the 3 precipitation intensity thresholds. As expected, the model skill decreases on average as the forecast lead time increases, with very little skill in any of the forecasts after 3 hours. This is due partly to the simplicity of the precipitation field representation, and partly to the linear nature of the forecast scheme applied. Each cell is advected linearly given the advection at its centre, an assumption which remains relevant only for shorter forecast lead times. At short time scales, particular at t+30 min the model has more skill when forecasting heavy precipitation (>10 $mm.h^{-1}$) than light precipitation (1 $mm.h^{-1}$). This is a useful feature, since for most flood forecasting applications the heavy precipitation is the most important to predict well. This is a common feature of many advection models, since the heavier precipitation tends to be more temporally persistent.

7.3.3 Variogram

The variogram (Cressie, 1992; Marzban, 2007) is a tool commonly used in geostatistics to quantify the structure of a spatial field. The variogram is defined as the variance of the difference in intensity at any two points, as function of the distance (lag) separating these points. Given a two-dimensional field $\mathbf{y}$, if the set of distinct point pairs separated by a lag l is defined as $S(l) = \left\{(\mathbf{s}_i, \mathbf{s}_j)_{i<j} \mid \| \mathbf{s}_i - \mathbf{s}_j \| = l\right\}$ and the difference in intensity at any such two points as $\Delta\mathbf{Y}(l) = \left\{\mathbf{Y}(\mathbf{s}_i) - \mathbf{Y}(\mathbf{s}_j) \mid (\mathbf{s}_i, \mathbf{s}_j) \in S(l)\right\}$, then the variogram can then be expressed a the following function of the lag:

$$2\gamma(\mathbf{s}_i - \mathbf{s}_j) = \mathrm{var}\left[\Delta\mathbf{Y}(l)\right]. \tag{7.7}$$

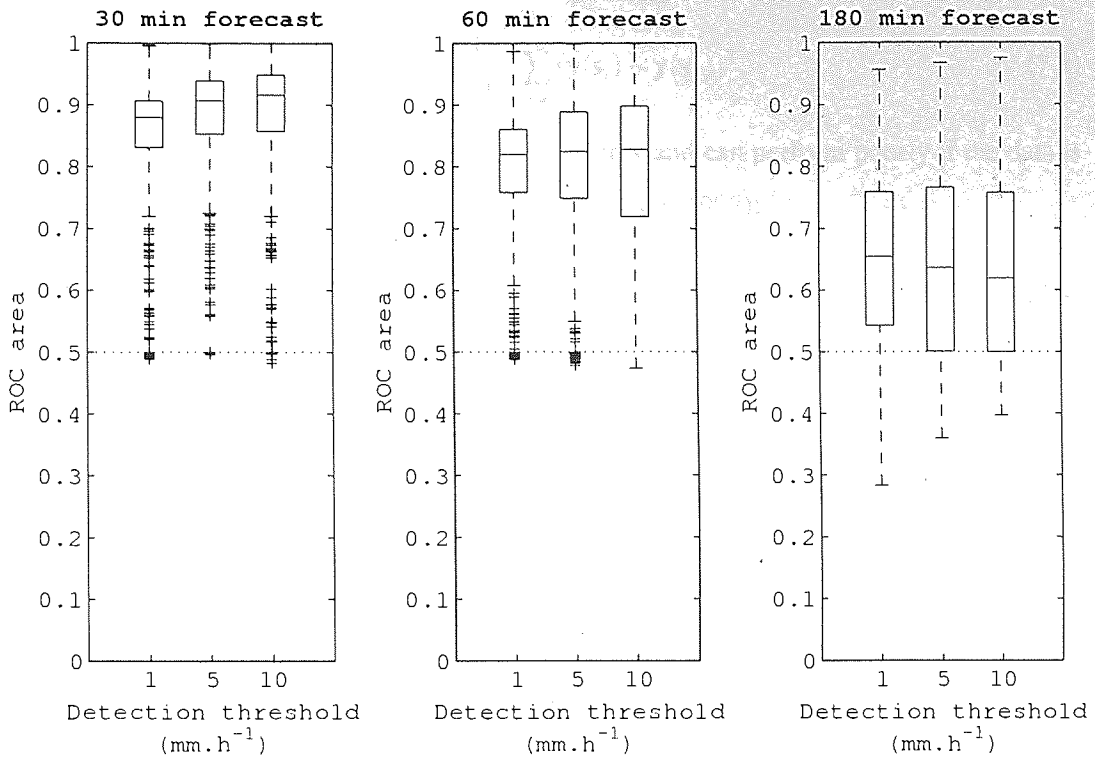


Figure 7.26: Statistics of the area under the ROC curve for a convective event (July 2005)

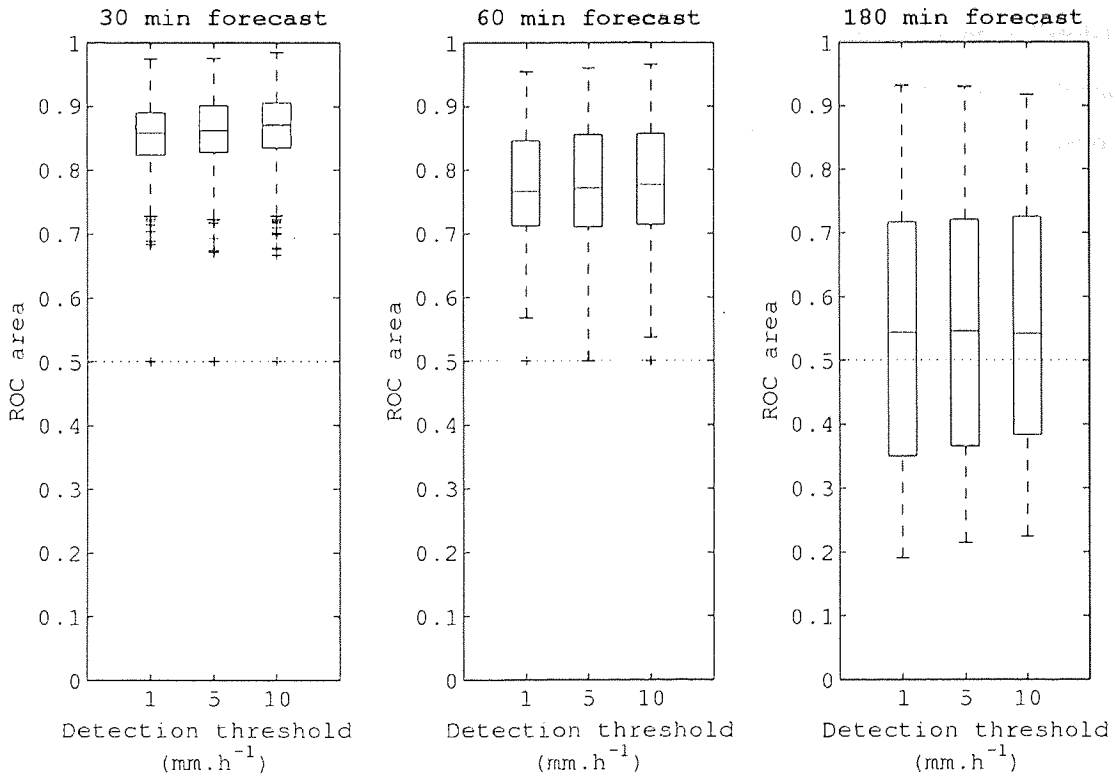


Figure 7.27: Statistics of the area under the ROC curve for a frontal event (January 2005)

An estimator of the variogram for discrete data is given by (Cressie, 1992):

$$2\hat{\gamma}(\mathbf{s}_i - \mathbf{s}_j) = \frac{1}{|S(h)|} \sum_{S(h)} (\mathbf{y}(\mathbf{s}_i) - \mathbf{y}(\mathbf{s}_j))^2. \tag{7.8}$$

However, this estimator is sensitive to large differences in $\mathbf{y}$ and can perform poorly if the data is not consistent. A more robust estimator is provided in Cressie (1992):

$$2\hat{\gamma}(\mathbf{s}_i - \mathbf{s}_j) = \left\{ \frac{1}{|S(h)|} \sum_{S(h)} |\mathbf{y}(\mathbf{s}_i) - \mathbf{y}(\mathbf{s}_j)|^{1/2} \right\}^4 \tag{7.9}$$

and is the one used in this work.

Figure 7.28 shows, on left, the variogram computed, from the top, for the observation and 30min, 60min and 180min forecasts. The corresponding rainfield is plotted on the right. 4 sample realisations are plotted for each forecast. All forecasts are able to retain the spatial structure at small scales (lag < 100), but the larger scale structure is only retained until up to 30min on that example.

7.4 Discussion and future work

This chapter and the previous one presented a new probabilistic data assimilation algorithm which can be applied to nowcasting using a simple advection based precipitation forecasting model. The algorithm has several desirable features, in particular the ability to estimate the posterior distribution of the model state using optimisation methods, which provides control over the computational complexity. While the initial derivation is quite mathematically demanding, the implementation can be employed within any optimisation framework, which forms the basis for most traditional variational assimilation methods.

The new method is extensively tested on two large events characterised by convective and frontal dominated rainfall. The results show the model is robust, and could be applied operationally. The ROC curves show probabilistic skill at all forecast horizons, but it is clear that skill is lost rapidly, which is typical of such advection / extrapolation based systems. Future work should ideally compare the results of the variational Bayes methods with other approaches. This would be greatly facilitated by a suit of standard test cases and diagnostics that could be agreed by the nowcasting community to allow model and method comparison.

There are several areas in which the algorithm could be improved:

- Parallelisation would allow the algorithm to be optimised for more cycles resulting in an optimal KL-divergence based fit, and the number of cells used in the approximation to the precipitation field could also be increased. For instance, the space could be split into smaller sections processed independently. Parts of the minimisation of the KL divergence could

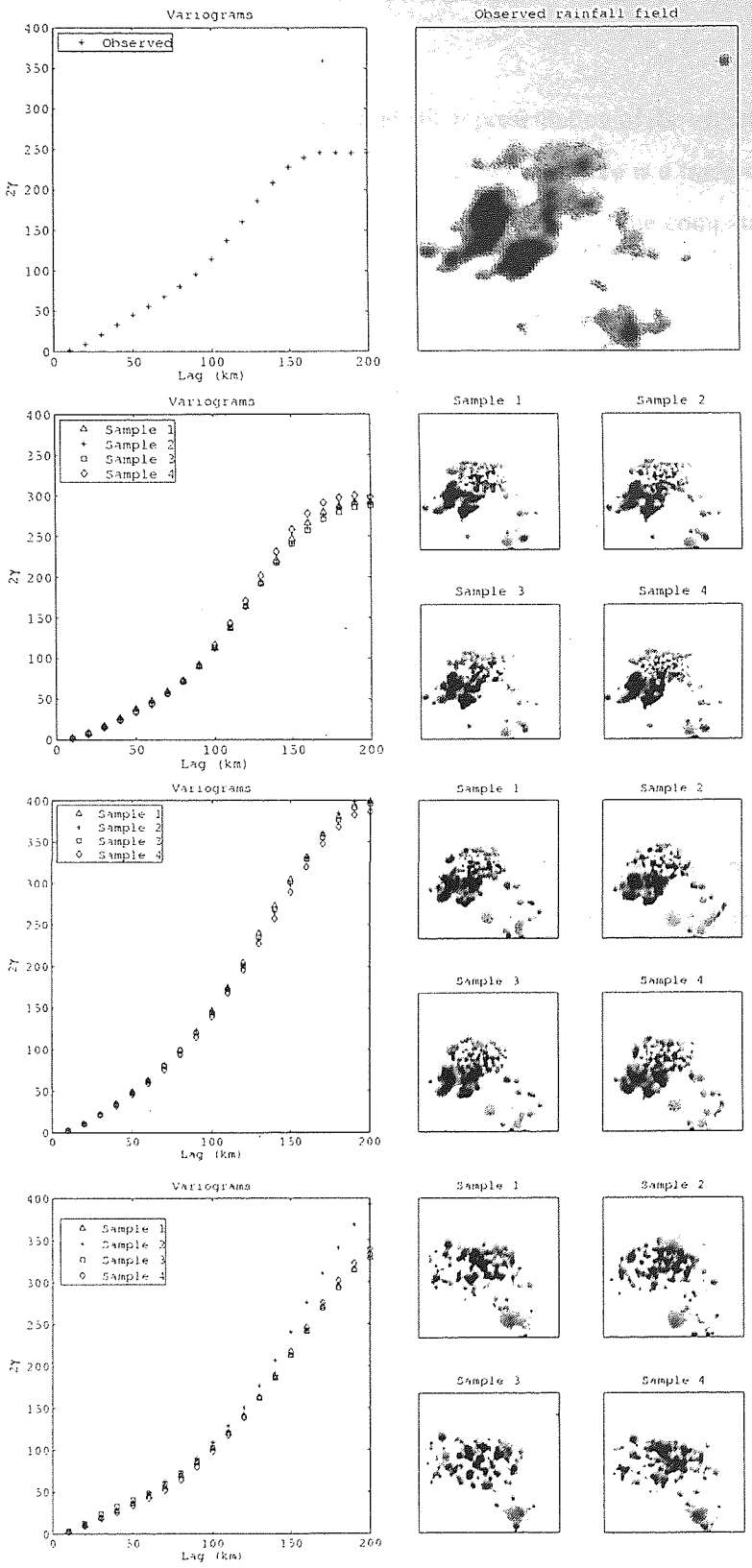


Figure 7.28: Variograms of the observed and forecast precipitation fields (July 2006, t=70h). Variograms and rainfall fields correspond, from top, to observed, 30min, 60min and 180min forecasts. Four forecast realisations are plotted.

also be performed on several processors since cells are assumed uncorrelated factorisation assumption the computation .

- Isotropic Gaussian cells do not provide a realistic representation of the true rain cells, which would often be better approximated by ellipses. However, there is a trade-off between the number of cells needed (fewer in the case of elliptic cells) and the computational costs of the assimilation (heavier in the case of elliptic cells due to the need for a full 2×2 matrix for the width parameter). Ideally, when using elliptic cells, one would also want to estimate the rotational velocity of the cell's axes in addition to the advection of it's centre, which again increases the computational cost.

- The inclusion of an automatic method to select the complexity of the model, using methods similar to those used in the relevance vector machine (Tipping, 2001) would further improve the robustness and scaleability of the algorithm.

- The advection process representation is also rather crude and could be improved with a better representation of the advection field based on a fixed grid and non-linear forecasting methods.

- At the moment, the advection uses the cell's displacement as a "pseudo-observation" in the assimilation step. A further interesting extension to the model is the incorporation of real-time observations of the wind field (froDopplerer sounders) to update the advection field.

- It must also be stated that the model, or propagation, error parameters have been set with reference to other studies and tuning on short data sequences and these are almost certainly not optimised. It would be possible to use the marginal likelihood approximation, derived from the variational Bayes analysis to optimise these parameters as part of the data assimilation method, indeed these could be made adaptive since it is likely that the model error will be state dependent in this application.

