University of Crete
School of Sciences and Engineering
Department of Mathematics and Applied Mathematics

Master's Diploma Thesis

Data Assimilation Methods

Georgia Sfakianaki

Supervisor: Prof. Michael Plexousakis

Assessment Committee

Prof. Dimitrios Bagkavos
Prof. Georgios Kossioris
Prof. Michael Plexousakis

M.Sc. Program Applied and Computational Mathematics
Specialization: Scientific Computing

Heraklion, June 2016

Acknowledgments

This thesis is the final step of my studies for the acquisition of Master's degree in Applied and Computational Mathematics from the University of Crete, Greece. During the course of my work, there have been difficulties and challenges, however the journey has left me with more experience and deeper appreciation for the subject.

First and foremost, I express my gratitude and appreciation to my thesis supervisor, Prof. Michael Plexousakis, for his guidance and support. He was always available to answer my questions and provide counsel in all stages of my work. His advice throughout my studies has been instrumental, and his influence on the path that I have followed has been essential.

I also thank Prof. Georgios Kossioris for his useful insights and all the encouragement during the past months. Our discussions are invaluable to me and I deeply appreciate his availability and interest.

During my work on this thesis, I had the opportunity to meet and discuss with Prof. Roland Potthast, to whom I express my vast respect and gratitude. His dedicating time from his short stay at the University of Crete to our discussions meant a great deal to me and helped me in more than one way. Results for the Ensemble Kalman Filter algorithm are product of our discussions.

I also thank the assessment committee, composed by Prof. Dimitrios Bagkavos, Prof. Georgios Kossioris and Prof. Michael Plexousakis, for taking the time to read this thesis and for providing useful suggestions.

Last, but certainly not least, a special "thank you" to my family, for their endless support and encouragement.

Περίληψη

Μέθοδοι αφομοίωσης δεδομένων (Data assimilation methods) έχουν χρησιμοποιηθεί εκτενώς τις περασμένες δεκαετίες σε επιστήμες όπως για παράδειγμα τη μετεωρολογία, την υδρολογία και την ωκεανογραφία. Γενικά, η αφομοίωση δεδομένων είναι μία τεχνική η οποία συνδυάζει προηγούμενη γνώση ενός συστήματος, υπό τη μορφή ενός αριθμητικού μοντέλου, με νέα πληροφορία για την κατάσταση του συστήματος, υπό την μορφή παρατηρήσεων. Οι μέθοδοι αυτές χωρίζονται σε δύο μεγάλες κατηγορίες, ονομαστικά, τις *μεταβολικές* (variational) και τις *ακολουθιακές* (sequential).

Οι μεταβολικές μέθοδοι βασίζονται στην θεωρία βέλτιστου ελέγχου και σκοπός σε αυτές είναι η ελαχιστοποίηση ενός συναρτησιακού που μετράει την απόσταση των δεδομένων από το μοντέλο. Από την άλλη πλευρά, το κύριο χαρακτηριστικό των ακολουθιακών μεθόδων είναι ότι οι παρατηρήσεις αφομοιώνονται αμέσως μόλις γίνονται διαθέσιμες. Στην παρούσα εργασία μελετάμε και παρουσιάζουμε δύο μεθόδους αφομοίωσης δεδομένων από κάθε κατηγορία. Από την οικογένεια των μεταβολικών μεθόδων μελετάμε τη *Μεταβολική Αφομοίωση Δεδομένων στις Τρεις Διαστάσεις* (Three-dimensional Variational Assimilation ή 3D-Var) και τη *Μεταβολική Αφομοίωση Δεδομένων στις Τέσσερις Διαστάσεις* (Four-dimensional Variational Assimilation ή 4D-Var). Η πιο διαδεδομένη μέθοδος στη δεύτερη κατηγορία, είναι το *Φίλτρο Kalman* (Kalman Filter). Μελετάμε δύο εκδοχές του φίλτρου Kalman, ονομαστικά, το *Extended Kalman Filter* και το *Ensemble Kalman Filter*.

Στην εργασία αυτή, μελετάμε την εφαρμογή των παραπανω μεθόδων αφομοίωσης δεδομένων στην αριθμητική πρόβλεψη του καιρού (numerical weather prediction). Σε αυτό το πλαίσιο, η αφομοίωση δεδομένων χρησιμοποιείται για την ανάλυση της τρέχουσας κατάστασης της ατμόσφαιρας, ώστε να καθοριστούν οι κατάλληλες αρχικές συνθήκες που θα χρειαστούν για τη βελτίωση μίας μεταγενέστερης αριθμητικής πρόγνωσης του καιρού. Θεωρούμε λοιπόν το μη-γραμμικό δυναμικό σύστημα του Lorenz (Lorenz 1996) και παρουσιάζουμε αποτελέσματα της υλοποίησης των 3D-Var και Ensemble Kalman Filter σε αυτό.

Παρουσιάζουμε επίσης, το Weather Research and Forecasting (WRF) model το οποίο αποτελεί ένα από τα πιο προηγμένα μοντέλα αριθμητικής πρόβλεψης του καιρού και είναι ένα ελεύθερο λογισμικό που χρησιμοποιείται τόσο για ερευνητικούς, όσο και για επιχειρησιακούς σκοπούς. Εστιάζουμε στο πακέτο WRFDA που είναι ενσωματωμένο σε αυτό, και παρέχει δυνατότητες αφομοίωσης μετεωρολογικών δεδομένων και συγκεκριμένα, περιλαμβάνει την υλοποίηση των 3D-Var και 4D-Var, καθώς και μία υβριδική μέθοδο που συνδυάζει τις μεταβολικές και ακολουθιακές μεθόδους.

Abstract

Data Assimilation (DA) methods have been extensively used the past decades in many fields of science, among which meteorology, hydrology and oceanography, to mention just a few. Generally speaking, data assimilation is a technique to combine past knowledge of the system, in the form of a numerical model, and information about the system's state, in the form of observations. Two main categories of DA methods can be recognized, variational and sequential.

Variational methods are based on optimal control theory and the aim is to minimize a given cost function that measures the model-to-data misfit. In sequential methods, on the other hand, observations are assimilated as soon as they become available. In this thesis we study and present two methods from each family. The Three-Dimensional Variational assimilation (3D-Var) and the Four-Dimensional Variational assimilation methods (4D-Var) fall into the first category. The most well-known DA method in the sequential family is the Kalman Filtering, and we focus on two of its variants, the Extended and the Ensemble Kalman Filtering.

The methods presented in our work are viewed under the prism of numerical weather prediction. In this context, data assimilation is used to produce an analysis of the current state of the atmosphere to be used as initial conditions in a subsequent weather forecast, leading to more accurate predictions. We provide implementations and numerical results for the 3D-Var and Ensemble Kalman Filtering methods applied to the Lorenz-96 model.

In the frame of weather forecasting, we also present the Weather Research and Forecasting (WRF) model which is a state-of-the-art atmospheric modeling system designed for numerical weather prediction and is currently being used in operational centers. We focus on the WRF Data Assimilation (WRFDA) module that includes implementations for the 3D-Var and 4D-Var methods, as well as a hybrid scheme between the variational and ensemble assimilation methods.

Contents

Abbreviations

BLUE	Best Linear Unbiased Estimate
DA	Data Assimilation
EnKF	Ensemble Kalman Filter
EnSRF	Ensemble Square Root Filter
EKF	Extended Kalman Filter
KF	Kalman Filter
MM5	NCAR Mesoscale Model modeling system
MSE	Mean Square Error
NCAR	National Center for Atmospheric Research
NCEP	National Centers for Environmental Prediction
NOAA	National Oceanic and Atmospheric Administration
NWP	Numerical Weather Prediction
OI	Optimal Interpolation
RMSE	Root Mean Square Error
TLM	Tangent Linear Model
WRF	Weather Research and Forecasting Model
WRF-ARW	Advanced Research WRF
WRFDA	Data Assimilation Module of the WRF-Model
3D-Var	Three-Dimensional Variational Analysis
4D-Var	Four-Dimensional Variational Analysis

List of Symbols

Symbol	Description
$\mathbf{B}$	background error covariance matrix
$\mathbf{d}$	innovation or observational increments vector
$E\{\cdot\}$	expected value
F	forcing term
H	nonlinear observation operator
$\mathbf{H}$	linear observation operator matrix
$\mathbf{I}$	identity matrix
J	cost function
$\mathbf{K}$	Kalman gain matrix
$\mathbf{L}$	TLM matrix of the model operator M
$\mathcal{L}$	likelihood function
ℓ	additive covariance inflation factor
M	nonlinear model operator
p	probability, distribution function
$\mathbf{P}^a$	analysis error covariance matrix
$\mathbf{P}^f$	forecast error covariance matrix
$\mathbf{Q}$	forecast model error covariance
$\mathbf{R}$	observations error covariance matrix
T	temperature
W	optimal weight
$\mathbf{W}$	weight matrix
$\mathbf{x}$	model state vector
X	model space
$\mathbf{x}^a$	analysis state
$\overline{\mathbf{x}}^a$	analysis ensemble mean
$\mathbf{X}^a$	analysis ensemble perturbations matrix
$\mathbf{x}^b$	background field or "first guess"
$\mathbf{x}^f$	forecast state
$\overline{\mathbf{x}}^f$	forecast or background ensemble mean
$\mathbf{X}^f$	forecast or background ensemble perturbations matrix
$\mathbf{x}^t$	true model state
Y	observation space
$\mathbf{y}^o$	observation vector
$\overline{\mathbf{y}}^f$	forecast or background ensemble observation mean
$\mathbf{Y}^f$	forecast or background ensemble observation perturbations
γ	multiplicative covariance inflation factor
ε^a	analysis error
ε^b	background error
ε^f	forecast error
ε^o	observational error
σ	standard deviation
σ^2	variance

List of Figures

Chapter 1

Introduction

Data Assimilation (DA) is a technique combining past knowledge of a system, in the form of a numerical model, and new information about the system, in the form of observations. The idea of Data Assimilation was firstly introduced in the form of the least-squares method by Carl Friedrich Gauss, in 1801, who managed to predict with impressive accuracy the position and date of the reappearance of the planetoid Ceres[1].

Initially, DA methods arose from the need to improve weather forecasting in meteorology by determining the initial conditions needed for subsequent computer forecasts. Today, there are many other fields in which DA methods find application: in oceanography, in hydrology, in seismology, in nuclear fusion, in medicine, etc. Except from finding initial conditions for predictions, they can also be used for calibration and validation, for designing of the observation system, monitoring and assessment, as well as for reanalysis and better understanding of a system (model and data errors, physical process interactions, parameters, etc).

Assimilation methods are divided into two classes: *sequential* and *variational* assimilation methods. Sequential DA takes into account observations created in the past until the time of the analysis, which means that the observations are assimilated as soon as they become available. This approach corresponds to the real-time assimilation systems. On the other hand, variational DA is based on the optimal control theory. In this approach we seek a state that best fits the data within an assimilation time-window. Therefore, optimization is performed on unknown parameters by minimizing a given cost function that measures the model-to-data misfit.

There is also the so-called *ensemble modeling* or *ensemble forecasting*. Many forecasts are run with slightly perturbed initial conditions (or with different models) and an average, or ensemble mean of the different forecasts is being created. The ensemble techniques provide information about the uncertainty in the initial conditions, since a large ensemble of forecasts is available. In this thesis, we present DA methods that are extensively used in the fields described above, but we focus on the Numerical Weather Prediction (NWP) problem and how weather forecasting can be optimized using assimilation methods.

[1] Giuseppe Piazzi discovered Ceres on January 1, 1801 and made 19 observations over 42 days, before the object disappeared in the vicinity of the Sun. Gauss was able to calculate the orbit of Ceres using only three of Piazzi's observations and hence, initiated the least-squares theory.

1.1 Data Assimilation for NWP

The intention of NWP is to predict the weather using the mathematical models of the atmosphere and the ocean. It is classically viewed as an initial-value problem[2], according to which the governing equations of geophysical fluid dynamics are integrated forward in time from a given estimate of the state of the atmosphere at some initial time. In order to make a valid forecast the numerical model must be a realistic representation of the atmosphere and also the initial conditions must be known accurately.

One of the problems encountered in the procedure of forecasting is the nature of the system describing the weather. In 1963, the famous meteorologist Edward N. Lorenz in [24] concluded that the weather exhibits chaotic behavior, i.e., small errors in the initial conditions of a forecast grow rapidly affecting the predictability. In order to overcome this problem, data assimilation techniques have been developed and employed in the estimation of suitable initial conditions. The aim of this thesis is the study and implementation of such methods, since the problem of determining suitable initial conditions for a forecast model is very important and complex, and has become a science itself.

Our main goal is to be able to produce an accurate estimate of the true state of the atmosphere at a given time. This estimate is called *analysis* state. The basic information that can be used to produce the analysis is a collection of observations[3] of the true state. If the model state is over-determined by the observations, then the analysis reduces to an interpolation problem. Most of the time, the analysis problem is under-determined because the data are sparse and usually indirectly related to the model variables. In order to make it a well-posed problem, it is necessary to add some *background* information in the form of an *a-priori* model state estimate. The background information initially was a climatology, but as the forecasting became better, *a short-range forecast is chosen as the first-guess* in operational data assimilation systems or analysis cycles. A typical 6-hour data assimilation cycle, performed four times a day for a global model, needs a 6-hour forecast as background field. The definition of data assimilation for the NWP problem follows.