More speculatively it would be interesting to attempt to include satellite imagery to track the evolution of the cloud field to better estimate the advection field where precipitation has yet to begin, but where clouds are present. Assimilation of other observation types, includinDopplerer lidar and other more direct measurements of advection would further improve the estimation in the model. Further work could consider a hybrid approach that combines the knowledge of the physical system embodied in high resolution numerical models (Done et al., 2006) but is sufficiently simple to be run on the short assimilation cycles required for short range forecasts. Since the model formulation is probabilistic and the uncertainty represents the model uncertainty rea-

sonably well, Bayesian model averaging could be applied to merge smoothly into a more physics based forecast at longer lead times, so long as the uncertainty on both were well characterised.

8 Conclusions

CONTENTS

8.1 Thesis summary

8.1.1 A comparison of state of the art data assimilation methods

Chapter 2 discussed models and observations as the two key components of a data assimilation system. The existence of error in both models and observations was underlined and its causes discussed. The data assimilation problem was formulated both from a deterministic point of view, where a single best estimate of the state is sought, and from a stochastic point of view, where the uncertainty associated with the best estimate is also quantified (through estimation of the probability distribution of the state).

In Chapter 3, data assimilation methods were first introduced in a static context, where no model is taken into account and the data assimilation problem reduces to estimating the state given observations. It was shown how the problem boiled down to a least squares estimation problem in the case of observations related linearly to the state. 3D VAR, an alternative, variational approach based on minimising the least squares cost function rather than solving it exactly was described. In the case of a non-linear observation operator, it was shown that a sub-optimal solution could be derived provided the operator could be linearised using a Taylor approximation. The least squares estimation method was then derived in a stochastic context, yielding a set of equations for the optimal Gaussian estimate.

Chapter 4 extended the discussion to dynamic data assimilation. It was shown that when one observation is assimilated at a time, dynamic data assimilation was easily derived from the static case. This lead to deterministic methods such as dynamic least squares and dynamic 3D VAR, and stochastic filtering methods such as the Kalman Filter. Several extensions of the Kalman Filter for non-linear models and non-Gaussian distributions were discussed: the Extended Kalman Filter uses a linear approximation to the model in the computation of the Kalman Filter prediction equations; the Ensemble Kalman Filter propagates an ensemble of state realisations from which the first moments of the (Gaussian) state's distribution are approximated, each ensemble member being updated using the Kalman Filter update equation; the Particle Filter uses a full Monte-Carlo approach which can be applied to non-linear models and non-Gaussian distributions.

In the case where several sequential observations are used in the assimilation, dynamic data assimilation leads to the deterministic 4D VAR method (the standard for weather forecasting in the UK and France) and to stochastic smoothing methods, which were only briefly mentioned in this work.

Two experiments were then set up in order to compare these data assimilation methods and try and determine their strengths and weaknesses. The methods were run on two non-linear systems often used in the atmospheric science literature: the Lorenz 63 (low dimension) and the Lorenz 96

systems (medium dimension). The effects of dimension, non-linearity and method's parameters were discussed for each method. The 4D VAR algorithm was shown to outperform the other methods when the effects of non-linearity remain limited, provided a sufficiently long time-window was considered. The Particle Filter showed some very good skill in low dimensions, and seemed to be, of all methods, the more robust to non-linearity. However, it was also demonstrated that the Particle Filter suffers from filter divergence issues in higher dimensions and would need an unrealistic number of particles to achieve satisfactory results. The Ensemble Kalman Filter was able to provide a good assimilation in linear and non-linear regimes, at low and high dimensions, while keeping computation time to a minimum. The limitations of 4D VAR and the Extended Kalman Filter, which both rely on linearisation approximations, became apparent in a strongly non-linear regime.

8.1.2 Application to precipitation forecasting

Moving from the general to the particular, a new data assimilation method is then developed and applied to the problem of very short-term precipitation forecasting. Chapter 6 presents a new probabilistic data assimilation algorithm which can be applied to nowcasting using a simple advection based precipitation forecasting model. The algorithm has several desirable features, in particular the ability to estimate the posterior distribution of the model state using optimisation methods, which provides control over the computational complexity. While the initial derivation is quite mathematically demanding, the implementation can be employed within any optimisation framework, which forms the basis for most traditional variational assimilation methods.

In Chapter 7, the new method is tested on two large events characterised by convective and frontal dominated rainfall. The results show the model is robust, although further testing is necessary to assess its applicability to operational nowcasting. The ROC curves show probabilistic skill at all forecast horizons, but it is clear that skill is lost rapidly, which is typical of such advection / extrapolation based systems.

8.2 Directions for future research

8.2.1 Towards a benchmark for data assimilation methods

The comparison of data assimilation methods which has been undertaken in this thesis is a first step towards a unified benchmark to be used both as a reference and testbed for new data assimilation methods. Much work remains to be done to achieve this objective. Some the directions in which this work could be taken forward are listed below:

- **Further methods** – The presented work only focused on 4 key data assimilation methods: the Extended Kalman Filter, the Ensemble Kalman Filter, the Particle Filter and 4D VAR (strong and weak constraint). Several methods have been omitted, which ought to be added for the sake of completeness. Such methods would include, for instance: 3D VAR, the Unscented Kalman Filter and path sampling techniques. Variations on a given method should also be taken into account, e.g. Particle Filter with different resampling schemes or various implementations of the Ensemble Kalman Filter.

- **Further models** – Two classical non-linear deterministic models were considered. The addition of more models would certainly help provide a better understanding of each methods abilities in different situations. In particular, stochastic models like the double-well model would provide a new dimension to the test bench (so far, all models considered were deterministic).

- **Imperfect model setting** – The assimilation methods were tested in a perfect model setting, which does not provide a true account of the method's capabilities when the model is unknown, as is the case in almost all real applications. Further experiments are thus needed in an imperfect model setting.

- **Error models** – Model and observation errors are very often assumed Gaussian to keep the computations simple. However, this assumption is often unrealistic. Better, more realistic error models should be investigated, with the aim to move towards on-line estimation of error parameters (the error being estimated directly from the model and the data).

- **Computational aspects** – Although often considered less important, the computational aspects of data assimilation still need to be taken into account. In particular in the case of short-term forecasting, computation time becomes critical as data assimilation needs to be completed witihn strong time constraints. The application of dimensionality reduction techniques and the emulator setting to data assimilation could lead to new assimilation methods and reduce the limitations of existing methods.

8.2.2 Precipitation nowcasting model

Several extensions to the precipitation nowcasting model presented in Chapters 6 and 7 have been mentioned already. They include:

- **Improved forecasting** – A better forecasting scheme in which the advection of the cells is computed based on their forecast location rather than being assumed constant.

- **Other cell shapes** – Other shapes could be considered for the rain cell. For instance, elliptic Gaussian cells or plateau-shaped cells might lead to a better representation of the rainfall field. The question of the computational burden would however arise as more parameters would possibly be needed. It is assumed that by allowing more flexibility in the rain cells, fewer cells would be needed to capture the rainfall field.

- **Scale decomposition** – At the moment, the model treats the rainfall field as a whole. However, it is known than 2D precipitation fields are a projection of precipitation occurring at different scales in the atmosphere. Estimating precipitation at different spatial scales is likely to lead to an improved and more flexible estimate of the advection field.

- **Parallelisation of the algorithm** could improve computational speed significantly. With the current assumptions in the update step that cells are conditionally uncorrelated, the cost function can be factorised. This factorisation could easily be exploited to run the optimisation on several concurrent processors.

- **Comparison with existing methods** – Future work should ideally compare the results of the variational Bayes methods with other approaches. This would be greatly facilitated by a suit of standard test cases and diagnostics that could be agreed up by the nowcasting community to allow model and method comparison.

Bibliography

PP Alberoni, V. Ducrocq, G. Gregoric, G. Haase, I. Holleman, M. Lindskog, B. Macpherson, M. Nuret, and A. Rossa. Quality and assimilation of radar data for nwp - a review. Technical report, COST-717, 2003.

E. Andersson, J. Haseler, P. Unden, P. Courtier, G. Kelly, D. Vasiljevic, C. Brankovic, C. Cardsali, C. Gaffard, and A. Hollingsworth. The ECMWF implementation of three-dimensional variational assimilation (3D-Var). III: Experimental results. *Quarterly Journal of the Royal Meteorological Society*, 124:1831–1860, 1998.

A. Andrews. A square root formulation of the Kalman covariance equations. *AIAA Journal*, 6: 1165–1168, 1968.

A. Apte, M. Hairer, AM Stuart, and J. Voss. Sampling the posterior: An approach to non-Gaussian data assimilation. *Physica D: Nonlinear Phenomena*, 230(1-2):50–64, 2007.

A. Apte, CK Jones, AM Stuart, and J. Voss. Data assimilation: Mathematical and statistical perspectives. *International Journal for Numerical Methods in Fluids*, 56(8):1033, 2008.

S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2): 174–188, February 2002.

F. Atger. Verification of intense precipitation forecasts from single models and ensemble prediction systems. *Nonlinear Processes in Geophysics*, 8:401–417, 2001.

R. Barillec and D. Cornford. Implementing a Particle Filter with the Lorenz 96 system. Annual Meeting of the Royal Meteorological Society (oral presentation), 2005.

R. Barillec and D. Cornford. An empirical comparison of the effect of model non-linearity on state of the art data assimilation methods. *Geophysical Research Abstracts*, 8:02830, 2006.

R. Barillec and D. Cornford. Data assimilation for precipitation nowcasting using Bayesian inference. *Advances in Weather Resources (special issue)*, 2008a. (revised paper submitted).

R. Barillec and D. Cornford. Precipitation Nowcasting using Bayesian Inference. In *Conference on Weather Radar and Hydrology*, March 2008b.

R. G. Barry and R. J. Chorley. *Atmosphere, weather and climate*. Routledge, 2003.

N. Bergman, Dept. of Electrical Engineering, and U. i Linköping. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Univ., 1999.

J. Bernardo and A. Smith. *Bayesian theory*. Wiley series in probability and statistics. 1994.

A. Berne, G. Delrieux, J.-D. Creutin, and C. Obled. Temporal and spatial resolution of rainfall measurements required for urban hydrology. *Journal of Hydrology*, 299(3-4):166–179, 2004. doi: 10.1016/j.jhydrol.2004.08.002.

G.J. Bierman. *Factorization Methods for Discrete Sequential Estimation*, volume 128 of *Mathematics in Science and Engineering*. 1977.

C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

D. Bocchiola and R. Rosso. The use of scale recursive estimation for short term quantitative precipitation forecast. *Physics and Chemistry of the Earth*, 31(18):1228–1239, 2006.

F. Bouttier and P. Courtier. *Data assimilation concepts and methods*. European Centre for medium-range Weather Forecasting, ecmwf lecture notes edition, March 1999.

N.E. Bowler, C.E. Pierce, and A.W. Seed. STEPS: a probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled NWP. Quart. *Quarterly Journal of the Royal Meteorological Society*, 132:2127–2155, 2006.

K.A. Browning, A.M. Blyth, P.A. Clark, U. Corsmeier, C.J. Morcrette, J.L. Agnew, S.P. Ballard, D. Bamber, C. Barthlott, L.J. Bennett, et al. The Convective Storm Initiation Project. *Bulletin of the American Meteorological Society*, 88(12):1939–1955, 2007.

M. Buehner. Ensemble-derived stationary and flow-dependent background-error covariances: Evaluation in a quasi-operational NWP setting. *Quarterly Journal of the Royal Meteorological Society*, 131(607):1013–1043, 2005.

M. Buehner, P. Gauthier, and Z. Liu. Evaluation of new estimates of background-and observation-error covariances for variational assimilation. *Quarterly Journal of the Royal Meteorological Society*, 131(613):3373, 2005.

R. Buizza, A. Hollingsworth, F. Lalaurette, and A. Ghelli. Probabilistic predictions of precipitation using the ecmwf ensemble prediction system. *Weather and Forecasting*, 14:168–189, 1999.

G. Burgers, P. J. van Leeuwen, and G. Evensen. On the analysis scheme in the ensemble kalman filter. *Monthly Weather Review*, 126:1719–1724, 1998.

A. Burton and PE O'Connell. Model based algorithms for short lead time nowcasting. Technical Report 5.3, MUSIC project report, 2003.

B. Casati, L. J. Wilson, D. B. Stephenson, P. Nurmi, A. Ghelli, M. Pocernich, U. Damrath, E. E. Ebert, B. G. Brown, and S. Mason. Forecast verification: current status and future directions. *Meteorological Applications*, (15), 2008.

S.E. Cohn. An introduction to estimation theory. *J. Meteor. Soc. Japan*, 75:257–288, 1997.

V. Collinge. *The development of weather radar in the UK*. Wiley, 1987.

D. Cornford. Random field models and priors on wind. Technical Report NCRG/97/023, Neural Computing Research Group, Aston University, Birmingham, UK, March 1998a.

D. Cornford. Flexible gaussian process wind field models. Technical Report NCRG/98/017, Neural Computing Research Group, Aston University, Birmingham, UK, August 1998b.

D. Cornford. A bayesian state space modelling approach to probabilistic quantitative prediction forecasting. *Journal of Hydrology*, 288(1-2):92–104, 2004. doi: 10.1016/j.jhydrol.2003.11.040.

D. Cornford, I.T. Nabney, and C.K.I. Williams. Modelling Frontal Discontinuities in Wind Fields. *Nonparametric Statistics*, 14, 2002.

J. CotÃl', S. Gravel, A. MÃl'thot, A. Patoine, M. Roch, and A. Staniforth. The Operational CMC–MRB Global Environmental Multiscale (GEM) Model. Part I: Design Considerations and Formulation. *Monthly Weather Review*, 126(6):1373–1395, 1998.

P Courtier, J N Thepaut, and A Hollingsworth. A strategy for operational implementation of 4D-VAR, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120:1367–1387, 1994.

P. Courtier, E. Andersson, W. Heckley, J. Pailleux, D. Vasiljevic, M. Hamrud, A. Hollingsworth, F. Rabier, and M. Fisher. The ECMWF implementation of three-dimensional variational assimilation (3D-Var). I: Formulation. *Quarterly Journal of the Royal Meteorological Society*, 124 (550):1783–1807, 1998.

PSP Cowpertwait, PE O'Connell, AV Metcalfe, and JA Mawdsley. Stochastic point process modelling of rainfall. I. Single-site fitting and validation. *Journal of Hydrology*, 175(1-4):17–46, 1996.

PSP Cowpertwait, V. Isham, and C. Onof. Point process models of rainfall: Developments for fine-scale structure. Technical Report 277, University College London, Department of Statistical Science, June 2007.

D. Cox and V. Isham. A simple spatial-temporal model of rainfall. *Proceedings of the Royal Society of London*, A415:317–328, 1988.