> *Data assimilation is an analysis technique in which observations that are distributed in time, are incorporated into the model state of a dynamical numerical model to produce an estimate of the true state of the atmosphere as accurately as possible.*

Model and Observations

Let $X = \mathbb{R}^n$ be the *model space* and let $\mathbf{x} \in X$ the model's state vector. We shall denote by $\mathbf{x}^t \in X$ the *truth* or the *true state* of the system. Our goal is to produce an estimate $\mathbf{x}^a$ (where a stands for the *analysis*) of the true state $\mathbf{x}^t$. As explained earlier, the outcome of a previous forecast can serve as the background information in the new analysis, denoted by $\mathbf{x}^b$, where b refers to *background*[4]. The evolution of an atmospheric or oceanic system from time t_{i-1} to time t_i is governed by an equation of the form

$$\mathbf{x}^b(t_i) = M_{i-1}[\mathbf{x}^a(t_{i-1})],$$

[2] A system of differential equations is referred to as an initial value problem when the solution depends not only on boundary conditions, but also on the values of the unknown fields or their derivatives at some initial time.

[3] Many types of data are currently available, such as satellite and radar observations, but, usually, they do not measure directly the variables of the model (temperature, wind, surface pressure, moisture, etc).

[4] The notation $\mathbf{x}^b$ is meant to be the best estimate of the current state $\mathbf{x}(t)$ prior to using the observations at time t. In the methods presented in the sequel, $\mathbf{x}^b$ will commonly be the result of a previous short-range forecast (or even, the result of a previous assimilation) hence, the notations $\mathbf{x}^b$ or $\mathbf{x}^f$ will be used correspondingly.

where $M : X \to X$ is the (nonlinear) *model operator* that represents the dynamical model and is a function from the model space into itself.

Let $Y = \mathbb{R}^p$ be the *observation space*. Observations $\mathbf{y}^o \in Y$ become available through the *observation model*

$$\mathbf{y}^o = H(\mathbf{x}^t) + \boldsymbol{\varepsilon}_o,$$

where $H : X \to Y$ is the (nonlinear) *observation operator*, a function from the model space to the observation space, with $\boldsymbol{\varepsilon}_o$ be the observation error.

The background, the analysis and the observations contain errors defined, respectively, as

$$\begin{aligned}\varepsilon_b &= \mathbf{x}^b - \mathbf{x}^t,\\ \varepsilon_a &= \mathbf{x}^a - \mathbf{x}^t,\\ \varepsilon_o &= \mathbf{y}^o - H(\mathbf{x}^t).\end{aligned}$$

Since the true state $\mathbf{x}^t$ is unknown, we don't know the errors of the background and the observations. However, we can make some assumptions about their statistical properties. We assume that the errors are having mean zero, i.e.,

$$E\{\varepsilon_b\} = E\{\varepsilon_a\} = E\{\varepsilon_o\} = 0.$$

Therefore, their respective error covariance matrices are defined as

$$\mathbf{B} = E\{\varepsilon_b \varepsilon_b^T\}, \quad \mathbf{P}^a = E\{\varepsilon_a \varepsilon_a^T\}, \quad \mathbf{R} = E\{\varepsilon_o \varepsilon_o^T\}.$$

and we also assume that the background and the observation errors are uncorrelated, i.e., $E\{\varepsilon_o \varepsilon_b^T\} = 0$.

In order to incorporate the available observations into our estimate, we compare the observation vector $\mathbf{y}^o$ with the current state estimate $\mathbf{x}^b$. To do so, we define the difference between the observations and the background as the *observational increments* or *innovation vector*

$$\mathbf{d} = \mathbf{y}^o - H(\mathbf{x}^b).$$

The observation operator in general is nonlinear but it can be linearized using a first order Taylor approximation, i.e.,

$$H(\mathbf{x} + \delta\mathbf{x}) = H(\mathbf{x}) + \mathbf{H}\delta\mathbf{x},$$

where $\mathbf{H}$ is a $p \times n$ matrix whose elements are the first-order partial derivatives $h_{i,j} = \partial \mathbf{H}_i / \partial \mathbf{x}_j$. Then, the innovation vector can be written in the form

$$\mathbf{d} = \mathbf{y}_o - H(\mathbf{x}^b) = \mathbf{y}_o - H(\mathbf{x}^t + (\mathbf{x}^b - \mathbf{x}^t)) = \varepsilon_o - \mathbf{H}\varepsilon_b.$$

It will become evident in the next chapter that, the analysis $\mathbf{x}^a$ is obtained by summing the background and the innovation multiplied by a weight $\mathbf{W}$, which can be determined by using the estimated statistical error covariances of the forecast and the observations. Hence, the analysis is

$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{W}[\mathbf{y}^o - H(\mathbf{x}^b)] \quad \text{or} \quad \mathbf{x}^a = \mathbf{x}^b + \mathbf{W}\mathbf{d}.$$

Subsequently, we shall prove step by step that, under the statistical assumptions we have made so far, $\mathbf{W}$ is the *optimal weight matrix* given by

$$\mathbf{W} = \mathbf{B}\mathbf{H}^T[\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T]^{-1},$$

which provides the optimal analysis $\mathbf{x}^a$. Having determined the optimal weight matrix, we are able to derive the formula of the analysis error covariance matrix, $\mathbf{P}_a$, namely,

$$\mathbf{P}_a = (\mathbf{I} - \mathbf{W}\mathbf{H})\mathbf{B}.$$

The procedure that we have followed so far is the so-called Optimal Interpolation.

Another approach for the same problem is the Three-Dimensional Variational method (3D-Var), in which, we wish to find the optimal analysis $\mathbf{x}^a$ that minimizes the functional

$$2J(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}^b) + [\mathbf{y}^o - H(\mathbf{x})]^T \mathbf{R}^{-1}[\mathbf{y}^o - H(\mathbf{x})].$$

This functional is defined as the distance between the model state $\mathbf{x}$ and the background $\mathbf{x}^b$, weighted be the inverse of the background error covariance, plus the distance to the observations $\mathbf{y}^o$ weighted by the inverse of the observation error covariance.

The minimum of the 3D-Var functional is obtained for $\mathbf{x} = \mathbf{x}^a$ such that

$$\nabla_{\mathbf{x}} J(\mathbf{x}^a) = 0.$$

Formally, the 3D-Var analysis is found to be

$$\begin{aligned} & \mathbf{x}^a = \mathbf{x}^b + \mathbf{W}[\mathbf{y}^o - H(\mathbf{x}^b)], \\ \text{where} \quad & \mathbf{W} = [\mathbf{B}^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}]^{-1} \mathbf{H}^T \mathbf{R}^{-1}, \end{aligned}$$

but in practice, the minimum of $J(\mathbf{x})$ is obtained using iterative methods, such as the Steepest Descent, the Conjugate Gradient or the quasi-Newton algorithms.

If the observations are distributed not only in space but also in time (i.e., they are available within a time window), then 3D-Var is generalized to the Four-Dimensional Variational assimilation method or 4D-Var, in which the minimization of the corresponding functional is defined over a four-dimensional space (3 dimensions for space and 1 for time).

The 4D-Var cost function includes a term measuring the distance to the background at the beginning of the time interval, together with a sum accounting for the observations collected over a k-hour time window:

$$J\left[\mathbf{x}(t_0)\right] = \frac{1}{2}\left[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)\right]^T \mathbf{B}_0^{-1}\left[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)\right] + \frac{1}{2}\sum_{i=0}^{N}\left[H(\mathbf{x}_i) - \mathbf{y}_i^o\right]^T \mathbf{R}_i^{-1}\left[H(\mathbf{x}_i) - \mathbf{y}_i^o\right].$$

The cost function is minimized with respect to the *initial* state of the model with the time interval $\mathbf{x}(t_0)$, and the analysis at the time of the interval is given by the *model integration* from the solution $\mathbf{x}(t_n) = M_0[\mathbf{x}(t_0)]$.

The gradient of the 4D-Var functional, as will be shown in the sequel, is given by

$$\frac{\partial J}{\partial \mathbf{x}(t_0)} = \mathbf{B}_0^{-1}\left[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)\right] + \sum_{i=0}^{N} \mathbf{L}^T(t_i, t_0)\mathbf{H}_i^T \mathbf{R}_i^{-1}[H(\mathbf{x}_i) - \mathbf{y}_i^o],$$

where $\mathbf{H}_i$ and $\mathbf{L}_i$ are the linearized Jacobian matrices $\partial H / \partial \mathrm{x}_i$ and $\partial M / \partial \mathrm{x}_0$, respectively.
In order to obtain the last equation, we have to define the *tangent linear* and *adjoint models*[5]

[5] Assuming a time interval $[t_0, t_i]$, the tangent linear model advances a perturbation from t_0 to t_i, whereas the adjoint model advances a perturbation backward in time, from the t_i to t_0. Refer to the Definition (2.3.1).

($\mathbf{L}(t_0, t_i)$ and $\mathbf{L}^T(t_i, t_0)$, respectively), concepts that will be clarified later.

We saw that OI and 3D-Var methods assume a constant background error covariance matrix. This assumption implies that all the background errors are statistically stationary, which is not true in weather forecasting. In fact, there is a day-to-day variability in the forecast error, therefore it is important to take into account the "*errors of the day*" during the assimilation process. We give a brief introduction to a more sophisticated method, that falls within the sequential assimilation techniques, the *Kalman Filtering* (KF). The Kalman Filter is formally similar to OI, but in KF the forecast[6] or background error covariance, $\mathbf{P}^f(t_i)$, is advanced using the model itself. Kalman Filtering initially was designed for linear models but in the present work, we focus on the *Extended Kalman Filter* (EKF) which is used for nonlinear applications of Kalman Filtering.

The forecast state is advanced from the previous analysis time t_{i-1} to the current time t_i through the nonlinear model

$$\mathbf{x}^f(t_i) = M_{i-1}[\mathbf{x}^a(t_{i-1})].$$

The model is not perfect, i.e., we assume that the true state of the atmosphere is given by

$$\mathbf{x}^t(t_i) = M_{i-1}[\mathbf{x}^t(t_{i-1})] + \eta(t_{i-1}),$$

where $\eta(t_{i-1})$ is a zero-mean noise process with covariance matrix $\mathbf{Q}_{i-1} = E\{\eta_{i-1}\eta_{i-1}^T\}$.

The model is nonlinear, therefore we have to linearize it between two consecutive time steps t_{i-1} and t_i, in order to obtain the forecast error covariance. Introducing a perturbation and performing a first-order approximation of the model, the forecast error is in the form

$$\boldsymbol{\varepsilon}_i^f = \mathbf{x}^t(t_i) - \mathbf{x}^f(t_i) \approx \mathbf{L}_{i-1}\boldsymbol{\varepsilon}_{i-1}^a + \eta_{i-1},$$

where $\mathbf{L}_{i-1}$ is the tangent linear model matrix, i.e., the matrix that transforms the initial perturbation from time t_{i-1} to time t_i. Consequently, the forecast error covariance matrix is found to be

$$\mathbf{P}^f(t_i) = \mathbf{L}_{i-1}\ \mathbf{P}^a(t_{i-1})\ \mathbf{L}_{i-1}^T + \mathbf{Q}_{i-1}.$$

Once again, we assume that the observations contain errors with zero mean and error covariance matrix $\mathbf{R}_i = E\{\boldsymbol{\varepsilon}_i^o, \boldsymbol{\varepsilon}_i^{oT}\}$ and are given by

$$\mathbf{y}_i^o = H[\mathbf{x}^t(t_i)] + \boldsymbol{\varepsilon}_i^o,$$

where H is the (nonlinear) observation operator. After completing the forecast step at time t_i, the innovation vector is

$$\mathbf{d}_i = \mathbf{y}_i^o - H[\mathbf{x}^f(t_i)]$$

and the optimal weight matrix, or the so-called *Kalman gain*, that minimizes the analysis error covariance $\mathbf{P}_i^a$ is found to be

$$\mathbf{K}_i = \mathbf{P}^f(t_i)\ \mathbf{H}_i^T \left[\mathbf{R}_i + \mathbf{H}_i\ \mathbf{P}^f(t_i)\ \mathbf{H}_i^T\right]^{-1}.$$

[6] We use the superscript f for forecast instead of b, since the Kalman Filter algorithm consists of two steps: the forecast and the analysis step.

Following the above calculations, we arrive at the analysis state and its error covariance written as in OI, using the calculated $\mathbf{P}^f(t_i)$ and $\mathbf{K}_i$ matrices, instead of $\mathbf{B}$ and $\mathbf{W}$, respectively:

$$\begin{aligned}\mathbf{x}^a(t_i) &= \mathbf{x}^f(t_i) + \mathbf{K}_i\mathbf{d}_i,\\ \mathbf{P}^a(t_i) &= (\mathbf{I} - \mathbf{K}_i\mathbf{H}_i)\mathbf{P}^f(t_i).\end{aligned}$$

Although the Extended Kalman Filter provides the analysis estimate *and* its uncertainty, it requires the calculation of the tangent linear model matrix, which has size n (the degrees of freedom of the model), as well as the update of the error covariance, which is equivalent to performing $\mathcal{O}(n)$ model integrations. Thus, for high-dimensional problems EKF is costly.

There is a simplification of Kalman Filtering, the *Ensemble Kalman Filter* (EnKF), which does not require the derivation of the tangent linear operator or integrations backward in time. It is an approximation of the EKF, which avoids evolving the entire error covariance matrix at every time step. Instead, an ensemble of K data assimilation cycles is used to estimate the forecast uncertainty. Thus, we seek an *analysis ensemble mean* which reflects both an estimate of the true atmospheric state and its uncertainty.

We begin with an ensemble $\{\mathbf{x}_k^a,\ k = 1, \ldots, K,\quad \mathbf{x}_k^a \in X\}$ consisting of K members at time t_{i-1}. Evolving each ensemble member according to the nonlinear forecast model

$$\mathbf{x}_k^f(t_i) = M_{i-1}[\mathbf{x}_k^a(t_{i-1})], \qquad k = 1, \ldots, K,$$

we obtain the forecast ensemble $\mathbf{x}_k^f(t_i)$ at time $t_i,\ k = 1, \ldots, K$.
Then, the best available estimate to the system state, before the observations are taken into account, is the background ensemble mean

$$\overline{\mathbf{x}}^f = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k^f.$$

We define the background ensemble perturbations matrix $\mathbf{X}^f$, whose k-th column is defined as

$$\mathbf{X}^f = \frac{1}{\sqrt{K-1}}\left(\mathbf{x}_1^f - \overline{\mathbf{x}}^f, \ldots, \mathbf{x}_K^f - \overline{\mathbf{x}}^f\right). \tag{1.1}$$

Then, the uncertainty in the state estimate is described by the background error covariance matrix

$$\mathbf{P}^f = \mathbf{X}^f(\mathbf{X}^f)^T.$$

We seek an ensemble $\{\mathbf{x}_k^a,\ k = 1, \ldots, K\}$ having sample mean

$$\overline{\mathbf{x}}^a = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k^a$$

and error covariance matrix

$$\mathbf{P}^a = \mathbf{X}^a(\mathbf{X}^a)^T,$$

where $\mathbf{X}^a$ is the $n \times K$ matrix of the analysis ensemble perturbations, defined as

$$\mathbf{X}^a = \frac{1}{\sqrt{K-1}}\left(\mathbf{x}_1^a - \overline{\mathbf{x}}^a, \ldots, \mathbf{x}_K^a - \overline{\mathbf{x}}^a\right). \tag{1.2}$$

Then, the analysis estimate is simply the analysis ensemble mean, which based on the standard KF is

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{K}[\mathbf{y}^o - H(\overline{\mathbf{x}}^f)],$$

where the Kalman Gain matrix is given by

$$\mathbf{K} = \mathbf{X}^f(\mathbf{X}^f)^T\mathbf{H}^T[\mathbf{H}\mathbf{X}^f(\mathbf{X}^f)^T\mathbf{H}^T + \mathbf{R}]^{-1}.$$

1.2 The Lorenz-96 Model

We shall test the 3D-Var and EnKF techniques using the Lorenz-96 model. This model was introduced by Edward N. Lorenz in 1996, see [25]. The system consists of N ordinary differential equations

$$\frac{dX_i}{dt} = \left(X_{i+1} - X_{i-2}\right) X_{i-1} - X_i + F, \tag{1.3}$$

for $i = 1, \ldots, N$ and cyclic boundary conditions $X_{-1} = X_{N-1}$, $X_0 = X_N$, $X_{N+1} = X_1$. Here, F is a constant forcing term. It has been extensively used as a toy-model for Numerical Weather Prediction because it shares certain properties with many atmospheric models:

- the nonlinear, quadratic term simulates advection and conserves the total energy, defined as $(X_1^2 + \ldots + X_N^2)/2$,
- the linear term is supposed to represent mechanical or thermal dissipation and decreases the total energy,
- the constant term represents external forcing, which prevents the total energy from decaying to zero.

Lorenz concluded that similar error growth characteristics to operational NWP systems are obtained in the Lorenz-96 system if a time unit is associated with 5 days. If we multiply (1.3) by X_i and average over all values of n and over a long enough time to make average time derivatives negligibly small, it follows that

$$\overline{X^2} = F\overline{X}, \tag{1.4}$$

where the bar ($^-$) over the quantities denotes the average. From the last expression, we can form the variance of X, i.e.,

$$\sigma^2 = E[X^2] - \left(E\left[X\right]\right)^2 = \overline{X^2} - \overline{X}^2 = \overline{X}(F - \overline{X}). \tag{1.5}$$

Since the variance is always nonnegative, it follows that the mean $\overline{X}$ lies in the interval $[0, F]$ and the standard deviation σ lies in the interval $[0, F/2]$. In the steady state solution where $X_i = F$ for each i, we have $\overline{X} = F$ and $\sigma = 0$.

We assume $N = 40$ variables and the forcing term to be $F = 8$. For that value of F, it has been shown that the system exhibits chaotic behavior and for $N = 40$ there are 13 positive Lyapunov exponents, where the largest corresponds to a doubling time of 2.1 days (a value close to one that seems to prevail in some large atmospheric models). The variables fluctuate about the mean with a climatological standard deviation $\sigma_{clim} \approx 3.6$ (refer to [25, 26] for a more detailed discussion of the model).

We set as initial condition of the system the steady state solution $X_i = F$, for each $i = 1, \ldots, N$. We introduce a perturbation in the middle variable X_{20} by 0.008, i.e., $X_{20} = F + 0.008 = 8.008$ and perform numerical integration using a hand-coded fourth-order Runge-Kutta scheme [1], with a time-step $\Delta t = 0.05$ or 6 hours.

The remaining of this thesis is organized as follows: In chapter 2, we begin by presenting a simple least-squares linear estimation problem, seeking the optimal analysis estimate of a scalar quantity, given two independent observations. Thereafter, we try to find the same best estimate using the variational approach of the problem, i.e., we proceed to the minimization of a particular cost function. We introduce the Optimal Interpolation (OI), Three-Dimensional Variational assimilation (3D-Var) and Four-Dimensional Variational assimilation (4D-Var) methods and present some results of the 3D-Var implementation for the Lorenz-96 model.

In chapter 3, we present two variants of Kalman Filtering (KF): the Extended Kalman Filter (EKF) and the Ensemble Kalman Filter (EnKF). As seen in the sequel, the EKF is being used for strongly nonlinear models as it enables the linearization of the operators encountered in the model, while the EnKF, which falls within the category of Ensemble Forecasting, has an easier implementation since it does not require any linearization and has a big computational advantage compared to the EKF. The implementation of the Ensemble Kalman Filter is not unique, it can be regarded either as a Stochastic or as a Deterministic filter. In this thesis, we consider the deterministic implementation, which is known as the *Ensemble Square Root Filter* (EnSRF). After the description of the Ensemble Kalman filtering, we discuss some results of the EnSRF implementation for the Lorenz-96 model.

In chapter 4, we present the Weather, Research and Forecasting Model (WRF model), which is a state-of-the-art atmospheric modeling system designed for both meteorological research and numerical weather prediction. We focus on the WRF Data Assimilation system (WRFDA) provided by the WRF Model, which includes an incremental variational 3D-Var and 4D-Var algorithm, as well as a hybrid scheme between the variational and ensemble approaches, known as Hybrid ETKF-3DVAR.

Chapter 2

Data Assimilation Techniques

According to Talagrand [33] and Kalnay [18], the best estimate of the state of the atmosphere (i.e., the analysis state) is obtained from a statistical combination of prior information about the atmosphere, and observations. That prior information is what we call *background* or *first guess*. In order to obtain the optimal estimate, we need statistical information about the errors included in the observations.

Least-squares linear estimation
We begin with a classic example based on statistical estimation theory. Suppose we want to find the best estimate of the true value of a scalar quantity, for instance the true temperature T_t, given two independent observations T_1 and T_2 of the form

$$\begin{aligned} T_1 &= T_t + \varepsilon_1, \\ T_2 &= T_t + \varepsilon_2, \end{aligned} \tag{2.1}$$

where $\varepsilon_i,\ i = 1, 2$ are "observational" errors. We represent the expected value as $E\{\cdot\}$ and we assume that the instruments that measure T_1 and T_2 are unbiased, i.e., $E\{T_1\} - T_t = 0$ and $E\{T_2\} - T_t = 0$ that is,

$$E\{\varepsilon_1\} = E\{\varepsilon_2\} = 0. \tag{2.2}$$

Given the assumption made in (2.2), we have that the variances of the observational errors are

$$E\{\varepsilon_1^2\} = \sigma_1^2 \quad \text{and} \quad E\{\varepsilon_2^2\} = \sigma_2^2 \tag{2.3}$$

and we also assume that the errors of the two observations are uncorrelated, i.e., $E\{\varepsilon_1\varepsilon_2\} = 0$. Using a linear combination of T_1, T_2 which represent the available information about the true value of the temperature, we try to estimate T_t. This linear combination, is the *analysis* T_a:

$$T_a = a_1T_1 + a_2T_2. \tag{2.4}$$

Since the expected value is a linear operator, we have that

$$\begin{aligned} E\{T_a\} &= E\{a_1T_1 + a_2T_2\} \\ &= a_1E\{T_1\} + a_2E\{T_2\} \\ &= a_1E\{T_t + \varepsilon_1\} + a_2E\{T_t + \varepsilon_2\} \\ &= a_1E\{T_t\} + E\{\varepsilon_1\} + a_2E\{T_t\} + E\{\varepsilon_2\}, \end{aligned}$$

which leads to the conclusion that

$$E\{T_a\} = a_1 E\{T_t\} + a_2 E\{T_t\}. \tag{2.5}$$

The requirement that the analysis be unbiased, i.e., $E\{T_a\} = T_t$, implies that

$$a_1 + a_2 = 1. \tag{2.6}$$

We understand that T_a will be the best estimate of T_t if the coefficients a_1 and a_2 are chosen as to minimize, for example, the *mean square error* (MSE) of T_a:

$$\mathrm{MSE}(T_a) = E\left\{(T_a - T_t)^2\right\} = E\left\{(a_1\varepsilon_1 + a_2\varepsilon_2)^2\right\} = E\left\{\left[a_1(T_1 - T_t) + a_2(T_2 - T_t)\right]^2\right\}. \tag{2.7}$$

The last expression holds because

$$\begin{aligned} T_a - T_t &= a_1T_1 + a_2T_2 - T_t \\ &= a_1(T_t + \varepsilon_1) + a_2(T_t + \varepsilon_2) - T_t \\ &= (a_1 + a_2)T_t + a_1\varepsilon_1 + a_2\varepsilon_2 - T_t \\ &= T_t + a_1\varepsilon_1 + a_2\varepsilon_2 - T_t \\ &= a_1\varepsilon_1 + a_2\varepsilon_2. \end{aligned}$$

For an unbiased estimator, such as T_a, the MSE is the variance of the estimator. Combining this fact together with (2.6) and (2.7), we get an expression for the variance of T_a:

$$\begin{aligned} \sigma_a^2 &= E\{(a_1\varepsilon_1 + a_2\varepsilon_2)^2\} \\ &= E\{a_1^2\varepsilon_1^2 + (1 - a_1)^2\varepsilon_2^2 + 2a_1(1 - a_1)\varepsilon_1\varepsilon_2\} \\ &= E\{a_1^2\varepsilon_1^2 + \varepsilon_2^2 + a_1^2\varepsilon_2^2 - 2a_1\varepsilon_2^2 + 2a_1\varepsilon_1\varepsilon_2 - 2a_1^2\varepsilon_1\varepsilon_2\} \\ &= a_1^2E\{\varepsilon_1^2\} + E\{\varepsilon_2^2\} + a_1^2E\{\varepsilon_2^2\} - 2a_1E\{\varepsilon_2^2\} + 2a_1E\{\varepsilon_1\varepsilon_2\} - 2a_1^2E\{\varepsilon_1\varepsilon_2\} \\ &= a_1^2\sigma_1^2 + \sigma_2^2 + a_1^2\sigma_2^2 - 2a_1\sigma_2^2. \end{aligned} \tag{2.8}$$

We wish to minimize σ_a^2 with respect to a_1 which translates to $\partial\sigma_a^2/\partial a_1 = 0$, i.e.,

$$2a_1\sigma_1^2 + 2a_1\sigma_2^2 - 2\sigma_2^2 = 0. \tag{2.9}$$

Solving for a_1 and a_2 yields

$$a_1 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \quad \text{and} \quad a_2 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}. \tag{2.10}$$

An alternative form for a_1, a_2 can be found by dividing equation (2.9) by $\sigma_1^2\sigma_2^2$

$$\frac{a_1}{\sigma_2^2} + \frac{a_1}{\sigma_1^2} - \frac{1}{\sigma_1^2} = 0,$$

or equivalently,

$$a_1\left(\frac{1}{\sigma_2^2} + \frac{1}{\sigma_1^2}\right) - \frac{1}{\sigma_1^2} = 0.$$