J.M. Cram and M.L. Kaplan. Variational Assimilation of VAS Data into a Mesoscale Model; Assimilation Method and Sensitivity Experiments. *Monthly Weather Review*, 113(4):467–484, 1985.

N. Cressie. Statistics for Spatial Data. *Terra Nova*, 4(5):613–617, 1992.

R. Daley. *Atmospheric Data Analysis*, chapter 5, pages 155–163. Cambridge University Press, 1991.

S L Dance. Issues in high resolution limited area data assimilation for quantitative precipitation forecasting. *Physica D*, 196:1–27, 2004.

D. Dee and G. Gaspari. Development of anisotropic correlation models for atmospheric data assimilation. *Preprints, 11th Conf. on Numerical Weather Prediction, Norfolk, VA, Amer. Meteor. Soc*, pages 249–251, 1996.

R. Deidda. Rainfall downscaling in a space-time multifractal framework. *Water Resources Research*, 36(7):1779–1794, 2000.

G. Desroziers. A Coordinate Change for Data Assimilation in Spherical Geometry of Frontal Structures. *Monthly Weather Review*, 125(11):3030–3038, 1997.

M. Dixon and G. Wiener. Titan: Thunderstorm identification, tracking, analysis, and nowcasting - a radar-based methodology. *Journal of Atmospheric and Oceanic Technology*, 10(6):785–797, 1993.

J. M. Done, P. A. Clark, G. C. Craig, M. E. B. Gray, and S. L. Gray. Mesoscale simulations of organised convection: Importance of convective-equilibrium. *Quarterly Journal of the Royal Meteorological Society*, 132:737–756, 2006.

R. Douc, O. Cappe, and E. Moulines. Comparison of Resampling Schemes for Particle Filtering. In *Proceedings of the 4th International Symposium onImage and Signal Processing and Analysis (ISPA 2005)*, pages 64–69, 2005.

A. Doucet. On Sequential Simulation-Based Methods for Bayesian Filtering. Technical Report CUED/F-INFENG/TR.310, Cambridge University Department of Engineering, 1998.

A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, 2001.

T. Einfalt, Karsten Arnbjerg-Nielsen, Claudia Golz, Niels-Einar Jensen, Markus Quirmbach, Guido Vaes, and Baxter Vieux. Towards a roadmap for use of radar rainfall data in urban drainage. *Journal of Hydrology*, 299(3-4):186–202, 2004. doi: 10.1016/j.jhydrol.2004.08.004.

J. Evans. *The History and Practice of Ancient Astronomy*. Oxford University Press, USA, 1998.

G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5):10143–10162, 1994.

G. Evensen. Advanced Data Assimilation for Strongly Nonlinear Dynamics. *Monthly Weather Review*, 125(6):1342–1354, 1997.

G. Evensen. *Data assimilation: The Ensemble Kalman Filter*. Springer-Verlag Berlin, 2007.

Geir Evensen. Using the Extended Kalman Filter with a multilayer quasi geostrophicmodel. *J. Geophys. Res.*, 97:17,905–17924, 1992.

Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, (27):861–874, 2005.

E.J. Fertig, J. Harlim, and B.R. Hunt. A comparative study of 4D-VAR and a 4D Ensemble Kalman Filter: perfect model simulations with Lorenz-96. *Tellus A*, 59(1):96–100, 2007.

M. Fisher. Background error covariance modelling. Seminar on Recent Development in Data Assimilation for Atmosphere and Ocean, pages 45–63. ECMWF, 2003.

M. Fisher. "Wavelet" J_b – A new way to model the statistics of background errors. In *ECMWF Newsletter*, volume Winter 2005/2006, pages 23–28. ECMWF, 2006.

M. Fisher, M. Leutbecher, and GA Kelly. On the equivalence between Kalman smoothing and weak-constraint four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 131:3235–3246, 2005.

R.J. Fitzgerald. Divergence of the Kalman Filter. *IEEE Transactions on Automatic Control*, 16 (6):688–706, December 1971.

B. D. Gammel. Matpack C++ Numerics and Graphics Library. Available from: `http://www.matpack.de/`, 2005.

P. Gauthier, C. Charette, L. Fillion, P. Koclas, and S. Laroche. Implementation of a 3D variational data assimilation system at the Canadian Meteorological Centre. *Atmosphere-Ocean*, 37:103–156, 1999.

U. Germann and I. Zawadzki. Scale-Dependence of the Predictability of Precipitation from Continental Radar Images. Part I: Description of the Methodology. *Monthly Weather Review*, 130 (12):2859–2873, 2002. doi: 10.1175/1520-0493(2002)130<2859:SDOTPO>2.0.CO;2.

M. Gibson. Report on further European radar data for use in Nimrod. Technical report, Forecasting Research Technical Report, 2001.

B. Golding. Nimrod: A system for generating automated very short range forecasts. *Meteorological Applications*, 5(1):1–16, 1998. doi: 10.1017/S1350482798000577.

B. Golding. Quantitative precipitation forecasting in the uk. *Journal of Hydrology*, 239:286–305, 2000. doi: 10.1016/S0022-1694(00)00354-1.

C. Golz, T. Einfalt, M. Gabella, and U. Germann. Quality control algorithms for rainfall measurements. *Atmospheric Research*, 77(1-4):247–255, 2005. doi: 10.1016/j.atmosres.2004.10.027.

R. C. Gonzalez and R. E. Woods. *Digital Image Processing*, chapter 3, pages 145–155. Pearson Prentice Hall, 3 edition, 2008.

NJ Gordon, DJ Salmond, and AFM Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Proceedings of the IEEE on Radar and Signal Processing (part F)*, 140(2): 107–113, 1993.

M. Grecu and W. F. Krajewski. A large-sample investigation of statistical procedures for radar-based short-term quantitative precipitation forecasting. *Journal of hydrology*, 239:69–84, 2000. doi: 10.1016/S0022-1694(00)00360-7.

V.K. Gupta and E. Waymire. Multiscaling properties of spatial rainfall and river flow distributions. *Journal of Geophysical Research*, 95(D3):1999–2009, 1990.

N. Gustafsson. Use of a digital filter as weak constraint in variational data assimilation. In *Proc. ECMWF Workshop on Variational Assimilation with Special Emphasis on Three-dimensional Aspects*, pages 327–338, Proceedings of the ECMWF Workshop on Variational Assimilation-Shineld Park, Reading, Berks. RG2 9AX, UK, 1992. European Centre for Medium Range Weather Forecasting.

M. Hairer, AM Stuart, and J. Voss. A Bayesian approach to data assimilation. *Physica D*, 2005.

T.M. Hamill and C. Snyder. A Hybrid Ensemble Kalman Filter–3D Variational Analysis Scheme. *Monthly Weather Review*, 128(8):2905–2919, 2000.

WH Hand and BJ Conway. An Object-Oriented Approach to Nowcasting Showers. *Weather and Forecasting*, 10(2):327–341, 1995.

J. Harlim and B.R. Hunt. A non-Gaussian Ensemble Filter for Assimilating Infrequent Noisy Observations. *Tellus A*, 59(2):225–237, 2007.

L. O. Harvey, K. R. Hammond, C. M. Lusk, and E. F. Mross. The application of signal detection theory to weather forecasting. *Monthly Weather Review*, May 1992.

G. Hinton and D. van Camp. Keeping neural networks simple by minimizing the description length of the weights. *Proceedings of the sixth annual conference on computational learning theory*, pages 5–13, 1993.

J.D. Hol, T.B. Schon, and F. Gustafsson. On Resampling Algorithms for Particle Filters. *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 79–82, 2006.

E.V. Holm. *Lecture notes on assimilation algorithm*. European Centre for medium-range Weather Forecasting, ecmwf lecture notes edition, June 2003.

P. L. Houtekamer and H. L. Mitchell. Data assimilation using an ensemble kalman filter technique. *Monthly Weather Review*, 126:796–811, 1998.

K. Ide, P. Courtier, M. Ghil, and A. C. Lorenc. Unified notation for data assimilation: Operational, sequential and variational. *Journal of the Meteorological Society of Japan*, 75:181–189, 1997.

A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. New York : Academic Press, 1970.

T. Jennings. *Atmosphere and weather*. Smart Apple Media, 2005.

SJ Julier. The scaled unscented transformation. *American Control Conference, 2002. Proceedings of the 2002*, 6, 2002.

SJ Julier, JK Uhlmann, and HF Durrant-Whyte. A new approach for filtering nonlinear systems. *American Control Conference, 1995. Proceedings of the*, 3, 1995.

SJ Julier, JK Uhlmann, I. Ind, and MO Jefferson City. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

E. Kalnay. *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, 2003.

E. Kalnay, H. Li, T. Miyoshi, S.H.U.C. Yang, and J. Ballabrera-Poy. 4-D-Var or ensemble Kalman filter? *Tellus A*, 59(5):758–773, 2007.

P. Kaminski, A. Bryson Jr, and S. Schmidt. Discrete square root filtering: A survey of current techniques. *Automatic Control, IEEE Transactions on*, 16(6):727–736, 1971.

RJ Keeler and SM Ellis. Observational error covariance matrices for radar data assimilation. *Physics and Chemistry of the Earth, Part B*, 25(10-12):1277–1280, 2000.

G. Kitagawa. Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5:1–25, 1996.

E. Klinker, F. Rabier, G. Kelly, and JF Mahfouf. The ECMWF operational implementation of four-dimensional variational assimilation. III: Experimental results and diagnostics with operational configuration. *Quarterly Journal of the Royal Meteorological Society*, 126(564):1191–1215, 2000.

JH Kotecha and PM Djuric. Gaussian particle filtering. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 51(10):2592–2601, 2003a.

JH Kotecha and PM Djuric. Gaussian sum particle filtering. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 51(10):2602–2612, 2003b.

D. Koutsoyiannis and C. Onof. Rainfall disaggregation using adjusting procedures on a Poisson cluster model. *Journal of Hydrology*, 246(1-4):109–122, 2001.

R. Krzysztofowicz. The case for probabilistic forecasting in hydrology. *Journal of Hydrology*, 249:2–9, 2001.

R.J. Kuligowski and A.P. Barros. Experiments in Short-Term Precipitation Forecasting Using Artificial Neural Networks. *Monthly Weather Review*, 126(2):470–482, 1998.

S. Laroche, P. Gauthier, J. St-James, and J. Morneau. Implementation of a 3D Variational Data Assmilation System at the Canadian Meteorological Centre. Part II: The Regional Analysis. *Atmosphere-Ocean*, 37(3):281–307, 1999.

F.X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A*, 38, 1986.

H.W. Lean, P.A. Clark, M. Dixon, N.M. Roberts, A. Fitch, R. Forbes, and C. Halliwell. Characteristics of High Resolution Versions of the Met Office Unified Model for Forecasting Convection Over the UK. *Monthly Weather Review*, preprint, 2008.

L. LeCam. A stochastic description of precipitation. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, 3:165–186, 1961.

JM Lewis and JC Derber. The use of adjoint equations to solve a variational adjustment problem with advective constraints. *Tellus A*, 37, 1985.

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

L. Li, W. Schmid, and J. Joss. Nowcasting of Motion and Growth of Precipitation with Radar over a Complex Orography. *Journal of Applied Meteorology*, 34(6):1286–1300, 1995.

Z. Li and IM Navon. Optimality of variational data assimilation and its relationship with the Kalman filter and smoother. *Quarterly Journal of the Royal Meteorological Society*, 127(572): 661–683, 2001.

C. Lin, S. Vasic, I. Zawadzki, and B. Turner. Precipitation Forecast Based on Numerical Weather Prediction Models and Radar Nowcasts. *Proceedings of ERAD*, 201(205), 2004.

M. Lindskog. *On errors in meteorological data assimilation.* PhD thesis, Department of Meteorology, Stockholm University, 2007.

H. Liu, V. Chandrasekar, and G. Xu. An Adaptive Neural Network Scheme for Radar Rainfall Estimation from WSR-88D Observations. *Journal of Applied Meteorology*, 40(11):2038–2050, 2001.

J.S. Liu and R. Chen. Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.

A. C. Lorenc. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112:1177–1194, 1986.

AC Lorenc, SP Ballard, RS Bell, NB Ingleby, PLF Andrews, DM Barker, JR Bray, AM Clayton, T. Dalby, D. Li, et al. The Met. Office global three-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 126(570):2991–3012, 2000.

Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Science*, 20: 130–141, 1963.

Edward N. Lorenz. Predictability – A problem partly solved. *Proc. seminar on predictability*, 1996.

Edward N. Lorenz and Kerry A. Emanuel. Optimal sites for supplementary weather observations: simulation with a small model. *Journal of the Atmospheric Science*, 55:399–414, 1998.

S. Lovejoy and BB Mandelbrot. Fractal Properties of Rain, and a Fractal Model. *Tellus*, 37, 1985.

S. Lovejoy and D. Schertzer. Multifractals and rain. In *New Uncertainty Concepts in Hydrology and Hydrological Modeling*, pages 61–103. Cambridge Press, 1995.

JF Mahfouf and F. Rabier. The ECMWF operational implementation of four-dimensional variational assimilation. II: Experimental results with improved physics. *Quarterly Journal of the Royal Meteorological Society*, 126(564):1171–1190, 2000.

H.R. Maier and G.C. Dandy. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling and Software*, 15(1):101–124, 2000.

C. Marzban. Variograms as a verification tool. Submitted to Mon. Wea. Rev., 2007.

P. S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. 1979.

P. Meischner. *Weather radar: principles and advanced applications*. Berlin; New York: Springer, 2004.

R.N. Miller, M. Ghil, and F. Gauthiez. Advanced Data Assimilation in Strongly Nonlinear Dynamical Systems. *Journal of the Atmospheric Sciences*, 51(8):1037–1056, 1994.

R. Moore, V. Bell, and D. Jones. Forecasting for flood warning. *Comptes Rendus Geoscience*, 337:203–217, 2005. doi: 10.1016/j.crte.2004.10.017.

C. Mueller, T. Saxen, R. Roberts, J. Wilson, T. Betancourt, S. Dettling, N. Oien, and J. Yee. Ncar auto-nowcast system. *Weather and Forecasting*, 18(4):545–561, 2003.

I. Nabney. *NETLAB: Algorithms for Pattern Recognition*. Springer-Verlag, 2001.

P. Northrop. A clustered spatial-temporal model of rainfall. *Proceedings of the Royal Society of London*, A454:1875–1888, 1997.

P. Nurmi. Recommendation on the verification of local weather forecasts. ECMWF Technical Memorandum 430, ECMWF Operations Department, October 2003.

C. Onof and H.S. Wheater. Modelling of British rainfall using a random parameter Bartlett-Lewis rectangular pulse model. *Journal of hydrology(Amsterdam)*, 149(1-4):67–95, 1993.

D. Orrell. Model error and predictabiliy over different timescales in the Lorenz '96 systems. *Journal of the Atmospheric Science*, 60:2219–2228, 2003.

D.F. Parrish and J.C. Derber. The National Meteorological Center's Spectral Statistical-Interpolation Analysis System. *Monthly Weather Review*, 120(8):1747–1763, 1992.

W. D. Penny. KL divergences of Normal, Gamma, Dirichlet and Wishart densities. Technical report, Wellcome Department of Cognitive Neurology, 2001.

K. B. Petersen and M. S. Pedersen. The matrix cookbook, feb 2006. Version 20051003.

A. Pfister and A. Cassar. Use and benefit of radar rainfall data in an urban real time control project. *Physics and Chemistry of the Earth, Part B*, 24(8):903–908, 1999.

D.T. Pham. Stochastic Methods for Sequential Data Assimilation in Strongly Nonlinear Systems. *Monthly Weather Review*, 129(5):1194–1207, 2001.

C. Pierce, P. Hardaker, C. Collier, and C. Hagget. Gandolf: a system for generating automated nowcasts of convective precipitation. *Meteorological Applications*, 7(4):341–360, 2000.

C. Pierce, E. Ebert, A. Seed, M. Sleigh, C. Collier, N. Fox, N. Donaldson, J. Wilson, R. Roberts, and K. Mueller. The nowcasting of precipitation during sydney 2000: an appraisal of the qpf algorithms. *Weather And Forecasting*, 19:7–21, 2004.

J.E. Potter. *Astronautical Guidance*, pages 338–339. McGraw-Hill, NY, 1964.

C. Price. An analysis of the divergence problem in the Kalman filter. *IEEE Transactions on Automatic Control*, 13(6):699–702, 1968.

F. Rabier, H. Jarvinen, E. Klinker, J.F. Mahfouf, and A. Simmons. The ECMWF operational implementation of four-dimensional variational assimilation. I: Experimental results with simplified physics. *Quarterly Journal of the Royal Meteorological Society*, 126:1143–1170, 2000.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*, chapter 4, pages 84–85. The MIT Press, 2006.

F. Rawlins, S.P. Ballaerd, K.J. Bovis, A.M. Clayton, D. Li, GW Inverarity, A.C. Lorenc, and T.J. Payne. The Met Ooffice global four-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 133(623 B):347–362, 2007.

S. Reed, J. Schaake, and Z. Zhang. A distributed hydrologic model and threshold frequency-based method for flash flood forecasting at ungauged locations. *Journal of Hydrology*, 337(3-4):402–420, 2007.

I.B. Rhodes. A tutorial introduction to estimation and filtering. *IEEE Transactions on Automatic Control*, 16(6):688–706, December 1971.

R. Rinehart and E. Garvey. Three-dimensional storm motion detection by conventional weather radar. *Nature*, 273(5660):287–289, 1978.

I. Rodriguez-Iturbe, DR Cox, and V. Isham. Some Models for Rainfall Based on Stochastic Point Processes. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences (1934-1990)*, 410(1839):269–288, 1987.

AW Seed. A Dynamic and Spatial Scaling Approach to Advection Forecasting. *Journal of Applied Meteorology*, 42(3):381–388, 2003.

Y. Shen, C. Archambeau, D. Cornford, M. Opper, J. Shawe-Taylor, and R. Barillec. Evaluation of Variational and Markov Chain Monte Carlo Methods for Inference in Partially Observed Stochastic Dynamic Systems. *Machine Learning for Signal Processing, 2007 IEEE Workshop on*, pages 306–311, 2007.

JA Smith and WF Krajewski. Statistical modeling of space-time rainfall using radar and rain gage observations. *Water Resources Research*, 23(10), 1987.

J.A. Smith, M.L. Baeck, K.L. Meierdiercks, A.J. Miller, and W.F. Krajewski. Radar rainfall estimation for flash flood forecasting in small urban watersheds. *Advances in Water Resources*, 30 (10):2087–2097, 2007.

K.T. Smith and G.L. Austin. Nowcasting precipitation - a proposal for a way forward. *Journal of Hydrology*, 239, 2000.

JC Smithers, GGS Pegram, and RE Schulze. Design rainfall estimation in South Africa using Bartlett–Lewis rectangular pulse rainfall models. *Journal of Hydrology*, 258(1-4):83–99, 2002.

X. Sun, RG Mein, TD Keenan, and JF Elliott. Flood estimation using radar and raingauge data. *Journal of Hydrology*, 239(1-4):4–18, 2000. doi: 10.1016/S0022-1694(00)00350-4.

Y. Tessier, S. Lovejoy, and D. Schertzer. Universal Multifractals: Theory and Observations for Rain and Clouds. *Journal of Applied Meteorology*, 32(2):223–250, 1993.

M K Tippet, J L Anderson, C H Bishop, T M Hamill, and J S Whitaker. Ensemble square-root filters. *Monthly Weather Review*, 131:1485–1490, 2003.

M. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

Y. Tremolet. Accounting for an imperfect model in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 132(621 B):2483–2504, 2006.

Y. Tremolet. Model-error estimation in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 133(626 A):1267–1280, 2007.

AA Tsonis and GL Austin. An evaluation of extrapolation techniques for the short-term prediction of rain amounts. *Atmos.–Ocean*, 19:54–65, 1981.

UK Meteorological Office. British atmospheric data centre, rain radar products (nimrod). 2003.

R. Van der Merwe, N. De Freitas, A. Doucet, and E. Wan. The unscented particle filter. Technical Report CUED/F-INFENG/TR380, Cambridge University Engineering Department, August 2000.

R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The Unscented Particle Filter. *Advances in Neural Information Processing Systems*, pages 584–590, 2001.

D. Veneziano, P. Furcolo, and V. Iacobellis. Imperfect scaling of time and space–time rainfall. *Journal of Hydrology*, 322(1-4):105–119, 2006.

B. Vieux and J. Vieux. Statistical evaluation of a radar rainfall system for sewer system management. *Atmospheric Research*, 77(1-4):322–336, 2005.

EA Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158, 2000.

E. W. Weisstein. *CRC Concise Encyclopedia of Mathematics*. Chapman & Hall/CRC, 2nd edition, 2002.

H. S. Wheater, V. S. Isham, D. R. Cox, R. E. Chandler, A. Kakou, P. K. Northrop, L. Oh, C. Onof, and I. Rodriguez-Iturbe. Spatial temporal rainfall fields: modelling and statistical aspects. *Hydrology and Earth System Sciences*, 4(4):581–601, 2000.

J. S. Whitaker and T. M. Hamill. Ensemble data assimilation without perturbed observations. *Monthly Weather Review*, 130:1913–1924, 2002.

C. Wikle. A kernel-based spectral model for non-gaussian spatio temporal processes. *Statistical Modelling*, 2(4):299–314, 2002.

J. Wilson, A. Crook, C. Mueller, J. Sun, and M. Dixon. Nowcasting thunderstorms: A status report. *Bulletin of the American Meteorological Society*, 79(10):2079–2099, October 1998.

X. Xiong and I.M. Navon. A Note on the Particle Filter with Posterior Gaussian Resampling. *Tellus*, 58:456–460, 2006.

K. Xu, C.K. Wikle, and N.I. Fox. A Kernel-Based Spatio-Temporal Dynamical Model for Nowcasting Weather Radar Reflectivities. *Journal of the American Statistical Association*, 100(472): 1133–1144, 2005.

H. Zhang and T. Casey. Verification of categorical probability forecasts. *Weather and Forecasting*, 15, 1999.

D. Zupanski. A General Weak Constraint Applicable to Operational 4DVAR Data Assimilation Systems. *Monthly Weather Review*, 125(9):2274–2292, 1997.

M. Zupanski. Regional Four-Dimensional Variational Data Assimilation in a Quasi-Operational Forecasting Environment. *Monthly Weather Review*, 121(8):2396–2408, 1993.

A Computation of the KL divergence

This appendix details the derivation of the KL divergence for the rainfall model from Chapter 6. Recall from Equation (6.48) the expression of the KL divergence between the posterior and the approximating distribution:

$$\mathrm{KL}(\mathrm{p} \parallel \mathrm{q}) = -\int \mathrm{q}(\mathbf{x}_t) \ln \frac{\mathrm{p}(\mathbf{x}_t|\mathbf{Y}_t)}{\mathrm{q}(\mathbf{x}_t)} \, \mathrm{d}\mathbf{x}_t. \tag{A.1}$$

Using Bayes rule (Eq. (2.11)), the negative logarithm of the posterior can be expanded:

$$-\ln \mathrm{p}(\mathbf{x}_t|\mathbf{Y}_t) = -\ln \mathrm{p}(\mathbf{y}_t|\mathbf{x}_t) - \ln \mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1}) + \ln \mathrm{p}(\mathbf{Y}_t). \tag{A.2}$$

Substituting (A.2) into (A.1) yields:

$$\mathrm{KL}(\mathrm{p} \parallel \mathrm{q}) = -\left\langle \ln \mathrm{p}(\mathbf{y}_t|\mathbf{x}_t) \right\rangle_{\mathrm{q}(\mathbf{x}_t)} - \left\langle \ln \frac{\mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1})}{\mathrm{q}(\mathbf{x}_t)} \right\rangle_{\mathrm{q}(\mathbf{x}_t)} + \left\langle \ln \mathrm{p}(\mathbf{Y}_t) \right\rangle_{\mathrm{q}(\mathbf{x}_t)} \tag{A.3}$$

The three terms are computed separately in the rest of this appendix. The first term is the KL divergence of the likelihood (with respect to q) and the second term is the KL divergence of the predicted distribution (acting here as a prior). The last term is a constant with respect to $\mathbf{x}_t$ and can be discarded since we are only interested in minimising (A.3).

To ease the notation, the time index is dropped in the developments below. The conditioning of the prior on $\mathbf{x}_{t-1}$ is not indicated anymore (i.e. $\mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1})$ is simply denoted $\mathrm{p}(\mathbf{x})$) but is still taken into account in the derivations.

A.1 Negative log likelihood term: $-\langle \ln p(\mathbf{y}|\mathbf{x}) \rangle_{q,\mathbf{x}}$

Let us first expand the negative log-likelihood. Under the assumption that $\mathbf{y}$ is a vector of length M (the number of pixels in the radar image) and that there is no spatial correlation, i.e. $\mathbf{R} = \sigma^2\mathbf{I}$, we can write:

$$-\ln p(\mathbf{y}|\mathbf{x}) = -\ln\left((2\pi)^{-M/2}\,|\mathbf{R}|^{-1/2}\,e^{-\frac{1}{2}(h(\mathbf{x})-\mathbf{y})^{\mathrm{T}}\mathbf{R}^{-1}(h(\mathbf{x})-\mathbf{y})}\right) \tag{A.4}$$

$$\approx \frac{1}{2}(h(\mathbf{x})-\mathbf{y})^{\mathrm{T}}\mathbf{R}^{-1}(h(\mathbf{x})-\mathbf{y}) \tag{A.5}$$

$$\approx \frac{1}{2\sigma^2}\sum_{j=1}^{M}\left(h(\mathbf{x},\mathbf{s}_j)-y_j\right)^2 \tag{A.6}$$

$$\approx \frac{1}{2\sigma^2}\sum_{j=1}^{M}\left(h(\mathbf{x},\mathbf{s}_j)^2 - 2h(\mathbf{x},\mathbf{s}_j)y_j + y_j^2\right) \tag{A.7}$$

under the assumption that the observations are uncorrelated in space, and after dropping the constant term.

Taking the expectation with respect to $q(\mathbf{x})$ thus yields:

$$-\left\langle \ln p(\mathbf{y}|\mathbf{x}) \right\rangle_{q(\mathbf{x})} \approx \frac{1}{2\sigma^2}\sum_{j=1}^{M}\left[\left\langle h(\mathbf{x},\mathbf{s}_j)^2\right\rangle_{q(\mathbf{x})} - 2y_j\left\langle h(\mathbf{x},\mathbf{s}_j)\right\rangle_{q(\mathbf{x})} + y_j^2\right] \tag{A.8}$$

Note that this can be rewritten as follow:

$$-\left\langle \ln p(\mathbf{y}|\mathbf{x}) \right\rangle_{q(\mathbf{x})} \approx \frac{1}{2\sigma^2}\sum_{j=1}^{M}\left[\left\langle\left(h(\mathbf{x},\mathbf{s}_j) - \left\langle h(\mathbf{x},\mathbf{s}_j)\right\rangle_{q(\mathbf{x})}\right)^2\right\rangle_{q(\mathbf{x})} + \left(\left\langle h(\mathbf{x},\mathbf{s}_j)\right\rangle_{q(\mathbf{x})} - y_j\right)^2\right] \tag{A.9}$$

where the first term in the sum is the variance of the rainfield and the second is the square deviation of the expected rainfield from the observation. However, we'll focus on the formulation from Equation (A.8) in the following sections.

A.1.1 Computation of $\langle h(\mathbf{x},\mathbf{s}_j)\rangle_{q,\mathbf{x}}$

The rainfield at a given location $\mathbf{s}_j$ is by definition the sum of the contributions of all rain cells at this location:

$$h(\mathbf{x},\mathbf{s}_j) = \sum_{k=1}^{N} h(\mathbf{x}_k,\mathbf{s}_j) \tag{A.10}$$

$$= \sum_{k=1}^{N} r_k\, e^{-\frac{1}{2w_k}(\mathbf{c}_k-\mathbf{s}_j)^{\mathrm{T}}(\mathbf{c}_k-\mathbf{s}_j)} \tag{A.11}$$

Since we assumed earlier that cells' parameters were uncorrelated, i.e. $\mathrm{q}(\mathbf{x}) = \prod_{k=1}^{N} \mathrm{q}(\mathbf{x}_k)$, the expectation of the rainfield can be factorised as the sum of the expectations for each cell:

$$\left\langle h(\mathbf{x}, \mathbf{s}_j) \right\rangle_{\mathrm{q}(\mathbf{x})} = \sum_{k=1}^{N} \left\langle h(\mathbf{x}_k, \mathbf{s}_j) \right\rangle_{\mathrm{q}(\mathbf{x}_k)} \tag{A.12}$$

For computation purposes, we can thus consider the case of a single-celled rainfield, and drop the k index for improved readability.

$$h(\mathbf{x}, \mathbf{s}_j) = r\, e^{-\frac{1}{2w}(\mathbf{c}-\mathbf{s}_j)^{\mathrm{T}}(\mathbf{c}-\mathbf{s}_j)} \tag{A.13}$$

Let us then compute the expectation. We note that the cell's height h can be integrated separately as it does not depend on the other two parameters, whereas the centre , conditionned on the width w, requires that we first integrate out one of the parameters (the centre, for computational ease).