This leads to the expressions

$$a_1 = \frac{1/\sigma_1^2}{1/\sigma_1^2 + 1/\sigma_2^2} \ \text{and} \ a_2 = \frac{1/\sigma_2^2}{1/\sigma_1^2 + 1/\sigma_2^2}\ . \tag{2.11}$$

Substituting the coefficients (2.10) into equation (2.8), we obtain a relationship between the analysis variance and the observational variances

$$\begin{aligned}
\sigma_a^2 &= a_1^2(\sigma_1^2+\sigma_2^2)+\sigma_2^2-2a_1\sigma_2^2 \\
&= \frac{\sigma_2^4}{(\sigma_1^2+\sigma_2^2)^2}(\sigma_1^2+\sigma_2^2)+\sigma_2^2-2\frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2}\sigma_2^2 \\
&= \frac{\sigma_2^4}{\sigma_1^2+\sigma_2^2}+\sigma_2^2-2\frac{\sigma_2^4}{\sigma_1^2+\sigma_2^2} \\
&= \sigma_2^2-\frac{\sigma_2^4}{\sigma_1^2+\sigma_2^2} \\
&= \frac{(\sigma_1^2+\sigma_2^2)\sigma_2^2-\sigma_2^4}{\sigma_1^2+\sigma_2^2} \\
&= \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}. \qquad (2.12)
\end{aligned}$$

Moreover, the inverse of the analysis error variance is

$$\frac{1}{\sigma_a^2}=\frac{\sigma_1^2+\sigma_2^2}{\sigma_1^2\sigma_2^2}=\frac{\sigma_1^2}{\sigma_1^2\sigma_2^2}+\frac{\sigma_2^2}{\sigma_1^2\sigma_2^2},$$

therefore,

$$\frac{1}{\sigma_a^2}=\frac{1}{\sigma_1^2}+\frac{1}{\sigma_2^2}. \qquad (2.13)$$

The last expression indicates that the *precision* of the analysis, i.e., the inverse of the corresponding error variance, is the sum of the precisions of the observations.

Variational approach

The same best estimate of T_t can be found using a different approach. We minimize the function $J(T)$, defined as *the sum of the square of the distance between the estimate* T *and the two observations*

$$J(T)=\frac{1}{2}\left[\frac{(T-T_1)^2}{\sigma_1^2}+\frac{(T-T_2)^2}{\sigma_2^2}\right], \qquad (2.14)$$

where the observational error variances σ_1^2, σ_2^2 account for the accuracy of the observations.

The minimum of the cost function (2.14) with respect to $T = T_a$, occurs when $\partial J/\partial T_a = 0$. Differentiating (2.14) with respect to T_a yields

$$\begin{aligned}
\frac{\partial J}{\partial T_a} &= \frac{\partial}{\partial T_a}\left[\frac{(T_a-T_1)^2}{2\sigma_1^2}+\frac{(T_a-T_2)^2}{2\sigma_2^2}\right] \\
&= \frac{\partial}{\partial T_a}\left[\frac{T_a^2+T_1^2-2T_aT_1}{2\sigma_1^2}+\frac{T_a^2+T_2^2-2T_aT_2}{2\sigma_2^2}\right] \\
&= \frac{2T_a-2T_1}{2\sigma_1^2}+\frac{2T_a-2T_2}{2\sigma_2^2}.
\end{aligned}$$

Therefore, we want

$$\frac{\partial J}{\partial T_a} = \frac{T_a - T_1}{\sigma_1^2} + \frac{T_a - T_2}{\sigma_2^2} = 0. \tag{2.15}$$

Initially, we calculate the quantities $T_a - T_1$ and $T_a - T_2$ using the definitions (2.1), (2.4) :

$$\begin{aligned}
T_a - T_1 &= a_1T_1 + a_2T_2 - T_1 \\
&= (a_1 - 1)T_1 + a_2T_2 \\
&= (a_1 - 1)(T_t + \varepsilon_1) + a_2(T_t + \varepsilon_2) \\
&= a_1T_t + a_1\varepsilon_1 - T_t - \varepsilon_1 + a_2T_t + a_2\varepsilon_2 \\
&= (a_1 + a_2)T_t - T_t + a_1\varepsilon_1 + a_2\varepsilon_2 - \varepsilon_1 \qquad \text{(we assumed that } a_1 + a_2 = 1) \\
&= T_t - T_t + a_1\varepsilon_1 + a_2\varepsilon_2 - \varepsilon_1 \\
&= a_1\varepsilon_1 + a_2\varepsilon_2 - \varepsilon_1, \qquad (2.16)
\end{aligned}$$

$$\begin{aligned}
T_a - T_2 &= a_1T_1 + a_2T_2 - T_2 \\
&= a_1T_1 + (a_2 - 1)T_2 \\
&= a_1(T_t + \varepsilon_1) + (a_2 - 1)(T_t + \varepsilon_2) \\
&= a_1T_t + a_1\varepsilon_1 + a_2T_t + a_2\varepsilon_2 - T_t - \varepsilon_2 \\
&= (a_1 + a_2)T_t - T_t + a_1\varepsilon_1 + a_2\varepsilon_2 - \varepsilon_2 \qquad \text{(we assumed that } a_1 + a_2 = 1) \\
&= T_t - T_t + a_1\varepsilon_1 + a_2\varepsilon_2 - \varepsilon_2 \\
&= a_1\varepsilon_1 + a_2\varepsilon_2 - \varepsilon_2. \qquad (2.17) \\
& \qquad (2.18)
\end{aligned}$$

Substituting (2.16), (2.17) into (2.15), the result is

$$\begin{aligned}
\frac{a_1\varepsilon_1 + a_2\varepsilon_2 - \varepsilon_1}{\sigma_1^2} + \frac{a_1\varepsilon_1 + a_2\varepsilon_2 - \varepsilon_2}{\sigma_2^2} &= 0 \\
\sigma_2^2(a_1\varepsilon_1 + a_2\varepsilon_2 - \varepsilon_1) + \sigma_1^2(a_1\varepsilon_1 + a_2\varepsilon_2 - \varepsilon_2) &= 0 \\
(a_1\varepsilon_1 + a_2\varepsilon_2)(\sigma_1^2 + \sigma_2^2) - \sigma_2^2\varepsilon_1 - \sigma_1^2\varepsilon_2 &= 0,
\end{aligned}$$

and thus we arrive at the expression

$$a_1\varepsilon_1 + a_2\varepsilon_2 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\varepsilon_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\varepsilon_2. \tag{2.19}$$

$$\tag{2.20}$$

In order to satisfy equation (2.19), the weights should be

$$a_1 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \quad \text{and} \quad a_2 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}.$$

We have arrived at the main results of this section, namely that the minimum of the cost function defined in (2.14) is obtained for $T = T_a$ with the same weights as in (2.10).

Maximum Likelihood Approach

The cost function (2.14) may also be formulated using the notion of the maximum likelihood:

> *Given two observations T_1 and T_2, which have normally distributed errors with standard deviations σ_1 and σ_2, respectively, find the most likely value of the true temperature T.*

We assume that the errors of the observations have Gaussian statistics. Therefore, the probability density functions of the observations T_1 and T_2 given a true value T and the corresponding standard deviations σ_1 and σ_2, are given by the Gaussian distributions:

$$p_{\sigma_1}(T_1|T) = \frac{1}{\sigma_1\sqrt{2\pi}} e^{-\frac{(T_1-T)^2}{2\sigma_1^2}} \qquad \text{and} \qquad p_{\sigma_2}(T_2|T) = \frac{1}{\sigma_2\sqrt{2\pi}} e^{-\frac{(T_2-T)^2}{2\sigma_2^2}}. \tag{2.21}$$

In order to use the maximum likelihood, we first need to specify the joint probability density function of the two observations. Since they are assumed to be independent, their joint probability is simply the product of their distributions $p_{\sigma_1}, p_{\sigma_2}$.

Now, we look at this joint probability from a different perspective, by considering the two observations to be fixed “parameters” of this function, whereas T will be the function’s variable. Therefore, we define the likelihood function:

$$\mathcal{L}_{\sigma_1,\sigma_2}(T||T_1,T_2) = p_{\sigma_1}(T_1|T)p_{\sigma_2}(T_2|T) = \frac{1}{2\pi\sigma_1\sigma_2}\exp\left\{-\tfrac{(T_1-T)^2}{2\sigma_1^2} - \tfrac{(T_2-T)^2}{2\sigma_2^2}\right\}. \tag{2.22}$$

The most likely value of T, given the two independent observations T_1 and T_2, is the one that maximizes the likelihood function

$$\max_T \; \mathcal{L}_{\sigma_1,\sigma_2}(T||T_1,T_2) = \frac{1}{2\pi\sigma_1\sigma_2}\exp\left\{-\tfrac{(T_1-T)^2}{2\sigma_1^2} - \tfrac{(T_2-T)^2}{2\sigma_2^2}\right\}. \tag{2.23}$$

The logarithm is a monotone function, therefore we can take the logarithm of the likelihood and obtain the same maximum likelihood temperature:

$$\begin{aligned} \max_T \; \ln\mathcal{L}_{\sigma_1,\sigma_2}(T||T_1,T_2) &= \max_T \; \left[\text{const.} - \tfrac{(T_1-T)^2}{2\sigma_1^2} - \tfrac{(T_2-T)^2}{2\sigma_2^2}\right] \\ &= \max_T \; \left[\text{const.} - J(T)\right]. \end{aligned} \tag{2.24}$$

Note that, since the standard deviations are constant, the maximum likelihood is attained when the cost function (2.14) is minimized.

Another alternative for the derivation of (2.14) is the Bayesian approach. We assume that we made an observation T_1 and thus, we have a *prior* probability distribution of the truth

$$p_{T_1,\sigma_1}(T) = \frac{1}{\sigma_1\sqrt{2\pi}} e^{-\frac{(T_1-T)^2}{2\sigma_1^2}}, \tag{2.25}$$

which precedes the second observation. From Bayes formula, the *a posteriori* probability of the truth given the observation T_2 is

$$p_{\sigma_2}(T|T_2) = \frac{p_{\sigma_2}(T_2|T)\; p_{T_1,\sigma_1}(T)}{p_{\sigma_2}(T_2)}. \tag{2.26}$$

Since the denominator

$$p_{\sigma_2}(T_2) = \int\limits_{T'} \frac{1}{\sigma_2\sqrt{2\pi}} e^{-\frac{(T_2-T')^2}{2\sigma_1^2}} dT',$$

is independent of T, the estimate of the truth which maximizes the *a posteriori* probability (2.26), is obtained by maximizing the logarithm of the numerator

$$\max_T \ \ln[p_{\sigma_2}(T_2|T)\ p_{T_1,\sigma_1}(T)] = \max_T \ \ln\left[\text{const.} - \tfrac{(T_2-T)^2}{2\sigma_2^2} - \tfrac{(T_1-T)^2}{2\sigma_1^2}\right]. \tag{2.27}$$

This is exactly the same as (2.24) and thus, the estimate is again the minimum of the cost function (2.14).

Sequential data assimilation
So far we have seen the simple example with the two pieces of information. Now, we assume that the information $T_1 = T_b$ is the forecast (or background), whereas $T_2 = T_o$ now represents an observation. The analysis is $T_a = a_bT_b + a_oT_o$. We know that

$$a_b = \frac{\sigma_o^2}{\sigma_b^2+\sigma_o^2}, \quad a_o = \frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2} \quad \text{and also,} \quad a_b + a_o = 1.$$

Thus, we rewrite the analysis as

$$T_a = (1-a_o)T_b + a_oT_o = T_b + a_o(T_o - T_b) = T_b + W(T_o - T_b), \tag{2.28}$$

where W is the *optimal weight* given by

$$W = a_o = \frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2}. \tag{2.29}$$

Looking back at the equation (2.28), the difference between the observation and the background is refered to as the *innovation*. This equation indicates that the analysis is obtained by adding to the background the innovation weighted by the optimal weight.
Regarding the optimal weight W, it is defined as the background error variance σ_b^2 divided by the sum of the background and the observation error variances, which means that the larger the background error variance, the larger the correction to the first guess.

The analysis error variance is

$$\sigma_a^2 = \frac{\sigma_b^2\sigma_o^2}{\sigma_b^2+\sigma_o^2},$$

as defined in (2.12). It can also be written (including the optimal weight) as

$$\sigma_a^2 = \frac{\sigma_o^2}{\sigma_b^2+\sigma_o^2}\sigma_b^2 = a_b\sigma_b^2 = (1-a_o)\sigma_b^2 = (1-W)\sigma_b^2. \tag{2.30}$$

These properties of the analysis are important because they also apply in multidimensional problems. In subsequent sections we study assimilation methods, such as the Optimal Interpolation and the Kalman Filter, in which T_a and T_b are three-dimensional fields and T_o represents a set of observations. We will have to replace the "*error variance*" by "*error covariance matrix*" and the "*optimal weight*" by an "*optimal gain matrix*".

2.1 Optimal Interpolation

In this section, we consider the complete Numerical Weather Prediction (NWP) operational problem:

> *Find an optimal analysis of a field of model variables $\mathbf{x}^a$, given*
> - *a background field $\mathbf{x}^b$ available at grid points in two or three dimensions,*
> - *a set of p observations $\mathbf{y}^o$ available at arbitrarily distributed locations $\boldsymbol{r}_i$, $i = 1, \ldots, p$.*

As indicated earlier, the analysis is the background plus the innovation weighted by the optimal weight which we obtain from statistical interpolation

$$\mathbf{x}^t - \mathbf{x}^b = \mathbf{W}[\mathbf{y}^o - H(\mathbf{x}^b)] - \boldsymbol{\varepsilon}_a = \mathbf{W}\mathbf{d} - \boldsymbol{\varepsilon}_a, \quad \boldsymbol{\varepsilon}_a = \mathbf{x}^a - \mathbf{x}^t, \tag{2.31}$$

$$\text{or} \quad \mathbf{x}^a = \mathbf{x}^b + \mathbf{W}[\mathbf{y}^o - H(\mathbf{x}^b)] = \mathbf{x}^b + \mathbf{W}\mathbf{d}. \tag{2.32}$$

The truth, $\mathbf{x}^t$, the background, $\mathbf{x}^b$, and the analysis, $\mathbf{x}^a$, are vectors of length n. The weights are given by the matrix W of dimension $n \times p$ and H is the observational operator which transforms model variables into observed variables and is, generally, nonlinear.
Moreover, the observation vector $\mathbf{y}^o$ is of length p (where p is the number of the available observations) and $\mathbf{d}$ is the innovation or "*observational increments*" vector, defined as

$$\mathbf{d} = \mathbf{y}^o - H(\mathbf{x}^b). \tag{2.33}$$

The Best Linear Unbiased Estimator (BLUE)
An important concept of the Optimal Interpolation is that it produces the best linear unbiased estimate of a field, given a set of arbitrarily distributed observations. Assume that we have two time series

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}, \qquad \mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_p(t) \end{bmatrix}, \tag{2.34}$$

with zero mean values $E(\mathbf{x}) = 0$, $E(\mathbf{y}) = 0$, respectively. The Best Linear Unbiased Estimation (BLUE) of $\mathbf{x}$ in terms of $\mathbf{y}$ is

$$\mathbf{x}^a(t) = \mathbf{W}\mathbf{y}(t), \tag{2.35}$$

which is an approximation of the true

$$\mathbf{x}(t) = \mathbf{W}\mathbf{y}(t) - \boldsymbol{\varepsilon}(t), \tag{2.36}$$

where $\boldsymbol{\varepsilon}(t) = \mathbf{x}^a - \mathbf{x}^t$ is the analysis error and the weight matrix $\mathbf{W}$ actually minimizes the *mean square error* $E\{\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}\}$.

In order to derive an explicit form for $\mathbf{W}$ we write (2.36) in component form

$$x_i(t) = \sum_{k=1}^{p} W_{ik} y_k(t) - \varepsilon_i(t), \quad i = 1, \ldots, n. \tag{2.37}$$

Solving for the analysis error, squaring and summing, gives

$$\sum_{i=1}^{n} \varepsilon_i^2(t) = \sum_{i=1}^{n} \left[\sum_{k=1}^{p} W_{ik} y_k(t) - x_i(t) \right]^2 . \tag{2.38}$$

Now, we take the derivative with respect to the components of $\mathbf{W}$

$$\frac{\partial \left(\sum\limits_{i=1}^{n} \varepsilon_i^2(t) \right)}{\partial W_{ij}} = 2 \left[\sum_{k=1}^{p} W_{ik} y_k(t) - x_i(t) \right] y_j(t) = 2 \left[\sum_{k=1}^{p} W_{ik} y_k(t) y_j(t) - x_i(t) y_j(t) \right] .$$

Setting it equal to zero, we have

$$\left[\mathbf{W}\mathbf{y}\mathbf{y}^T \right]_{ij} - \left[\mathbf{x}\mathbf{y}^T \right]_{ij} = 0.$$

Taking the expected value of the last equation we get the optimal weight matrix which gives the best linear unbiased estimation in (2.35)

$$\mathbf{W} E\{\mathbf{y}\mathbf{y}^T\} - E\{\mathbf{x}\mathbf{y}^T\} = 0,$$

therefore,

$$\mathbf{W} = E\left\{ \mathbf{x}\mathbf{y}^T \right\} \left[E\left\{ \mathbf{y}\mathbf{y}^T \right\} \right]^{-1} . \tag{2.39}$$

We make the following assumptions: The background error and the analysis error are vectors of length n:

$$\begin{aligned} \boldsymbol{\varepsilon}_b(x,y) &= \mathbf{x}^b(x,y) - \mathbf{x}^t(x,y), \\ \boldsymbol{\varepsilon}_a(x,y) &= \mathbf{x}^a(x,y) - \mathbf{x}^t(x,y). \end{aligned} \tag{2.40}$$

The available observations contain error which is defined as

$$\boldsymbol{\varepsilon}_o(\boldsymbol{r}_i) = \mathbf{y}^o(\boldsymbol{r}_i) - \mathbf{y}^t(\boldsymbol{r}_i) = \mathbf{y}^o(\boldsymbol{r}_i) - H[\mathbf{x}^t(\boldsymbol{r}_i)]. \tag{2.41}$$

We don't know the errors of the available background and observations, because we don't know the truth state $\mathbf{x}^t$. However, we can make some assumptions about the statistical properties of these errors. We assume that the background and the observations are unbiased:

$$\begin{aligned} E\{\boldsymbol{\varepsilon}_b(x,y)\} &= E\{\mathbf{x}^b(x,y)\} - E\{\mathbf{x}^t(x,y)\} = 0, \\ E\{\boldsymbol{\varepsilon}_o(\boldsymbol{r}_i)\} &= E\{\mathbf{y}^o(\boldsymbol{r}_i)\} - E\{\mathbf{y}^t(\boldsymbol{r}_i)\} = 0. \end{aligned} \tag{2.42}$$

We define the error covariance matrices for the background, the observations and the analysis, respectively,

$$\mathbf{B} = E\{\boldsymbol{\varepsilon}_b \boldsymbol{\varepsilon}_b^T\}, \quad \mathbf{R} = E\{\boldsymbol{\varepsilon}_o \boldsymbol{\varepsilon}_o^T\}, \quad \mathbf{P}^a = E\{\boldsymbol{\varepsilon}_a \boldsymbol{\varepsilon}_a^T\}, \tag{2.43}$$

and we assume that the background and the observation errors are uncorrelated, i.e.,

$$E\{\boldsymbol{\varepsilon}_o \boldsymbol{\varepsilon}_b^T\} = 0.$$

The error covariance matrices $\mathbf{B} \in \mathbb{R}^{n \times n}$ and $\mathbf{R} \in \mathbb{R}^{p \times p}$ are assumed to be known.

The nonlinear observation operator H can be linearized using the Taylor series expansion, as

$$H(\mathbf{x} + \delta\mathbf{x}) = H(\mathbf{x}) + \mathbf{H}\delta\mathbf{x}, \tag{2.44}$$

where $\mathbf{H}$ is a $p \times n$ matrix whose elements are the first-order partial derivatives $h_{i,j} = \partial H_i/\partial x_j$. The innovation vector (2.33) is now

$$\begin{aligned} \mathbf{d} &= \mathbf{y}^o - H(\mathbf{x}^b) = \mathbf{y}^o - H(\mathbf{x}^t + (\mathbf{x}^b - \mathbf{x}^t)) \\ &= \mathbf{y}^o - H(\mathbf{x}^t) - \mathbf{H}(\mathbf{x}^b - \mathbf{x}^t) = \boldsymbol{\varepsilon}_o - \mathbf{H}\boldsymbol{\varepsilon}_b. \end{aligned} \tag{2.45}$$

We use the BLUE formula to derive the optimal weight matrix $\mathbf{W}$ defined in (2.31). From the equations (2.45) and (2.39), we have

$$\begin{aligned} \mathbf{W} &= E\left\{(\mathbf{x}^t - \mathbf{x}^b)[\mathbf{y}^o - H(\mathbf{x}^b)]^T\right\}\left(E\left\{[\mathbf{y}^o - H(\mathbf{x}^b)][\mathbf{y}^o - H(\mathbf{x}^b)]^T\right\}\right)^{-1} \\ &= E\left\{(-\boldsymbol{\varepsilon}_b)(\boldsymbol{\varepsilon}_o - \mathbf{H}\boldsymbol{\varepsilon}_b)^T\right\}\left(E\left\{(\boldsymbol{\varepsilon}_o - \mathbf{H}\boldsymbol{\varepsilon}_b)(\boldsymbol{\varepsilon}_o - \mathbf{H}\boldsymbol{\varepsilon}_b)^T\right\}\right)^{-1} \\ &= \left[-E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_o^T\} + E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T\}\ \mathbf{H}^T\right]\left[E\{\boldsymbol{\varepsilon}_o\boldsymbol{\varepsilon}_o^T\} - E\{\boldsymbol{\varepsilon}_o\boldsymbol{\varepsilon}_b^T\}\ \mathbf{H}^T - \mathbf{H}E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_o^T\} + \mathbf{H}\ E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T\}\ \mathbf{H}^T\right]^{-1} \\ &= E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T\}\ \mathbf{H}^T\left[E\{\boldsymbol{\varepsilon}_o\boldsymbol{\varepsilon}_o^T\} + \mathbf{H}\ E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T\}\ \mathbf{H}^T\right]^{-1}, \end{aligned}$$

therefore, we conclude that

$$\mathbf{W} = \mathbf{B}\mathbf{H}^T[\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T]^{-1}. \tag{2.46}$$

Note that the last equation holds because the background and observation errors are uncorrelated thus, the terms which include $E\{\boldsymbol{\varepsilon}_o\boldsymbol{\varepsilon}_b^T\}$ vanish.

Looking back at (2.31), we have that

$$\boldsymbol{\varepsilon}_a = \mathbf{W}\mathbf{d} + \mathbf{x}^b - \mathbf{x}^t = \mathbf{W}\mathbf{d} + \boldsymbol{\varepsilon}_b, \qquad \text{where } \ \mathbf{d} = \boldsymbol{\varepsilon}_o - \mathbf{H}\boldsymbol{\varepsilon}_b.$$

We can also derive the error covariance matrix, $\mathbf{P}^a$, of the analysis

$$\begin{aligned} \mathbf{P}^a &= E\{\boldsymbol{\varepsilon}_a\boldsymbol{\varepsilon}_a^T\} = E\left\{(\mathbf{W}\mathbf{d} + \boldsymbol{\varepsilon}_b)(\mathbf{W}\mathbf{d} + \boldsymbol{\varepsilon}_b)^T\right\} \\ &= \mathbf{W}E\{\mathbf{d}\mathbf{d}^T\}\mathbf{W}^T + \mathbf{W}E\{\mathbf{d}\boldsymbol{\varepsilon}_b^T\} + E\{\boldsymbol{\varepsilon}_b\mathbf{d}^T\}\mathbf{W}^T + E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T\}, \end{aligned} \tag{2.47}$$

where

$$\begin{aligned} E\{\mathbf{d}\mathbf{d}^T\} &= E\{(\boldsymbol{\varepsilon}_o - \mathbf{H}\boldsymbol{\varepsilon}_b)(\boldsymbol{\varepsilon}_o - \mathbf{H}\boldsymbol{\varepsilon}_b)^T\} \\ &= E\{\boldsymbol{\varepsilon}_o\boldsymbol{\varepsilon}_o^T\} - E\{\boldsymbol{\varepsilon}_o\boldsymbol{\varepsilon}_b^T\}\mathbf{H}^T - \mathbf{H}E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_o^T\} + \mathbf{H}E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T\}\mathbf{H}^T = \mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T, \\ E\{\mathbf{d}\ \boldsymbol{\varepsilon}_b^T\} &= E\{(\boldsymbol{\varepsilon}_o - \mathbf{H}\boldsymbol{\varepsilon}_b)\boldsymbol{\varepsilon}_b^T\} = E\{\boldsymbol{\varepsilon}_o\boldsymbol{\varepsilon}_b^T\} - \mathbf{H}E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T\} = -\mathbf{H}\mathbf{B}, \\ E\{\boldsymbol{\varepsilon}_b\mathbf{d}^T\} &= E\{\boldsymbol{\varepsilon}_b(\boldsymbol{\varepsilon}_o^T - \boldsymbol{\varepsilon}_b^T\mathbf{H}^T)\} = E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_o^T\} - E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T\}\mathbf{H}^T = -\mathbf{B}\mathbf{H}^T, \\ E\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T\} &= \mathbf{B}. \end{aligned}$$

Substituting the above terms into (2.47) we have

$$\begin{aligned} \mathbf{P}^a &= \mathbf{W}(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T)\mathbf{W}^T - \mathbf{W}\mathbf{H}\mathbf{B} - \mathbf{B}\mathbf{H}^T\mathbf{W}^T + \mathbf{B} \\ &= \mathbf{B} - \mathbf{W}\mathbf{H}\mathbf{B} - \mathbf{B}\mathbf{H}^T\mathbf{W}^T + \mathbf{W}\mathbf{R}\mathbf{W}^T + \mathbf{W}\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{W}^T. \end{aligned} \tag{2.48}$$