$$\left\langle h(\mathbf{x}, \mathbf{s}_j) \right\rangle_{\mathrm{q}(\mathbf{x})} = \int \mathrm{q}(r)\, r\, \mathrm{d}r \iint \mathrm{q}(\mathbf{c}|w)\, \mathrm{q}(w)\, e^{-\frac{1}{2w}(\mathbf{c}-\mathbf{s}_j)^{\mathrm{T}}(\mathbf{c}-\mathbf{s}_j)}\, \mathrm{d}\mathbf{c}\, \mathrm{d}w \tag{A.14}$$

$$= \left\langle r \right\rangle_{\mathrm{q}(r)} \int \mathrm{q}(w) \left[\int \mathrm{q}(\mathbf{c}|w)\, e^{-\frac{1}{2w}(\mathbf{c}-\mathbf{s}_j)^{\mathrm{T}}(\mathbf{c}-\mathbf{s}_j)}\, \mathrm{d}\mathbf{c} \right] \mathrm{d}w \tag{A.15}$$

$$= \frac{\gamma}{\delta} \left\langle f_1(w) \right\rangle_{\mathrm{q}(w)} \tag{A.16}$$

where we used the notation

$$f_1(w) = \int \mathrm{q}(\mathbf{c}|w)\, e^{-\frac{1}{2w}(\mathbf{c}-\mathbf{s}_j)^{\mathrm{T}}(\mathbf{c}-\mathbf{s}_j)}\, \mathrm{d}\mathbf{c} \tag{A.17}$$

$$= \frac{1}{2\pi\xi w} \int e^{-\frac{1}{2\xi w}(\mathbf{c}-\bar{\mathbf{c}})^{\mathrm{T}}(\mathbf{c}-\bar{\mathbf{c}})}\, e^{-\frac{1}{2w}(\mathbf{c}-\mathbf{s}_j)^{\mathrm{T}}(\mathbf{c}-\mathbf{s}_j)}\, \mathrm{d}\mathbf{c} \tag{A.18}$$

The product of exponentials in Equation (A.18) can be reformulated so that parameter $\mathbf{c}$ appears in only one of them (see for example Petersen and Pedersen (2006)). Such manipulation leaves us now with the following integral:

$$f_1(w) = \frac{1}{2\pi\xi w} e^{-\frac{1}{2w(1+\xi)}(\bar{\mathbf{c}}-\mathbf{s}_j)^{\mathrm{T}}(\bar{\mathbf{c}}-\mathbf{s}_j)} \int e^{-\frac{1+\xi}{2\xi w}(\mathbf{c}-\frac{\bar{\mathbf{c}}+\xi\mathbf{s}_j}{1+\xi})^{\mathrm{T}}(\mathbf{c}-\frac{\bar{\mathbf{c}}+\xi\mathbf{s}_j}{1+\xi})}\, \mathrm{d}\mathbf{c} \tag{A.19}$$

The Gaussian integral in $\mathbf{c}$ integrates out to the constant $\frac{2\pi\xi w}{1+\xi}$, thus leaving:

$$f_1(w) = \frac{1}{1+\xi} e^{-\frac{1}{2w(1+\xi)}(\bar{\mathbf{c}}-\mathbf{s}_j)^{\mathrm{T}}(\bar{\mathbf{c}}-\mathbf{s}_j)} \tag{A.20}$$

We now only need to compute the expectation of $f_1(w)$ to solve Equation (A.16). To that effect, let us consider the following expectation, where E is some constant factor and $\mathrm{q}(w)$ an Inverse

Gamma distribution:

$$\left\langle e^{-\frac{1}{w}E} \right\rangle_{q(w)} = \int q(w)\, e^{-\frac{1}{w}E}\, dw \tag{A.21}$$

$$= \frac{\beta^\alpha}{\Gamma(\alpha)} \int w^{-\alpha-1} e^{-\frac{\beta}{w}} e^{-\frac{E}{w}}\, dw \tag{A.22}$$

$$= \frac{\beta^\alpha}{\Gamma(\alpha)} \int \lambda^{\alpha-1} e^{-\beta\lambda} e^{-E\lambda}\, d\lambda. \qquad \lambda = \frac{1}{w} \tag{A.23}$$

$$= \frac{\beta^\alpha}{\Gamma(\alpha)} \int \lambda^{\alpha-1} e^{-(\beta+E)\lambda}\, d\lambda \tag{A.24}$$

$$= \frac{\beta^\alpha}{\Gamma(\alpha)} (\beta+E)^{-\alpha} \Gamma(\alpha) \tag{A.25}$$

$$= \left(1+\frac{E}{\beta}\right)^{-\alpha} \tag{A.26}$$

Applying this result to Equation (A.20) and substituting in Equation (A.16) leads to the following expression, after restoring the k index:

$$\left\langle h(\mathbf{x}_k, \mathbf{s}_j) \right\rangle_{q(\mathbf{x}_k)} = \frac{\gamma_k}{\delta_k} \frac{1}{1+\xi_k} \left(1 + \frac{1}{2\beta_k(1+\xi_k)} (\bar{\mathbf{c}}_k - \mathbf{s}_j)^{\mathrm{T}} (\bar{\mathbf{c}}_k - \mathbf{s}_j)\right)^{-\alpha_k} \tag{A.27}$$

A.1.2 Computation of $\left\langle h(\mathbf{x}, \mathbf{s}_j)^2 \right\rangle_{q,\mathbf{x}}$

We first expand $\left\langle h(\mathbf{x}, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x})}$ to separate the square term and the double products.

$$h(\mathbf{x}, \mathbf{s}_j)^2 = \left(\sum_{k=1}^{N} h(\mathbf{x}_k, \mathbf{s}_j)\right)^2 \tag{A.28}$$

$$= \sum_{k=1}^{N} h(\mathbf{x}_k, \mathbf{s}_j)^2 + 2 \sum_{1 \le k < l \le N} h(\mathbf{x}_k, \mathbf{s}_j) h(\mathbf{x}_l, \mathbf{s}_j) \tag{A.29}$$

Using our assumption that cells' parameters are uncorrelated, i.e. $q(\mathbf{x}) = q(\mathbf{x}_1, .., \mathbf{x}_N) = \prod_{k=1}^{N} q(\mathbf{x}_k)$, the expectation becomes:

$$\begin{aligned} \left\langle h(\mathbf{x}, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x})} &= \sum_{k=1}^{N} \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)} \\ &\quad + 2 \sum_{1 \le k < l \le N} \left\langle h(\mathbf{x}_k, \mathbf{s}_j) \right\rangle_{q(\mathbf{x}_k)} \left\langle h(\mathbf{x}_l, \mathbf{s}_j) \right\rangle_{q(\mathbf{x}_l)} \end{aligned} \tag{A.30}$$

The double product is evaluated using the result from Equation (A.27). The computation of the $\left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)}$ term is very similar to that of $\left\langle h(\mathbf{x}, \mathbf{s}_j) \right\rangle_{q(\mathbf{x})}$ (see paragraph A.1.1). Following the same reasoning, we can restrict ourselves to the case of a single-celled rainfield and drop the k

index for readability. Expanding $\left\langle h(\mathbf{x},\mathbf{s}_j)^2 \right\rangle_{\mathrm{q}(\mathbf{x})}$ gives:

$$\left\langle h(\mathbf{x},\mathbf{s}_j)^2 \right\rangle_{\mathrm{q}(\mathbf{x})} = \iiint \mathrm{q}(r)\mathrm{q}(\mathbf{c}|w)\,\mathrm{q}(w)\, r^2 e^{-\frac{2}{2w}(\mathbf{c}-\mathbf{s}_j)^{\mathrm{T}}(\mathbf{c}-\mathbf{s}_j)}\,\mathrm{d}h\,\mathrm{d}\mathbf{c}\,\mathrm{d}w \tag{A.31}$$

$$= \left\langle r^2 \right\rangle_{\mathrm{q}(r)} \iint \mathrm{q}(\mathbf{c}|w)\,\mathrm{q}(w)\, e^{-\frac{2}{2w}(\mathbf{c}-\mathbf{s}_j)^{\mathrm{T}}(\mathbf{c}-\mathbf{s}_j)}\,\mathrm{d}\mathbf{c}\,\mathrm{d}w \tag{A.32}$$

$$= \frac{\gamma(\gamma+1)}{\delta^2} \left\langle f_2(w) \right\rangle_{\mathrm{q}(w)} \tag{A.33}$$

with

$$f_2(w) = \int \mathrm{q}(\mathbf{c}|w)\, e^{-\frac{2}{2w}(\mathbf{c}-\mathbf{s}_j)^{\mathrm{T}}(\mathbf{c}-\mathbf{s}_j)}\,\mathrm{d}\mathbf{c} \tag{A.34}$$

$$= \frac{1}{2\pi\xi w} \int e^{-\frac{1}{2\xi w}(\mathbf{c}-\bar{\mathbf{c}})^{\mathrm{T}}(\mathbf{c}-\bar{\mathbf{c}})}\, e^{-\frac{2}{2w}(\mathbf{c}-\mathbf{s}_j)^{\mathrm{T}}(\mathbf{c}-\mathbf{s}_j)}\,\mathrm{d}\mathbf{c} \tag{A.35}$$

Equation (A.35) can be rewritten so that $\mathbf{c}$ appears in only one of the exponentials:

$$f_2(w) = \frac{1}{2\pi\xi w} e^{-\frac{1}{(1+2\xi)w}(\bar{\mathbf{c}}-\mathbf{s}_j)^{\mathrm{T}}(\bar{\mathbf{c}}-\mathbf{s}_j)} \int e^{-\frac{1+2\xi}{2\xi w}(\mathbf{c}-\frac{\bar{\mathbf{c}}+2\xi\mathbf{s}_j}{1+2\xi})^{\mathrm{T}}(\mathbf{c}-\frac{\bar{\mathbf{c}}+2\xi\mathbf{s}_j}{1+2\xi})}\,\mathrm{d}\mathbf{c} \tag{A.36}$$

The integral evaluates to the constant $\frac{2\pi\xi w}{1+2\xi}$, leaving:

$$f_2(w) = \frac{1}{1+2\xi} e^{-\frac{1}{(1+2\xi)w}(\bar{\mathbf{c}}-\mathbf{s}_j)^{\mathrm{T}}(\bar{\mathbf{c}}-\mathbf{s}_j)} \tag{A.37}$$

Applying the result from Equation (A.26) to $f_2(w)$, and substituting in Equation (A.33) gives the following expression for $\left\langle h(\mathbf{x},\mathbf{s}_j)^2 \right\rangle_{\mathrm{q}(\mathbf{x})}$:

$$\left\langle h(\mathbf{x}_k,\mathbf{s}_j)^2 \right\rangle_{\mathrm{q}(\mathbf{x}_k)} = \frac{\gamma_k(\gamma_k+1)}{\delta_k^2} \frac{1}{1+2\xi_k} \left(1 + \frac{(\bar{\mathbf{c}}_k-\mathbf{s}_j)^{\mathrm{T}}(\bar{\mathbf{c}}_k-\mathbf{s}_j)}{\beta_k(1+2\xi_k)}\right)^{-\alpha_k} \tag{A.38}$$

Substituting (A.27) and (A.38) into (A.8) gives the complete expression for the likelihood term in the KL divergence.

A.1.3 Gradient of $\left\langle h(\mathbf{x}_k,\mathbf{s}_j) \right\rangle_{\mathrm{q},\mathbf{x}_k}$

We let $\mathrm{E}_{k,j} = 1 + \frac{(\bar{\mathbf{c}}_k-\mathbf{s}_j)^{\mathrm{T}}(\bar{\mathbf{c}}_k-\mathbf{s}_j)}{2\beta_k(1+\xi_k)}$ in the following expressions for better readability.

$$\frac{\partial \left\langle h(\mathbf{x}_k,\mathbf{s}_j) \right\rangle_{\mathrm{q}(\mathbf{x}_k)}}{\partial \bar{\mathbf{c}}_k} = -\frac{\gamma_k}{\delta_k} \frac{\alpha_k(\bar{\mathbf{c}}_k-\mathbf{s}_j)}{\beta_k(1+\xi_k)^2} (\mathrm{E}_{k,j})^{-\alpha_k-1} \tag{A.39}$$

$$= -\frac{\alpha_k}{\beta_k} \frac{\bar{\mathbf{c}}_k-\mathbf{s}_j}{(1+\xi_k)\mathrm{E}_{k,j}} \left\langle h(\mathbf{x}_k,\mathbf{s}_j) \right\rangle_{\mathrm{q}(\mathbf{x}_k)} \tag{A.40}$$

$$\frac{\partial \left\langle h(\mathbf{x}_k,\mathbf{s}_j) \right\rangle_{\mathrm{q}(\mathbf{x}_k)}}{\partial \xi_k} = \frac{\gamma_k}{\delta_k} \frac{1}{(1+\xi_k)^2} \left((\alpha_k-1)(\mathrm{E}_{k,j}-1)-1\right) (\mathrm{E}_{k,j})^{-\alpha_k-1} \tag{A.41}$$

$$= \frac{1}{1+\xi_k} \left((\alpha_k-1)(\mathrm{E}_{k,j}-1)-1\right) \frac{1}{\mathrm{E}_{k,j}} \left\langle h(\mathbf{x}_k,\mathbf{s}_j) \right\rangle_{\mathrm{q}(\mathbf{x}_k)} \tag{A.42}$$

$$\frac{\partial \left\langle h(\mathbf{x}_k, \mathbf{s}_j) \right\rangle_{q(\mathbf{x}_k)}}{\partial \alpha_k} = -\frac{\gamma_k}{\delta_k} \frac{1}{(1+\xi_k)} \ln(\mathrm{E}_{k,j}) \, (\mathrm{E}_{k,j})^{-\alpha_k} \tag{A.43}$$

$$= -\ln(\mathrm{E}_{k,j}) \left\langle h(\mathbf{x}_k, \mathbf{s}_j) \right\rangle_{q(\mathbf{x}_k)} \tag{A.44}$$

$$\tag{A.45}$$

$$\frac{\partial \left\langle h(\mathbf{x}_k, \mathbf{s}_j) \right\rangle_{q(\mathbf{x}_k)}}{\partial \beta_k} = \frac{\gamma_k}{\delta_k} \frac{\alpha_k}{\beta_k^2} \frac{(\bar{\mathbf{c}}_k - \mathbf{s}_j)^{\mathrm{T}} (\bar{\mathbf{c}}_k - \mathbf{s}_j)}{2(1+\xi_k)^2} \, (\mathrm{E}_{k,j})^{-\alpha_k - 1} \tag{A.46}$$

$$= \frac{\alpha_k}{\beta_k^2} \frac{(\bar{\mathbf{c}}_k - \mathbf{s}_j)^{\mathrm{T}} (\bar{\mathbf{c}}_k - \mathbf{s}_j)}{2(1+\xi_k)\mathrm{E}_{k,j}} \left\langle h(\mathbf{x}_k, \mathbf{s}_j) \right\rangle_{q(\mathbf{x}_k)} \tag{A.47}$$

$$\tag{A.48}$$

$$\frac{\partial \left\langle h(\mathbf{x}_k, \mathbf{s}_j) \right\rangle_{q(\mathbf{x}_k)}}{\partial \gamma_k} = \frac{1}{\delta_k} \frac{1}{1+\xi_k} \, (\mathrm{E}_{k,j})^{-\alpha_k} \tag{A.49}$$

$$= \frac{1}{\gamma_k} \left\langle h(\mathbf{x}_k, \mathbf{s}_j) \right\rangle_{q(\mathbf{x}_k)} \tag{A.50}$$

$$\tag{A.51}$$

$$\frac{\partial \left\langle h(\mathbf{x}_k, \mathbf{s}_j) \right\rangle_{q(\mathbf{x}_k)}}{\partial \gamma_k} = -\frac{\gamma_k}{\delta_k^2} \frac{1}{1+\xi_k} \, (\mathrm{E}_{k,j})^{-\alpha_k} \tag{A.52}$$

$$= -\frac{1}{\delta_k} \left\langle h(\mathbf{x}_k, \mathbf{s}_j) \right\rangle_{q(\mathbf{x}_k)} \tag{A.53}$$