Gathering some common terms in (2.48), yields

$$\begin{aligned}\mathbf{P}^a &= (\mathbf{I} - \mathbf{WH})\mathbf{B} - \mathbf{BH}^T\mathbf{W}^T + \mathbf{W}(\mathbf{R} + \mathbf{HBH}^T)\mathbf{W}^T \\ &= (\mathbf{I} - \mathbf{WH})\mathbf{B} + \left[\mathbf{W}(\mathbf{R} + \mathbf{HBH}^T) - \mathbf{BH}^T\right]\mathbf{W}^T \quad (\text{from (2.46): } \mathbf{BH}^T = \mathbf{W}[\mathbf{R} + \mathbf{HBH}^T]) \\ &= (\mathbf{I} - \mathbf{WH})\mathbf{B} + \left[\mathbf{W}\cancel{(\mathbf{R} + \mathbf{HBH}^T)} - \mathbf{W}\cancel{(\mathbf{R} + \mathbf{HBH}^T)}\right]\mathbf{W}^T.\end{aligned}$$

Thus, we obtain a compact form for the analysis error covariance matrix

$$\mathbf{P}^a = (\mathbf{I} - \mathbf{WH})\mathbf{B}. \tag{2.49}$$

Summarizing, we give the *Optimal Interpolation equations*

$$\begin{aligned}&\mathbf{x}^a = \mathbf{x}^b + \mathbf{W}\left[\mathbf{y}^o - H(\mathbf{x}^b)\right] = \mathbf{x}^b + \mathbf{Wd}, \\ &\mathbf{W} = \mathbf{BH}^T(\mathbf{R} + \mathbf{HBH}^T)^{-1}, \\ &\mathbf{P}^a = (\mathbf{I} - \mathbf{WH})\mathbf{B}.\end{aligned} \tag{2.50}$$

2.2 Three-Dimensional Variational Assimilation

In the beginning of the chapter, we saw that there is an equivalence between the optimal analysis of a scalar obtained by minimizing the analysis error variance (finding the optimal weights in a least-squares sense) and the variational approach of the same problem (finding the analysis that minimizes a specific cost function). In this chapter, we will see that the same equivalence holds when the analysis involves a full three-dimensional field. In the Optimal Interpolation, we found the optimal weight $\mathbf{W}$ that is minimizing the analysis error covariance matrix. Now we deal with the following *variational assimilation problem* introduced by Lorenc in 1986, [21]:

> *Find the optimal analysis $\mathbf{x}^a$ field that minimizes a (scalar) cost function of the state $\mathbf{x}$ defined as the distance between $\mathbf{x}$ and the background $\mathbf{x}^b$, weighted by the inverse of the background error covariance, plus the distance to the observations $\mathbf{y}^o$ weighted by the inverse of the observations error covariance:*
>
> $$2J(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}^b) + [\mathbf{y}^o - H(\mathbf{x})]^T\mathbf{R}^{-1}[\mathbf{y}^o - H(\mathbf{x})]. \tag{2.51}$$

We can define with the same way the likelihood of the true given a background state and the observations, assuming that they have Gaussian statistics

$$\mathcal{L}_{\mathbf{B}}(\mathbf{x}||\mathbf{x}^b) = p_{\mathbf{B}}(\mathbf{x}^b|\mathbf{x}) = \frac{1}{(2\pi)^{n/2}\,|\mathbf{B}|^{1/2}}\ \exp\left\{-\frac{1}{2}\left[(\mathbf{x}^b - \mathbf{x})^T\mathbf{B}^{-1}(\mathbf{x}^b - \mathbf{x})\right]\right\}, \tag{2.52}$$

$$\mathcal{L}_{\mathbf{R}}(\mathbf{x}||\mathbf{y}^o) = p_{\mathbf{R}}(\mathbf{y}^o|\mathbf{x}) = \frac{1}{(2\pi)^{p/2}\,|\mathbf{R}|^{1/2}}\ \exp\left\{-\frac{1}{2}\left[\left[\mathbf{y}^o - \mathbf{H}(\mathbf{x})\right]^T\mathbf{R}^{-1}\left[\mathbf{y}^o - H(\mathbf{x})\right]\right]\right\}. \tag{2.53}$$

The background and the observations are independent, therefore their joint probability is the product of their probabilities. The most likely state, $\mathbf{x}$, of the atmosphere maximizes the joint probability (likelihood function) or the logarithm of the joint probability (log-likelihood function). As mentioned before, this process is the same as minimizing the cost function (2.51).

From a Bayesian point of view, the 3D-Var cost function can be derived (given the background field) based on the assumption that the true field is a realization of a random process with prior probability distribution function:

$$p_{\mathbf{B}}(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}\,|\mathbf{B}|^{1/2}}\;\exp\left\{-\frac{1}{2}\left[(\mathbf{x}^b-\mathbf{x})^T\mathbf{B}^{-1}(\mathbf{x}^b-\mathbf{x})\right]\right\}. \tag{2.54}$$

Bayes theorem gives the *a posteriori* probability distribution of the true field, given new observations, as

$$p(\mathbf{x}|\mathbf{y}^o) = \frac{p_{\mathbf{R}}(\mathbf{y}^o|\mathbf{x})\;p_{\mathbf{B}}(\mathbf{x})}{p(\mathbf{y}^o)}. \tag{2.55}$$

The Bayesian estimate of the true field is the one that maximizes the *a posteriori* probability. Since the denominator does not depend on the current state $\mathbf{x}$, the maximum of the a posteriori probability is attained when the numerator is maximum or, as we saw earlier, when the cost function (2.51) is minimized.

Returning to the 3D-Var cost function (2.51), the minimum is attained for $\mathbf{x} = \mathbf{x}^a$ when

$$\nabla_{\mathbf{x}} J(\mathbf{x}^a) = 0. \tag{2.56}$$

In order to find the minimum, assuming that the analysis is close to the truth and the observations, we linearize the observation operator H around the background value $\mathbf{x}^b$, using the Taylor series expansion

$$\mathbf{y}^o - H(\mathbf{x}) = \mathbf{y}^o - H[\mathbf{x}^b + (\mathbf{x}-\mathbf{x}^b)] = [\mathbf{y}^o - H(\mathbf{x}^b)] - \mathbf{H}(\mathbf{x}-\mathbf{x}^b), \tag{2.57}$$

where $\mathbf{H}$ is the matrix of the first-order partial derivatives $h_{ij} = \partial H_i/\partial x_j$.

Substituting into the cost function we have

$$\begin{aligned}
2J(\mathbf{x}) &= (\mathbf{x}-\mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b) \\
&\quad + \left[[\mathbf{y}^o - H(\mathbf{x}^b)] - \mathbf{H}(\mathbf{x}-\mathbf{x}^b)\right]^T \mathbf{R}^{-1}\left[[\mathbf{y}^o - H(\mathbf{x}^b)] - \mathbf{H}(\mathbf{x}-\mathbf{x}^b)\right] \\
&= (\mathbf{x}-\mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b) \\
&\quad + [\mathbf{y}^o - H(\mathbf{x}^b)]^T\mathbf{R}^{-1}[\mathbf{y}^o - H(\mathbf{x}^b)] - [\mathbf{y}^o - H(\mathbf{x}^b)]^T\mathbf{R}^{-1}\mathbf{H}(\mathbf{x}-\mathbf{x}^b) \\
&\quad - [\mathbf{H}(\mathbf{x}-\mathbf{x}^b)]^T\mathbf{R}^{-1}[\mathbf{y}^o - H(\mathbf{x}^b)] + [\mathbf{H}(\mathbf{x}-\mathbf{x}^b)]^T\mathbf{R}^{-1}\mathbf{H}(\mathbf{x}-\mathbf{x}^b) \\
&= (\mathbf{x}-\mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b) \\
&\quad + [\mathbf{y}^o - H(\mathbf{x}^b)]^T\mathbf{R}^{-1}[\mathbf{y}^o - H(\mathbf{x}^b)] - [\mathbf{y}^o - H(\mathbf{x}^b)]^T\mathbf{R}^{-1}\mathbf{H}(\mathbf{x}-\mathbf{x}^b) \\
&\quad - (\mathbf{x}-\mathbf{x}^b)^T\mathbf{H}^T\mathbf{R}^{-1}[\mathbf{y}^o - H(\mathbf{x}^b)] + (\mathbf{x}-\mathbf{x}^b)^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}(\mathbf{x}-\mathbf{x}^b).
\end{aligned}$$

Remark 2.2.1. *Given a quadratic function* $F(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{d}^T\mathbf{x} + c$, *where* $\mathbf{A}$ *is a symmetric matrix,* $\mathbf{d}$ *is a vector and* c *is a scalar, it is easy to check that the gradient of* F *is given by* $\nabla F(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{d}$.

Since the cost function is a quadratic function of the analysis increments $(\mathbf{x}-\mathbf{x}^b)$, we can use the remark above. The gradient of the cost function with respect to $\mathbf{x}$ (or $(\mathbf{x}-\mathbf{x}^b)$) is

$$\nabla J(\mathbf{x}) = \mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b) + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}(\mathbf{x}-\mathbf{x}^b) - \mathbf{H}^T\mathbf{R}^{-1}[\mathbf{y}^o - H(\mathbf{x}^b)]. \tag{2.58}$$

The equation $\nabla J(\mathbf{x}^a) = 0$ is equivalent to

$$\left[\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right](\mathbf{x}^a - \mathbf{x}^b) - \mathbf{H}^T\mathbf{R}^{-1}[\mathbf{y}^o - H(\mathbf{x}^b)] = 0.$$

Solving for $\mathbf{x}^a$ we obtain the 3D-Var analysis

$$\mathbf{x}^a = \mathbf{x}^b + \Big[\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\Big]^{-1}\mathbf{H}^T\mathbf{R}^{-1}[\mathbf{y}^o - H(\mathbf{x}^b)].$$

Therefore,

$$\begin{aligned} \mathbf{x}^a &= \mathbf{x}^b + \mathbf{W}[\mathbf{y}^o - H(\mathbf{x}^b)], \\ \text{where} \quad \mathbf{W} &= [\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{R}^{-1}. \end{aligned} \tag{2.59}$$

Formally, this is the solution of the 3D-Var problem. In practice, however, the minimum of the cost function $J(\mathbf{x})$ is obtained using iterative algorithms such as the conjugate gradient or the quasi-Newton methods.

Equivalence between OI and 3D-Var

We should note that the solutions of OI and 3D-Var are equivalent. Recall that the subject treated in OI was to find the optimal weights that minimize the analysis error variance, while in 3D-Var we wish to find the minimum of the cost function $J(\mathbf{x})$. We will show that the weight matrices of the two methods are in fact the same, i.e., that

$$\mathbf{W} = \Big[\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\Big]^{-1}\mathbf{H}^T\mathbf{R}^{-1} = \mathbf{B}\mathbf{H}^T\Big[\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T\Big]^{-1}. \tag{2.60}$$

We use the *Sherman-Morrison-Woodbury* formula

$$(\mathbf{A} + \mathbf{U}\mathbf{V}^T)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I} + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{A}^{-1}, \tag{2.61}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n\times k}$, to express in a different way the inverse matrices appearing in (2.60) and prove the equality. For the weight matrix of the 3D-Var we have

$$\begin{aligned} W_{\text{3D-Var}} &= \Big[\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\Big]^{-1}\mathbf{H}^T\mathbf{R}^{-1} \\ &= \Big[\mathbf{B} - \mathbf{B}\mathbf{H}^T\mathbf{R}^{-1}\Big(\mathbf{I} + \mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{R}^{-1}\Big)^{-1}\mathbf{H}\mathbf{B}\Big]\mathbf{H}^T\mathbf{R}^{-1} \\ &= \Big[\mathbf{B} - \mathbf{B}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{B} - \mathbf{B}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{R}(\mathbf{H}^T)^{-1}\mathbf{B}^{-1}\mathbf{H}^{-1}\mathbf{H}\mathbf{B}\Big]\mathbf{H}^T\mathbf{R}^{-1} \\ &= \Big[\mathbf{B} - \mathbf{B}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{B} - \mathbf{B}\Big]\mathbf{H}^T\mathbf{R}^{-1} \\ &= -\mathbf{B}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{R}^{-1} \end{aligned}$$

and for the weight matrix of OI

$$\begin{aligned} W_{\text{OI}} &= \mathbf{B}\mathbf{H}^T\Big[\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T\Big]^{-1} \\ &= \mathbf{B}\mathbf{H}^T\Big[\mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{H}\mathbf{B}\Big(\mathbf{I} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{B}\Big)^{-1}\mathbf{H}^T\mathbf{R}^{-1}\Big] \\ &= \mathbf{B}\mathbf{H}^T\Big[\mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{H}\mathbf{B}\Big(\mathbf{I} + \mathbf{B}^{-1}\mathbf{H}^{-1}\mathbf{R}(\mathbf{H}^T)^{-1}\Big)\mathbf{H}^T\mathbf{R}^{-1}\Big] \\ &= \mathbf{B}\mathbf{H}^T\Big[\mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{H}\mathbf{B}\mathbf{B}^{-1}\mathbf{H}^{-1}\mathbf{R}(\mathbf{H}^T)^{-1}\mathbf{H}^T\mathbf{R}^{-1}\Big] \\ &= \mathbf{B}\mathbf{H}^T\Big[\mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{R}^{-1} - \mathbf{R}^{-1}\Big] \\ &= -\mathbf{B}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{R}^{-1}. \end{aligned}$$

Clearly, the two methods are equivalent since the corresponding weight matrices are identical.

2.2.1 Application to the Lorenz-96 model

We proceed on the numerical implementation of the 3D-Var data assimilation for the Lorenz-96 model. We solve the system (1.3) using the classical fourth order Runge-Kutta scheme, which gives

$$\mathbf{x}(t_i) = M_{i-1}[\mathbf{x}(t_{i-1})],$$

where M_{i-1} is the nonlinear model operator that propagates $\mathbf{x}(t_{i-1})$ to $\mathbf{x}(t_i)$. The minimization of the 3D-Var functional defined in (2.51) is carried out using the conjugate gradient method and we perform 200 integration time-steps. The observations are generated from the truth with an error $\boldsymbol{\varepsilon}_o \sim N(0, \sigma_o^2)$, having covariance matrix $\mathbf{R} = E\{\boldsymbol{\varepsilon}_o \boldsymbol{\varepsilon}_o^T\}$.
The 3D-Var method has been coded in Matlab and is based on [19] and [31]. Assuming that the background error covariance matrix $\mathbf{B}$ is constant, the errors of the day are not taken into account. In our test cases we use as background error covariance matrix either $B6h1$ or $Bloc$ that have been both generated in [19]. These covariance matrices are tuned for a 6-hourly observation frequency and observation errors $\boldsymbol{\varepsilon}_o$ with $\sigma_o = 0.15\sigma_{clim} = 0.54$. First, we present results obtained using $B6h1$ and in the sequel results obtained using $Bloc$. The structure of $B6h1$ is displayed in the following figures.

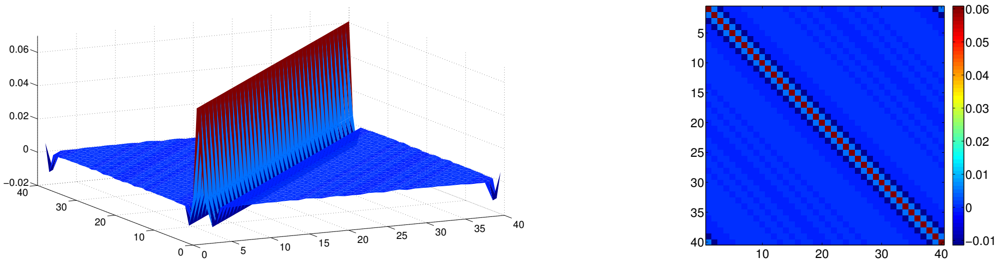


Figure 2.1: Structure of the Background Error Covariance Matrix B6h1.

We assume that we have a perfect model and we consider two different observation networks, each one of them tested for observational error $\boldsymbol{\varepsilon}_o$ that follows a zero-mean normal distribution, i.e., $\boldsymbol{\varepsilon}_o \sim N(0, \sigma_o^2)$, with standard deviation σ_o. The results presenting in the sequel correspond to observational error with $\sigma_o = 0.20$ and are summarized in the following table.

Analysis RMSE for 3D-Var using B6h1		
Assimilation: Network	each time step	every 5 time steps
1. observe all	1.3412	4.1616
2. observe every 2	4.0211	4.7868

Table 2.1: 3D-Var analysis RMSE results using B6h1 matrix and assuming $\boldsymbol{\varepsilon}_o \sim N(0, \sigma_o^2)$ with $\sigma_o = 0.20$.

In our simulations, the quality of the analysis estimate is measured by the root mean square error (RMSE) of the difference between the true and the analysis states, defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i^t - \mathbf{x}_i^a)^2}.$$

In the first example, we consider the first observation network for observations containing error $\varepsilon_o \sim N(0, \sigma_o^2)$, $\sigma_o = 0.2$, which are assimilated into our system at each integration step. In Figure 2.2 we present the evolution in time of the first four components of the true state (green line), the analysis estimate (dashed black line), as well as the available observations (red dots).

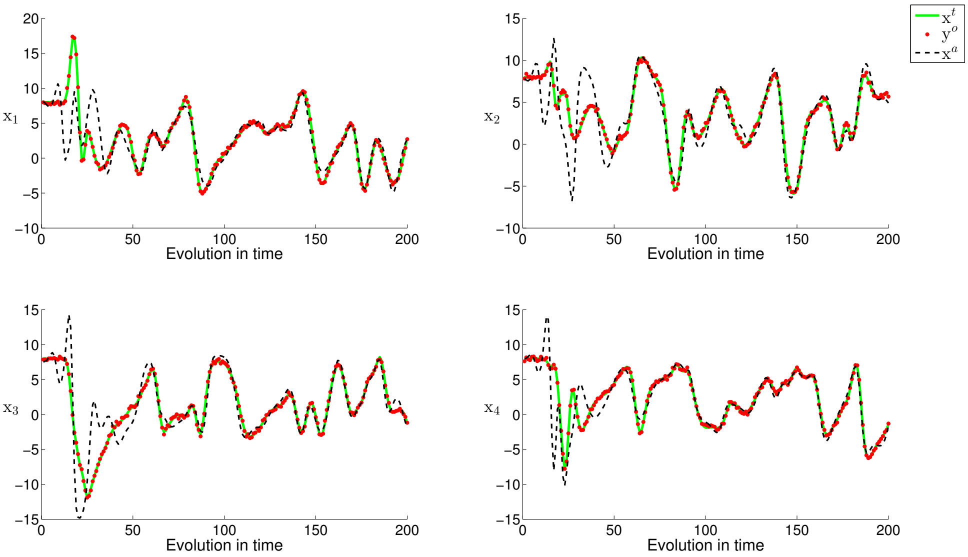


Figure 2.2: 3D-Var for the Lorenz-96 model with observation network 1, $\sigma_o = 0.20$ and assimilation performed at each time step.

For this particular setup, the analysis is a good estimate of the true state. Moreover, in Figure 2.3 we have plotted three elements: In the upper left corner, we present the analysis state obtained after the assimilation compared to the true state, as well as the observations. In the upper right corner, we have the absolute value of the analysis error plotted against the locations at which we have observations. Finally, in the lower part, we present the evolution in time of the analysis RMSE.

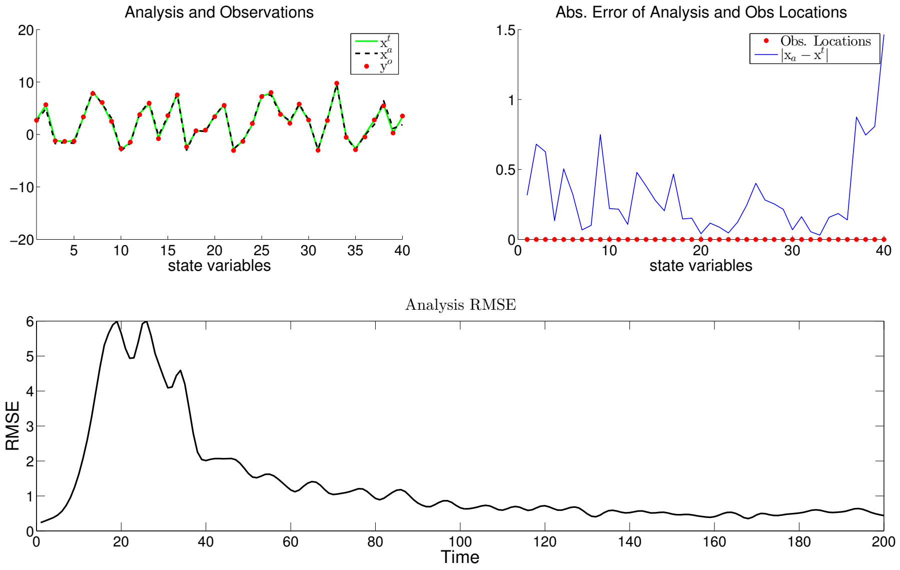


Figure 2.3: 3D-Var for the Lorenz-96 model with observation network 1 and $\sigma_o = 0.20$. The average analysis RMSE is 1.3412.

In Figures 2.4 and 2.5 we present results obtained for the second observation network. As can be seen, the analysis is trying to fit the true state in the components where there are available observations but still, the estimate is not as good as in network 1.

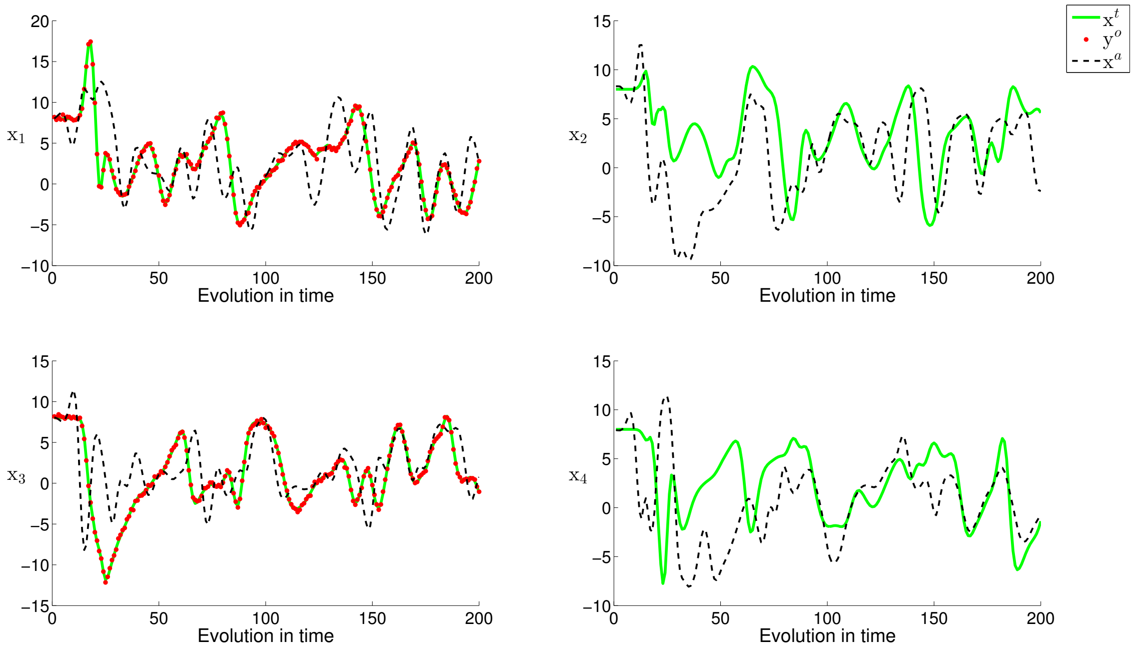


Figure 2.4: 3D-Var for the Lorenz-96 model with observation network 2, $\sigma_o = 0.20$ and assimilation performed at each time step.

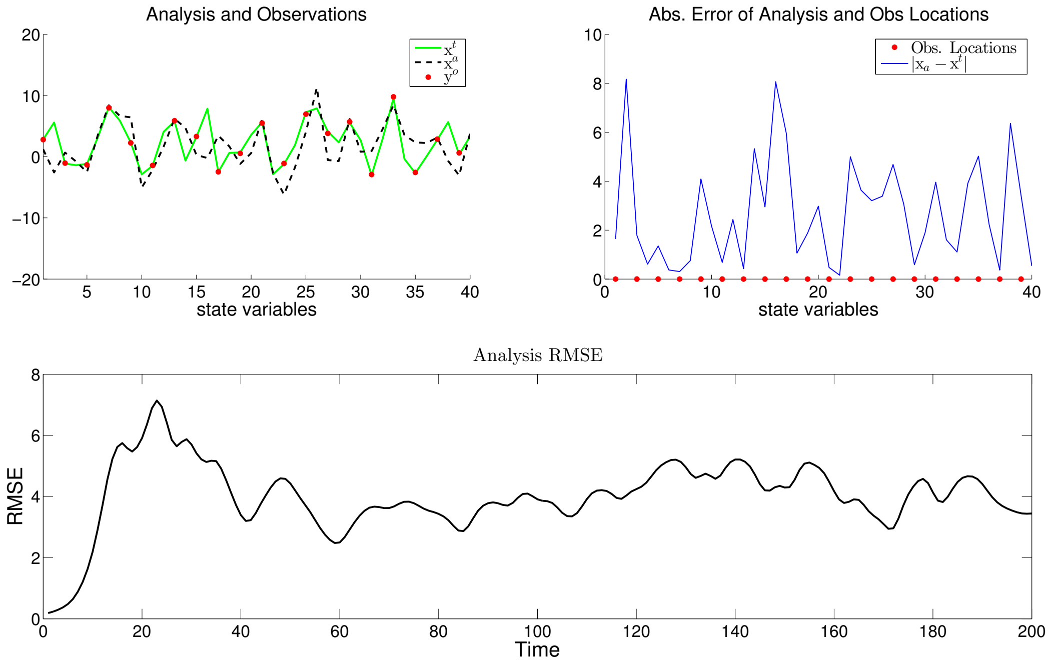


Figure 2.5: 3D-Var for the Lorenz-96 model with observation network 2 and $\sigma_o = 0.20$. The average analysis RMSE is 4.0211.