A.1.4 Gradient of $\int q \, h(\mathbf{x}_k, \mathbf{s}_j)^2 \, d\mathbf{x}_k$

We let $\mathrm{F}_{k,j} = 1 + \frac{(\bar{\mathbf{c}}_k - \mathbf{s}_j)^{\mathrm{T}} (\bar{\mathbf{c}}_k - \mathbf{s}_j)}{\beta_k (1+2\xi_k)}$ in the following expressions for better readability.

$$\frac{\partial \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)}}{\partial \bar{\mathbf{c}}_k} = -\frac{\gamma_k(\gamma_k+1)}{\delta^2} \frac{1}{1+2\xi_k} \frac{2\alpha_k}{\beta_k(1+2\xi_k)} (\bar{\mathbf{c}}_k - \mathbf{s}_j) \mathrm{F}_{k,j}^{-\alpha_k-1} \tag{A.54}$$

$$= -\frac{2\alpha_k}{\beta_k} \frac{(\bar{\mathbf{c}}_k - \mathbf{s}_j)}{(1+2\xi_k)\mathrm{F}_{k,j}} \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)} \tag{A.55}$$

(A.56)

$$\frac{\partial \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)}}{\partial \xi_k} = \frac{\gamma_k(\gamma_k+1)}{\delta^2} \frac{2}{(1+2\xi_k)^2} \left((\alpha_k - 1)(\mathrm{F}_{k,j} - 1) - 1 \right) \mathrm{F}_{k,j}^{-\alpha_k-1} \tag{A.57}$$

$$= \frac{2}{1+2\xi_k} \left((\alpha_k - 1)(\mathrm{F}_{k,j} - 1) - 1 \right) \frac{1}{\mathrm{F}_{k,j}} \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)} \tag{A.58}$$

(A.59)

$$\frac{\partial \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)}}{\partial \alpha_k} = -\frac{\gamma_k(\gamma_k+1)}{\delta^2} \frac{1}{1+2\xi_k} \ln(\mathrm{F}_{k,j}) (\mathrm{F}_{k,j})^{-\alpha_k} \tag{A.60}$$

$$= -\ln(\mathrm{F}_{k,j}) \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)} \tag{A.61}$$

(A.62)

$$\frac{\partial \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)}}{\partial \beta_k} = \frac{\gamma_k(\gamma_k+1)}{\delta^2} \frac{1}{1+2\xi_k} \frac{\alpha_k}{\beta_k^2} \frac{(\bar{\mathbf{c}}_k - \mathbf{s}_j)^{\mathrm{T}} (\bar{\mathbf{c}}_k - \mathbf{s}_j)}{(1+2\xi_k)} \mathrm{F}_{k,j}^{-\alpha_k-1} \tag{A.63}$$

$$= \frac{\alpha_k}{\beta_k^2} \frac{(\bar{\mathbf{c}}_k - \mathbf{s}_j)^{\mathrm{T}} (\bar{\mathbf{c}}_k - \mathbf{s}_j)}{(1+2\xi_k)\mathrm{F}_{k,j}} \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)} \tag{A.64}$$

(A.65)

$$\frac{\partial \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)}}{\partial \gamma_k} = \frac{2\gamma_k + 1}{\delta^2} \frac{1}{1+2\xi_k} (\mathrm{F}_{k,j})^{-\alpha_k} \tag{A.66}$$

$$= \left(\frac{1}{\gamma_k} + \frac{1}{\gamma_k + 1} \right) \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)} \tag{A.67}$$

(A.68)

$$\frac{\partial \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)}}{\partial \gamma_k} = -\frac{2\gamma_k(\gamma_k+1)}{\delta^3} \frac{1}{1+2\xi_k} (\mathrm{F}_{k,j})^{-\alpha_k} \tag{A.69}$$

$$= -\frac{2}{\delta_k} \left\langle h(\mathbf{x}_k, \mathbf{s}_j)^2 \right\rangle_{q(\mathbf{x}_k)} \tag{A.70}$$

A.2 KL divergence of the prior: $-\left\langle \ln \frac{\mathrm{p}(\mathbf{x})}{\mathrm{q}(\mathbf{x})} \right\rangle_{\mathrm{q},\mathbf{x}}$

Using the assumption that parameters are uncorrelated between cells, we can write:

$$\ln \frac{\mathrm{p}(\mathbf{x})}{\mathrm{q}(\mathbf{x})} = \ln \frac{\mathrm{p}(h)\mathrm{p}(c|w)\mathrm{p}(w)}{\mathrm{q}(h)\mathrm{q}(\mathbf{c}|w)\mathrm{q}(w)} \tag{A.71}$$

$$= \sum_{k=1}^{N} \ln \frac{\mathrm{p}(h_k)\mathrm{p}(\mathbf{c}_k|w_k)\mathrm{p}(w_k)}{\mathrm{q}(h_k)\mathrm{q}(\mathbf{c}_k|w_k)\mathrm{q}(w_k)} \tag{A.72}$$

$$= \sum_{k=1}^{N} \left(\ln \frac{\mathrm{p}(h_k)}{\mathrm{q}(h_k)} + \ln \frac{\mathrm{p}(\mathbf{c}_k|w_k)\mathrm{p}(w_k)}{\mathrm{q}(\mathbf{c}_k|w_k)\mathrm{q}(w_k)} \right) \tag{A.73}$$

and

$$-\left\langle \ln \frac{\mathrm{p}(\mathbf{x})}{\mathrm{q}(\mathbf{x})} \right\rangle_{\mathrm{q}(\mathbf{x})} = -\sum_{k=1}^{N} \left\langle \ln \frac{\mathrm{p}(h_k)}{\mathrm{q}(h_k)} \right\rangle_{\mathrm{q}(h_k)} - \sum_{k=1}^{N} \left\langle \ln \frac{\mathrm{p}(\mathbf{c}_k|w_k)\mathrm{p}(w_k)}{\mathrm{q}(\mathbf{c}_k|w_k)\mathrm{q}(w_k)} \right\rangle_{\mathrm{q}(\mathbf{c}_k,w_k)} \tag{A.74}$$

Considering a single cell, and dropping the k index, we compute separately the KL divergences for h and $(\mathbf{c}, w)$.

A.2.1 Computation of $-\left\langle \ln \frac{\mathrm{p}(h)}{\mathrm{q}(h)} \right\rangle_{\mathrm{q},h}$

As stated in Penny (2001), we have the following result for the KL divergence between the Gamma distributions $\mathrm{p}(r)$ and $\mathrm{q}(r)$ with respective parameters γ', δ' and γ, δ:

$$-\left\langle \ln \frac{\mathrm{p}(h)}{\mathrm{q}(h)} \right\rangle_{\mathrm{q}(h)} = -\int \mathrm{q}(h) \ln \frac{\mathrm{p}(h)}{\mathrm{q}(h)} \, \mathrm{d}h \tag{A.75}$$

$$= \gamma \ln \delta - \gamma' \ln \delta' - \ln \frac{\Gamma(\gamma)}{\Gamma(\gamma')} + (\gamma - \gamma')\left[\Psi(\gamma) - \ln \delta\right] - \gamma \left(1 - \frac{\delta'}{\delta}\right) \tag{A.76}$$

A.2.2 Computation of $-\left\langle \ln \frac{\mathrm{p}(\mathbf{c}|w)\mathrm{p}(w)}{\mathrm{q}(\mathbf{c}|w)\mathrm{q}(w)} \right\rangle_{\mathrm{q},\mathbf{c},w}$

$$-\left\langle \ln \frac{\mathrm{p}(\mathbf{c}|w)\mathrm{p}(w)}{\mathrm{q}(\mathbf{c}|w)\mathrm{q}(w)} \right\rangle_{\mathrm{q}(\mathbf{c},w)} = -\iint \mathrm{q}(\mathbf{c}|w)\,\mathrm{q}(w) \ln \frac{\mathrm{p}(\mathbf{c}|w)\,\mathrm{p}(w)}{\mathrm{q}(\mathbf{c}|w)\,\mathrm{q}(w)} \, \mathrm{d}\mathbf{c}\,\mathrm{d}w$$

$$= -\int \mathrm{q}(w) \ln \frac{\mathrm{p}(w)}{\mathrm{q}(w)} \, \mathrm{d}w + \int \mathrm{q}(w) \left(-\int \mathrm{q}(\mathbf{c}|w) \ln \frac{\mathrm{p}(\mathbf{c}|w)}{\mathrm{q}(\mathbf{c}|w)} \, \mathrm{d}\mathbf{c} \right) \mathrm{d}w \tag{A.77}$$

The first term in the sum is the KL divergence between two Inverse Gamma distributions with respective parameters α', β' and α, β. The following shows that the KL divergence between the two Inverse Gamma distributions is equivalent to the KL divergence between two Gamma

distributions with identical parameters.

$$-\int \mathrm{IGa}(w|\alpha,\beta)\ln\frac{\mathrm{IGa}(w|\alpha',\beta')}{\mathrm{IGa}(w|\alpha,\beta)}\,\mathrm{d}w \tag{A.78}$$

$$=-\int \frac{\beta^{\alpha}}{\Gamma(\alpha)}w^{-\alpha-1}e^{-\frac{\beta}{w}}\ln\left(\frac{\beta'^{\alpha'}\Gamma(\alpha)}{\beta^{\alpha}\Gamma(\alpha')}w^{-(\alpha'-\alpha)}e^{-\frac{\beta'-\beta}{w}}\right)\mathrm{d}w \tag{A.79}$$

$$=-\int \frac{\beta^{\alpha}}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}\ln\left(\frac{\beta'^{\alpha'}\Gamma(\alpha)}{\beta^{\alpha}\Gamma(\alpha')}\lambda^{\alpha'-\alpha}e^{-(\beta'-\beta)\lambda}\right)\mathrm{d}\lambda \tag{A.80}$$

$$=-\int \mathrm{Ga}(\lambda|\alpha,\beta)\ln\frac{\mathrm{Ga}(\lambda|\alpha',\beta')}{\mathrm{Ga}(\lambda|\alpha,\beta)}\,\mathrm{d}w \tag{A.81}$$

$$=\alpha\ln\beta-\alpha'\ln\beta'-\ln\frac{\Gamma(\alpha)}{\Gamma(\alpha')}+(\alpha-\alpha')\left[\Psi(\alpha)-\ln\beta\right]-\alpha\left(1-\frac{\beta'}{\beta}\right) \tag{A.82}$$

The integral in $\mathbf{c}$ is the KL divergence over the centre. Given that $\mathrm{p}(\mathbf{c}|w)=\mathcal{N}(\mathbf{c}\mid\bar{\mathbf{c}}',\xi'_k w_k)$ and similarly $\mathrm{q}(\mathbf{c}|w)=\mathcal{N}(\mathbf{c}\mid\bar{\mathbf{c}},\xi_k w_k)$, we can apply the result from Penny (2001) on the KL divergence between two normal distribution, which yields, after simplification:

$$-\int \mathrm{q}(\mathbf{c}|w)\ln\frac{\mathrm{p}(\mathbf{c}|w)}{\mathrm{q}(\mathbf{c}|w)}\,\mathrm{d}\mathbf{c}=\frac{1}{2}\left(\ln\frac{\xi'}{\xi}+\frac{\xi}{\xi'}+\frac{1}{\xi' w}(\bar{\mathbf{c}}-\bar{\mathbf{c}}')^{\mathrm{T}}(\bar{\mathbf{c}}-\bar{\mathbf{c}}')-1\right) \tag{A.83}$$

Computing the expectation of the above with respect to $\mathrm{q}(w)$ is straightforward and only requires showing the following:

$$\left\langle\frac{1}{w}\right\rangle_{\mathrm{q}(w)}=\int \mathrm{IGa}(w|\alpha,\beta)\frac{1}{w}\,\mathrm{d}w \tag{A.84}$$

$$=\int \frac{\beta^{\alpha}}{\Gamma(\alpha)}w^{-\alpha-1}e^{-\frac{\beta}{w}}\frac{1}{w}\,\mathrm{d}w \tag{A.85}$$

$$=\int \frac{\beta^{\alpha}}{\Gamma(\alpha)}\lambda^{\alpha+1}e^{-\beta\lambda}\lambda\lambda^{-2}\,\mathrm{d}\lambda \tag{A.86}$$

$$=\int \mathrm{Ga}(\lambda|\alpha,\beta)\lambda\,\mathrm{d}w \tag{A.87}$$

$$=\frac{\alpha}{\beta} \tag{A.88}$$

Putting Equations (A.82), (A.83) and (A.88) back into Equation (A.77) gives:

$$-\left\langle\ln\frac{\mathrm{p}(\mathbf{c}|w)\mathrm{p}(w)}{\mathrm{q}(\mathbf{c}|w)\mathrm{q}(w)}\right\rangle_{\mathrm{q}(\mathbf{c},w)}=$$
$$\alpha\ln\beta-\alpha'\ln\beta'-\ln\frac{\Gamma(\alpha)}{\Gamma(\alpha')}+(\alpha-\alpha')\left[\Psi(\alpha)-\ln\beta\right]-\alpha\left(1-\frac{\beta'}{\beta}\right) \tag{A.89}$$
$$+\frac{1}{2}\left(\ln\frac{\xi'}{\xi}+\frac{\xi}{\xi'}+\frac{1}{\xi'}\frac{\alpha}{\beta}(\bar{\mathbf{c}}-\bar{\mathbf{c}}')^{\mathrm{T}}(\bar{\mathbf{c}}-\bar{\mathbf{c}}')-1\right)$$

A.2.3 Result

After restoring the k index and substituting (A.76) and (A.89) into (A.74), we obtain the final result:

$$
\begin{aligned}
-\left\langle \ln \frac{p(\mathbf{x})}{q(\mathbf{x})} \right\rangle_{q(\mathbf{x})} = \sum_{k=1}^{N} \Bigg(& \gamma_k \ln \delta_k - \gamma'_k \ln \delta'_k - \ln \frac{\Gamma(\gamma_k)}{\Gamma(\gamma'_k)} \\
& + (\gamma_k - \gamma'_k)\left[\Psi(\gamma_k) - \ln \delta_k\right] - \gamma_k \left(1 - \frac{\delta'_k}{\delta_k}\right) \\
& + \alpha_k \ln \beta_k - \alpha'_k \ln \beta'_k - \ln \frac{\Gamma(\alpha_k)}{\Gamma(\alpha'_k)} \\
& + (\alpha_k - \alpha'_k)\left[\Psi(\alpha_k) - \ln \beta_k\right] - \alpha_k \left(1 - \frac{\beta'_k}{\beta_k}\right) \\
& + \frac{1}{2}\left(\ln \frac{\xi'_k}{\xi_k} + \frac{\xi_k}{\xi'_k} + \frac{1}{\xi'_k}\frac{\alpha_k}{\beta_k}(\bar{\mathbf{c}}_k - \bar{\mathbf{c}}'_k)^{\mathrm{T}}(\bar{\mathbf{c}}_k - \bar{\mathbf{c}}'_k) \right) \\
& \Bigg) - \frac{N}{2}
\end{aligned}
\tag{A.90}
$$

A.2.4 Gradient

Following are the partial derivatives of the previous expression with respect to each parameters.

$$\frac{\partial KL_{\mathrm{p}}}{\partial \bar{\mathbf{c}}_k} = \frac{1}{\xi'_k}\frac{\alpha_k}{\beta_k}(\bar{\mathbf{c}}_k - \bar{\mathbf{c}}'_k) \tag{A.91}$$

$$\frac{\partial KL_{\mathrm{p}}}{\partial \xi_k} = \frac{1}{2}\left(\frac{1}{\xi'_k} - \frac{1}{\xi_k}\right) \tag{A.92}$$

$$\frac{\partial KL_{\mathrm{p}}}{\partial \alpha_k} = (\alpha_k - \alpha'_k)\frac{\partial \Psi(\alpha_k)}{\partial \alpha_k} - \left(1 - \frac{\beta'_k}{\beta_k}\right) + \frac{1}{2\xi'_k\beta_k}(\bar{\mathbf{c}}_k - \bar{\mathbf{c}}'_k)^{\mathrm{T}}(\bar{\mathbf{c}}_k - \bar{\mathbf{c}}'_k) \tag{A.93}$$

$$\frac{\partial KL_{\mathrm{p}}}{\partial \beta_k} = \frac{\alpha'_k}{\beta_k} - \frac{\alpha_k\beta'_k}{\beta_k^2} - \frac{1}{2\xi'_k}\frac{\alpha_k}{\beta_k^2}(\bar{\mathbf{c}}_k - \bar{\mathbf{c}}'_k)^{\mathrm{T}}(\bar{\mathbf{c}}_k - \bar{\mathbf{c}}'_k) \tag{A.94}$$

$$\frac{\partial KL_{\mathrm{p}}}{\partial \gamma_k} = (\gamma_k - \gamma'_k)\frac{\partial \Psi(\gamma_k)}{\partial \gamma_k} - \left(1 - \frac{\delta'_k}{\delta_k}\right) \tag{A.95}$$

$$\frac{\partial KL_{\mathrm{p}}}{\partial \delta_k} = \frac{\gamma'_k}{\delta_k} - \frac{\gamma_k\delta'_k}{\delta_k^2} \tag{A.96}$$

B Data assimilation framework

This appendix details some of the technical aspects associated with the implementation of the data assimilation framework. Note that the framework described provides support for general data assimilation as discussed in Chapters 2 to 5. The particular implementation of the rainfall model discussed in Chapters 6 and 7 is not covered here.

B.1 General overview

The data assimilation framework comes as a set of libraries which can be used to create custom data assimilation experiments. In total the framework amounts to about 5600 lines of code in its original version, which is reasonably small for a library (as a comparison, the extension to the rainfall model of the framework to the rainfall nowcasting problem involves about 16000 lines of code). The organisation of the libraries is discussed below.

B.1.1 Overview of libraries

The data assimilation framework relies on the Matpack library (Gammel, 2005) to provide the numerical foundations such as linear algebra, differential equations solvers, random number generators, etc. Although Matpack is a very extensive library and provides support for a great number

of numerical applications, the support for linear algebra remains fairly limited. In consequence, we developed an extension to Matpack (MPEXT) aimed at improving support for linear algebra. Note that Matpack can be interfaced with the BLAS library, which is considered to be the standard for vector and matrix computation. The overall architecture of the framework is summarised on Figure B.1.

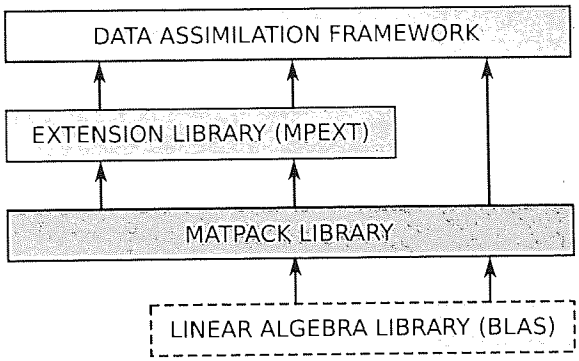


Figure B.1: Data assimilation framework – Software architecture

B.1.2 The extension library

When writing the extension library (MPEXT), care was taken to provide a syntax and functionalities which would be close enough to those found in MATLAB®. The following features were implemented as part of the MPEXT library.

Matrix/Vector generation

- methods to generate matrices and vectors filled with zeros, ones, or a custom number
- methods to generate diagonal matrices
- methods to generate uniform and Gaussian random matrices/vectors

General Matrix/Vector functions

- element-wise absolute value, element-wise modulus
- minimum/maximum of matrix elements (for each row or column)
- Cholesky decomposition of matrix
- mean and autocovariance of vector elements
- sum, weighted sum and mean of rows/columns of matrix
- weighted covariance and covariance of rows/columns in matrix

- cumulative sum of vector elements, distance matrix between two vectors of scalars, squared error between two matrices
- length of vector, size of matrices, number of rows/columns in matrix

Product functions

- Vector $\times$ Vector $\rightarrow$ scalar (inner product)
- Vector $\times$ Vector $\rightarrow$ Matrix (outer product)
- Matrix $\times$ Vector $\rightarrow$ Vector
- Vector $\times$ Vector $\rightarrow$ Vector, element-wise product
- Matrix $\times$ Matrix $\rightarrow$ Matrix, element-wise (Schur) product
- Vector $\div$ Vector $\rightarrow$ Vector, element-wise division
- Matrix $\div$ Matrix $\rightarrow$ Matrix, element-wise division

Element functions

- Range extraction from vector/matrix (subvector/submatrix)
- Replication of vector into matrix
- Reshaping of vector into matrix
- Aggregation of two vectors into a vector, aggregation of two matrices into a matrix (by rows or columns)
- Setting matrix row/column from vector
- Extract diagonal of matrix as a vector, create diagonal matrix from a vector

Input/Output functions

- Print matrix/vector on screen
- Write matrix/vector to binary file, read matrix/vector from binary file
- Write matrix/vector to CSV file, read matrix/vector from CSV file [1]

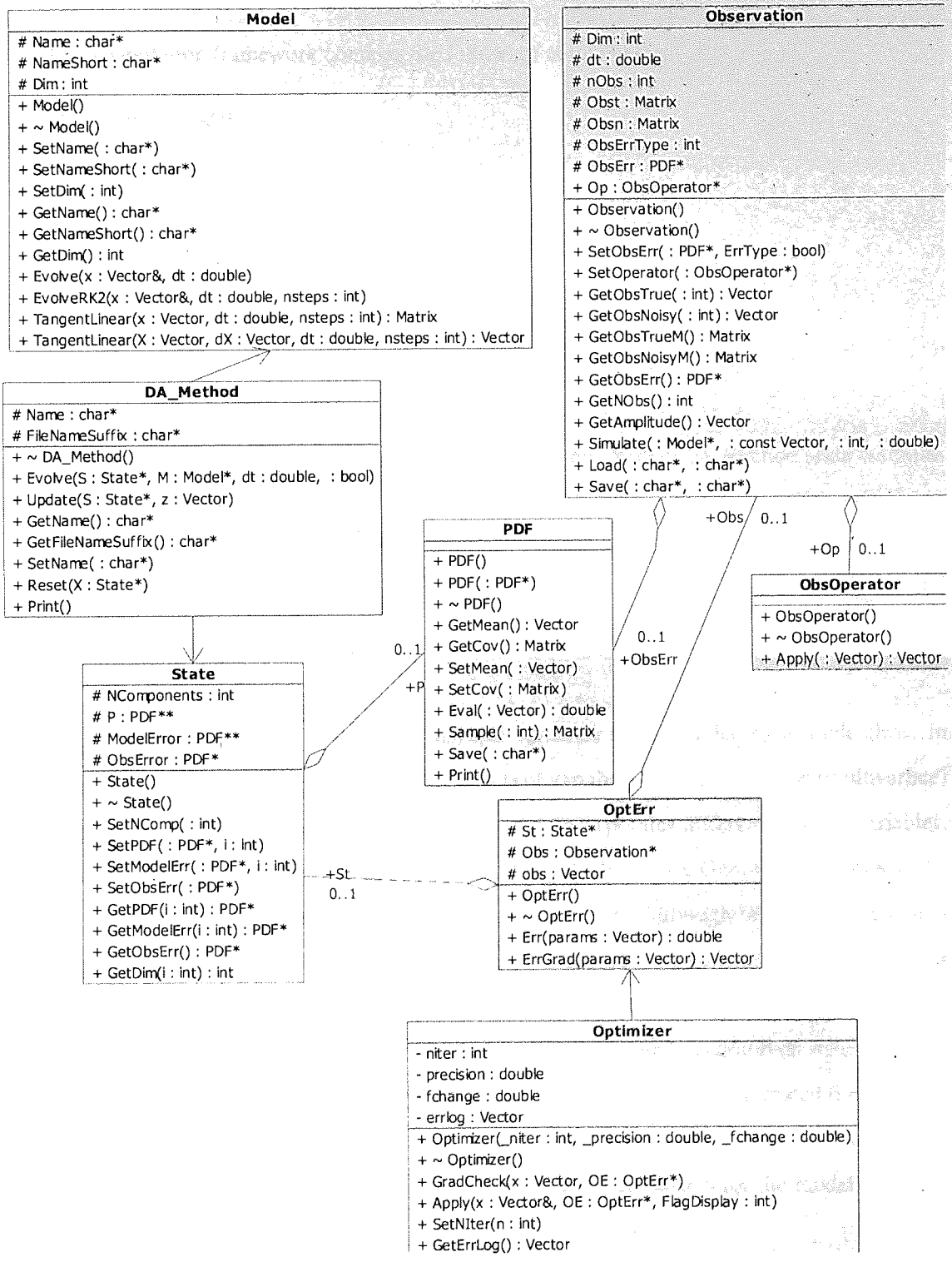


Figure B.2: Data assimilation framework – Top level class diagram

B.2 Components of the data assimilation framework

B.2.1 Overview

The data assimilation framework contains the following components:

- Dynamical models
- Observations
- State
- Data assimilation methods
- Probability distributions
- Optimisation and error functions

Figure B.2 shows the class diagram for these components. `Model`, `DA_Method` (data assimilation method), `PDF` (probability density function), `ObsOperator` (observation operator) and `OptErr` (error function for use in optimisation method) are interfaces. The `State`, `Observation` and `Optimizer` classes are standard classes[2].

B.2.2 State

The `State` has been written so as to allow multiple variables to be handled by a single class. In particular, it supports multiple "components", i.e. sets of variables having a common (multivariate) distribution. This is useful for models in which the state incorporates different types of variables, some of which have a Gaussian distribution, some others of which have Gamma distributions, etc.

The `State` also handles observation error and model error. Although one would argue that these belong to the `Model` and `Observation` interfaces respectively, two reasons motivate their implementation in the `State` class. First, one might want to estimate the parameters of errors along with the state. In such situation, errors can effectively be treated as additional state variables. A second reason for keeping the errors in the state even when they are not estimated (i.e. they are fixed) is that it is more practical to pass a reference to a single object (the state) to models and observation operators rather than having to also include separate objects for the model error and the observation error.

[1] the CSV (comma separated values) file format is supported by many application including Microsoft Excel and Matlab.

[2] The `Optimizer` ought to be rewritten as an interface to allow different optimisation methods to be used within the framework. At the moment, the `Optimizer` class implements the Scaled Conjugate Gradient algorithm

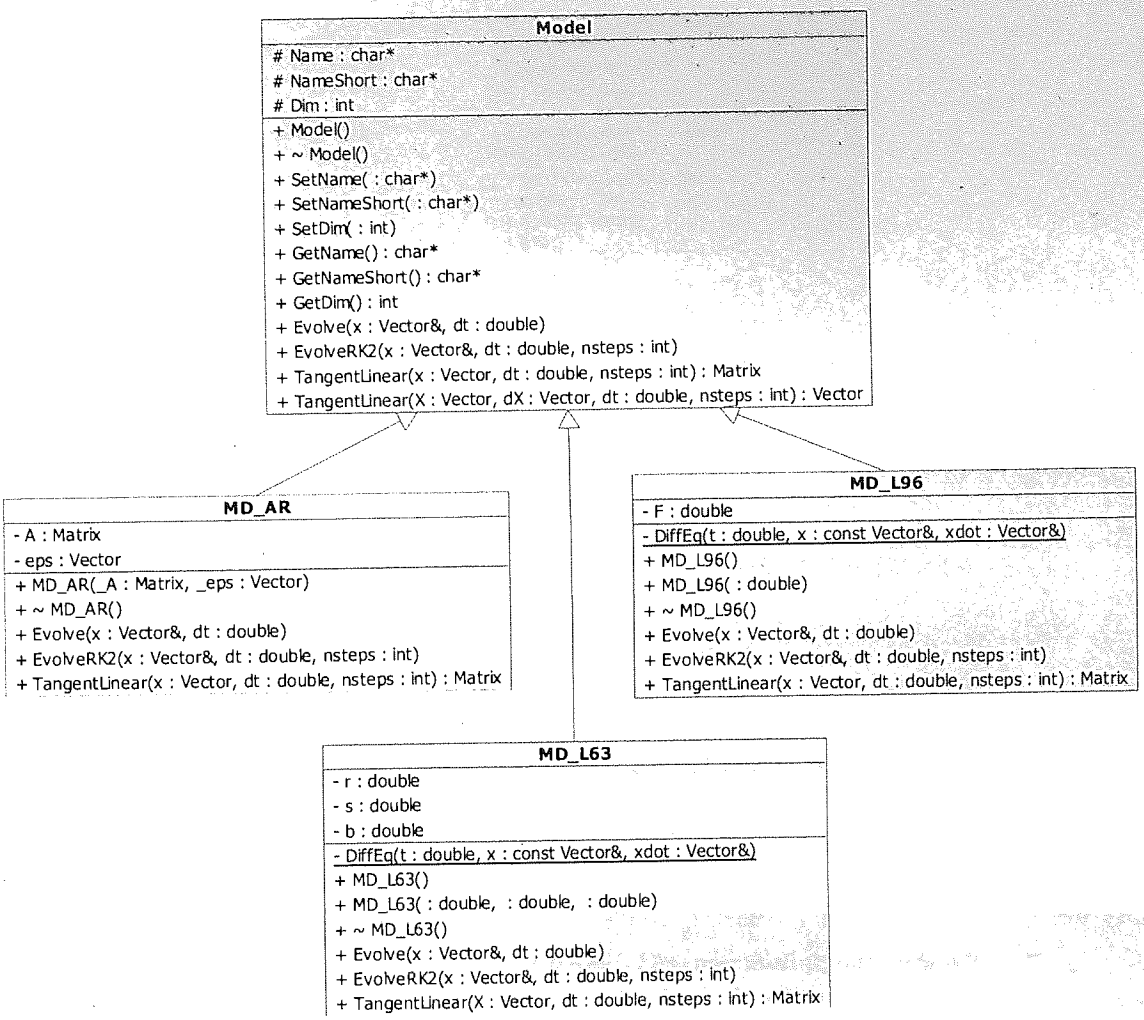


Figure B.3: Data assimilation framework – Dynamical models class diagram

B.2.3 Dynamic models

A model needs to provide methods which propagate the state forward in time and compute the tangent linear operator about a given state when necessary. Three models have been implemented within the framework: the Lorenz-63 system, the Lorenz-96 system and an simple autoregressive model used for testing purposes. There is no model error involved in the `Model` (it is handled by the data assimilation method `DA_Method`). Figure B.3 provides a class diagram of the models implemented.

B.2.4 Data assimilation methods

Data assimilation methods typically need to be able to propagate the state forward in time using the `Model` and update the `State` given the `Observation`. Methods implemented include the Extended Kalman Filter, the Ensemble Kalman Filter, the Particle Filter and 4D VAR (both strong and weak constraint). Two methods which are not listed here nor discussed in this work have also been

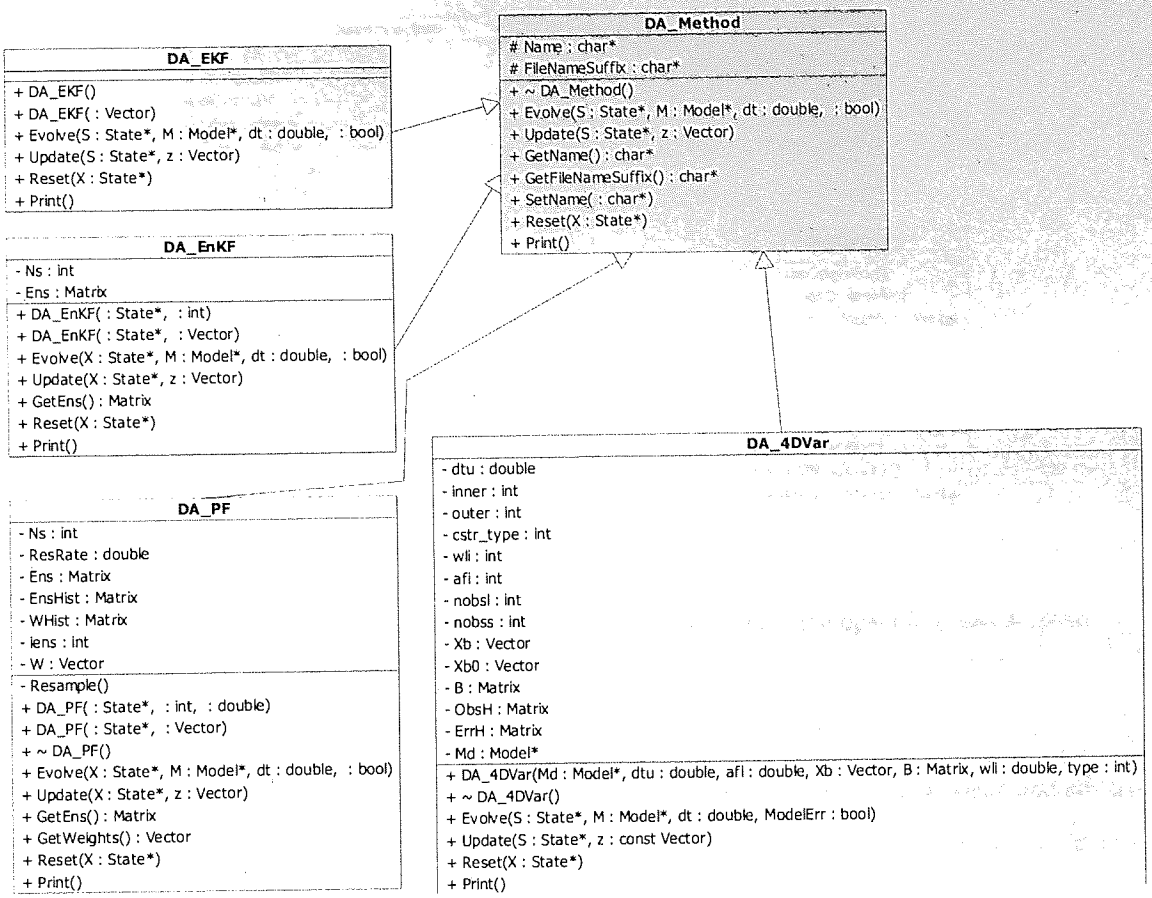


Figure B.4: Data assimilation framework – Data assimilation class diagram

added later on: the Extended Kalman Smoother and the Ensemble Kalman Smoother.

The EKF and 4DVAR rely on the `Model` for the computation of the tangent linear model. Both the strong and weak constraints of 4D VAR are handled within a single class. A better design would propably separate these two classes and have one inherit (in an object-oriented sense) from the other. The Particle Filter uses Systematic Resampling, however it should be extended to allow other resampling schemes to be used.

Figure B.4 provides a class diagram of the data assimilation methods implemented (restricted to those discussed in this work).

B.2.5 Observations

The `Observation` class is responsible for handling the observations. The tasks provided include simulating observations (using a given `Model`), saving to and loading from files, accessing a particular observation ("true" or "noisy"), etc. The observation operator (`ObsOperator`) maps the `State` to observation space. As shown on the class diagram on Figure B.5, only the simplest observation operator has been implemented in the framework, i.e. the direct (or identity) operator (`OP_Identity`) for use when the state is observed directly.

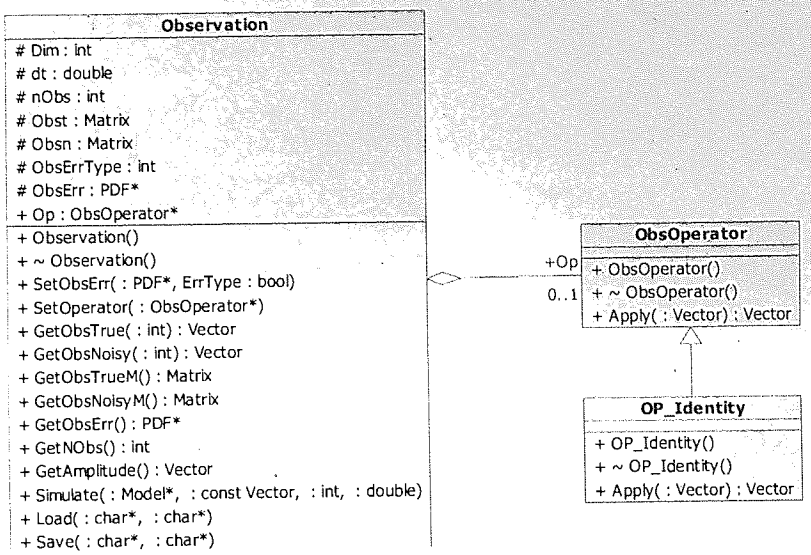


Figure B.5: Data assimilation framework – Observation and observation operators class diagram

B.2.6 Probability density functions

A probability density function (`PDF`) must allow for probabilities to be evaluated, mean and covariance to be computed and samples to be drawn. Only Gaussian distributions have been considered, both generic (`PDF_Gauss`) and diagonal (`PDF_GaussDiag`). Further non-Gaussian distributions have been implemented which are not shown here (namely the Gamma and Inverse-Gamma distributions). Figure B.6 gives the class diagram for the probability density functions.

B.2.7 Optimisation and error functions

The `Optimizer` class implements a Scaled Conjugate Gradient algorithm. It has been adapted from Netlab's implementation of `scg()` (Nabney, 2001). Typically, this class requires an error class (`OptErr`) to provide the error function and its gradient. It also provides the ability to test the gradient validity by comparison with a finite difference approximation (adapted from Netlab's `gradchek()`). Two error functions have been implemented, which correspond to the cost functions in 4D VAR strong constraint and weak constraint. The class diagram for the `Optimizer` and error functions is given on Figure B.7.

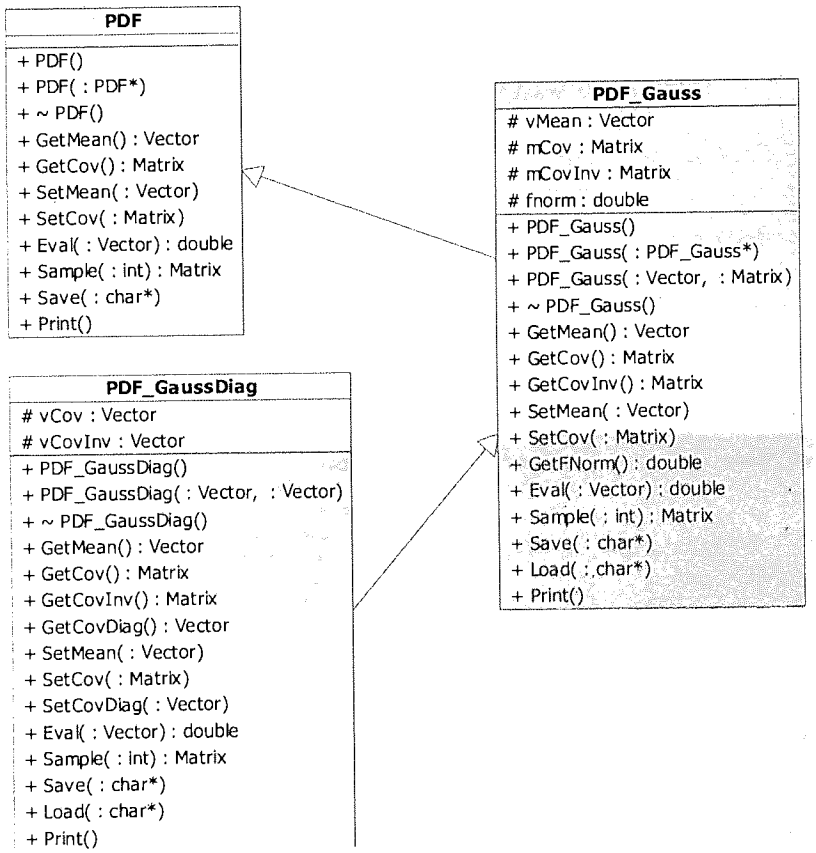


Figure B.6: Data assimilation framework – Probability density functions class diagram

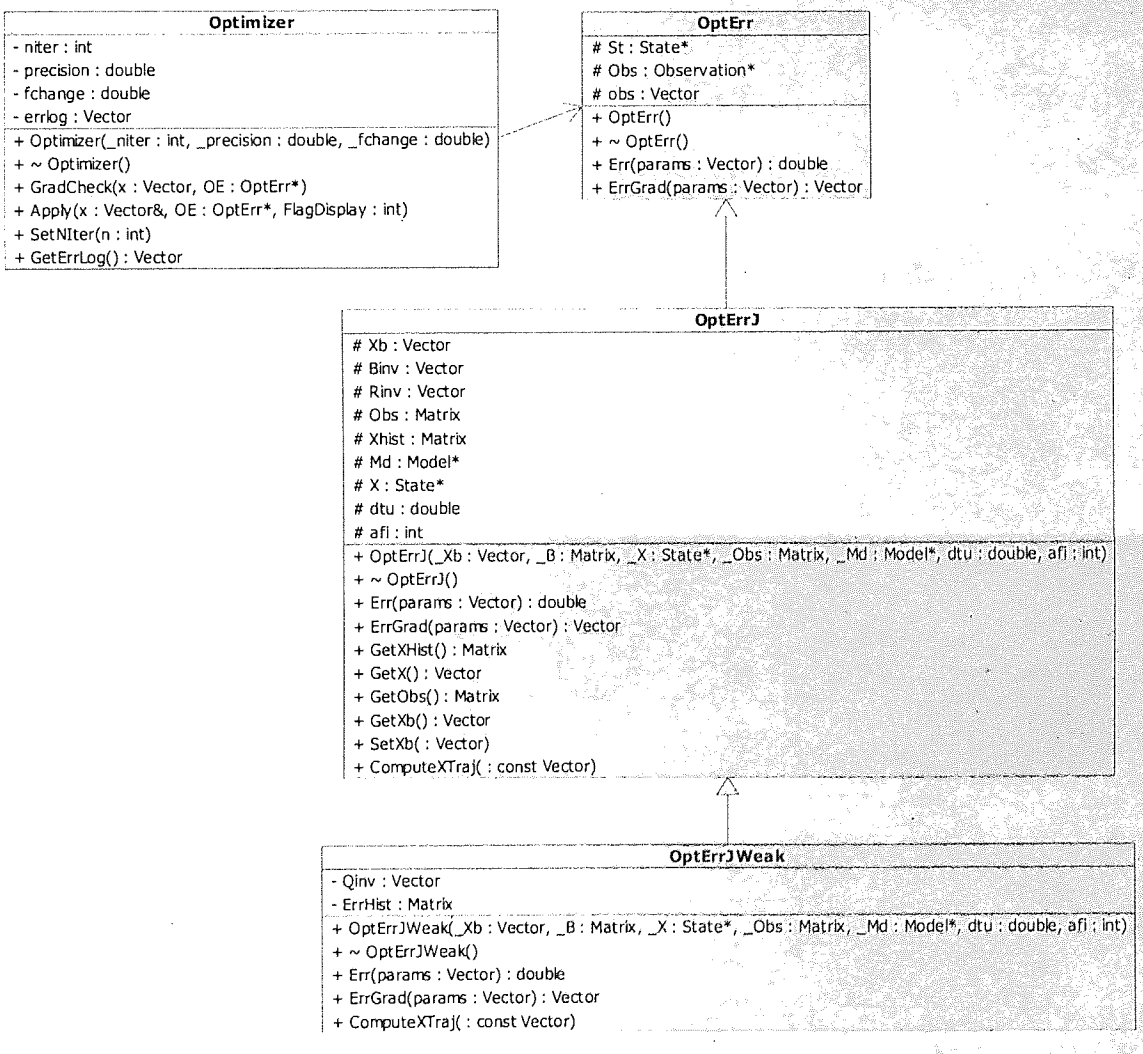


Figure B.7: Data assimilation framework – Optimisation and error functions class diagram