Looking at Figures 2.3 and 2.5, we observe that for a small observation error 3D-Var gives a good analysis estimate for the observation network 1, whereas in the case were observations are taken every 2 sites, the RMSE is clearly higher. This phenomenon is expected because less information about the true state translates to a poor analysis estimate.

So far, we have presented results of 3D-Var assimilation at each integration step. We consider now that observations are assimilated into our system every 5 integration steps. In Figures 2.6 and 2.7 we have the results for the first observation network, while in Figures 2.8 and 2.9 the results for the second network. In both cases, the analysis produced is not close to the true state, causing a high analysis RMSE.

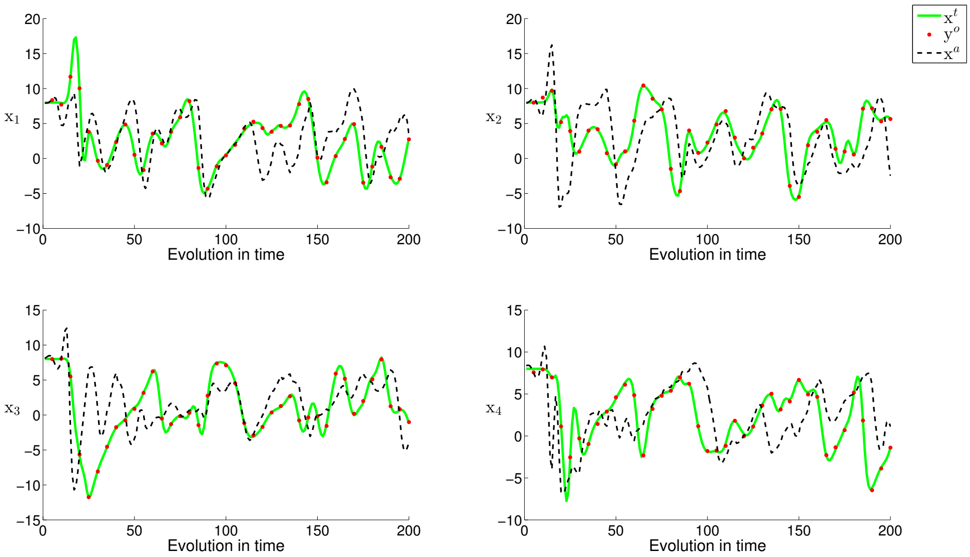


Figure 2.6: 3D-Var for the Lorenz-96 model with observation network 1, $\sigma_o = 0.20$ and assimilation performed every 5 integration steps.

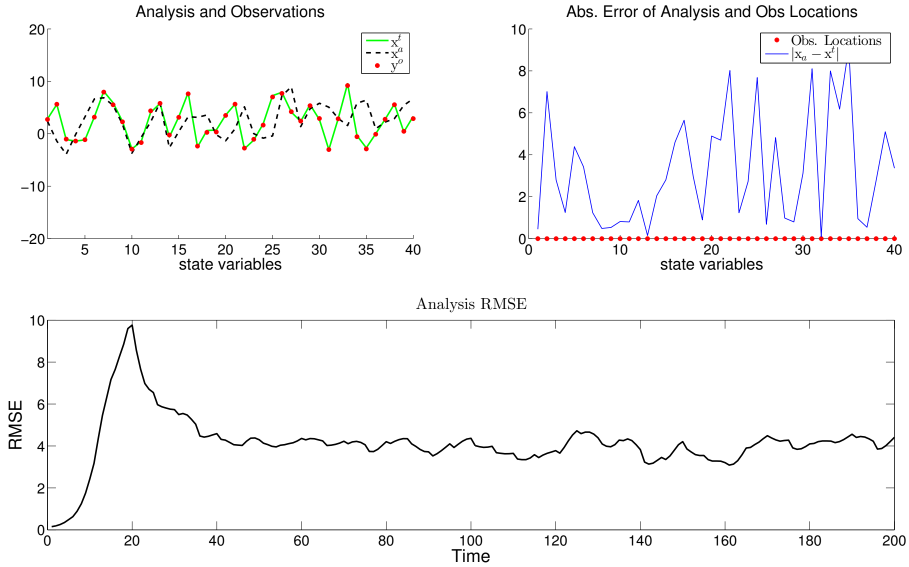


Figure 2.7: 3D-Var for the Lorenz-96 model with observation network 1 and $\sigma_o = 0.20$. The average analysis RMSE is 4.1616.

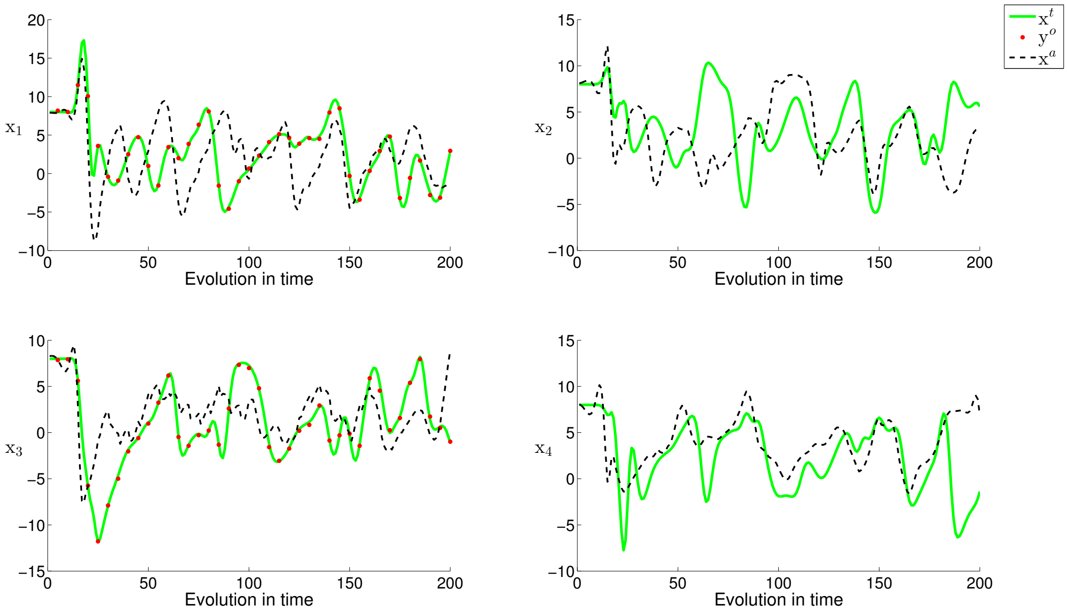


Figure 2.8: 3D-Var for the Lorenz-96 model with observation network 2, $\sigma_o = 0.20$ and assimilation performed every 5 integration steps.

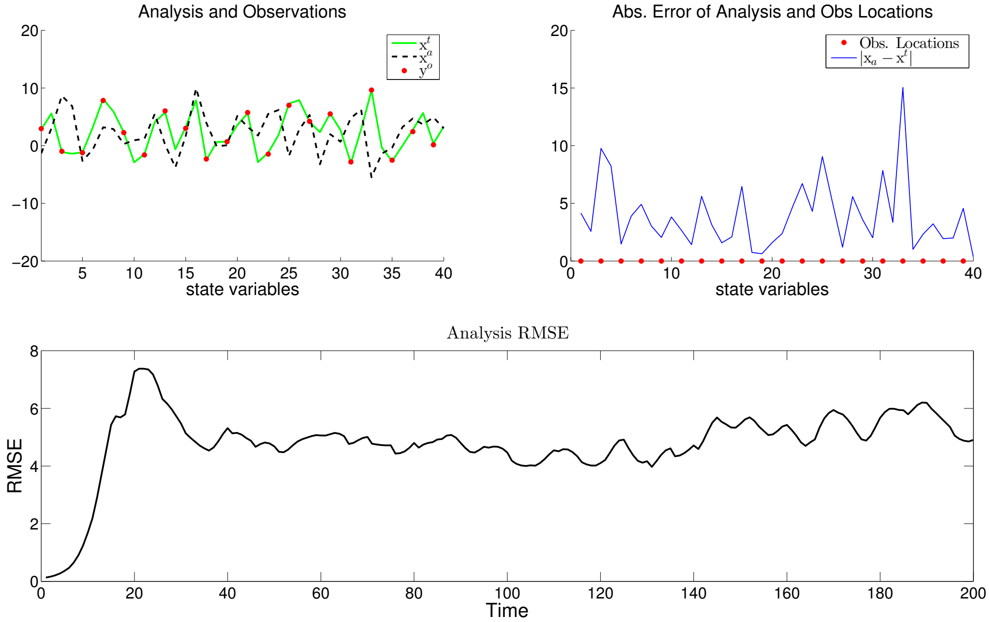


Figure 2.9: 3D-Var for the Lorenz-96 model with observation network 2 and $\sigma_o = 0.20$. The average analysis RMSE is 4.7868.

In the results presented so far, we assumed that the background error covariance had a specific structure. The question arises whether we are able to obtain a better analysis estimate by using another background error covariance. Therefore, we employ the matrix Bloc[1], which is also taken from [19] and is a matrix whose greater values in the main diagonal (variances)

[1] This matrix has been obtained by a LETKF (Local Ensemble Transform Kalman Filter) run for the same setup as B6h1. Refer to [16] for a detailed description of the LETKF.

are localized around the last 20 components. This is known as *covariance localization* and the structure of Bloc is shown in Figure 2.10.

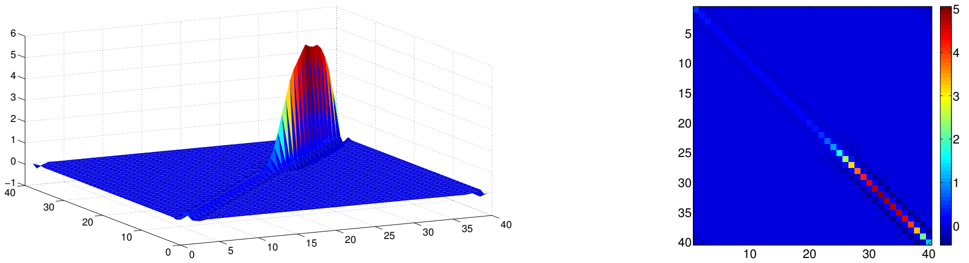


Figure 2.10: Structure of the Background Error Covariance Matrix Bloc.

Once again, we consider the same observation networks, for $\sigma_o = 0.20$ and the results are summarized in the following table.

Analysis RMSE for 3D-Var using Bloc		
Assimilation: Network	each time step	every 5 time steps
1. observe all	0.2257	2.7451
2. observe every 2	1.5791	3.5467

Table 2.2: 3D-Var analysis RMSE results using Bloc matrix and assuming $\varepsilon_o \sim N(0, \sigma_o^2)$ with $\sigma_o = 0.20$.

In Figures 2.11 and 2.12, where observations are available at each location and we have used the localized background error covariance, 3D-Var gives a pretty small RMSE and the analysis estimate coincides with the true state. In comparison with the analogous case using B6h1, the error reduction is significant.

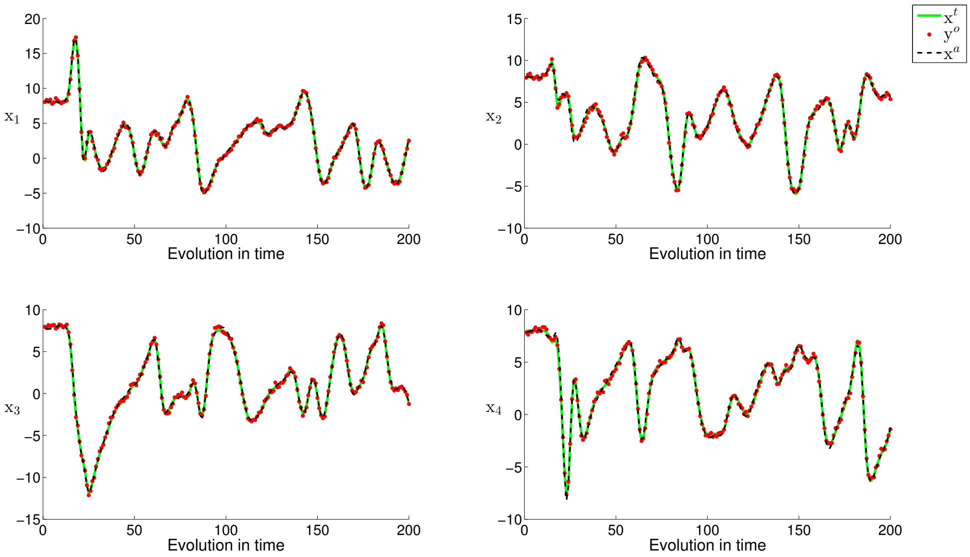


Figure 2.11: 3D-Var for the Lorenz-96 model with observation network 1, $\sigma_o = 0.20$ and assimilation performed at each time step.

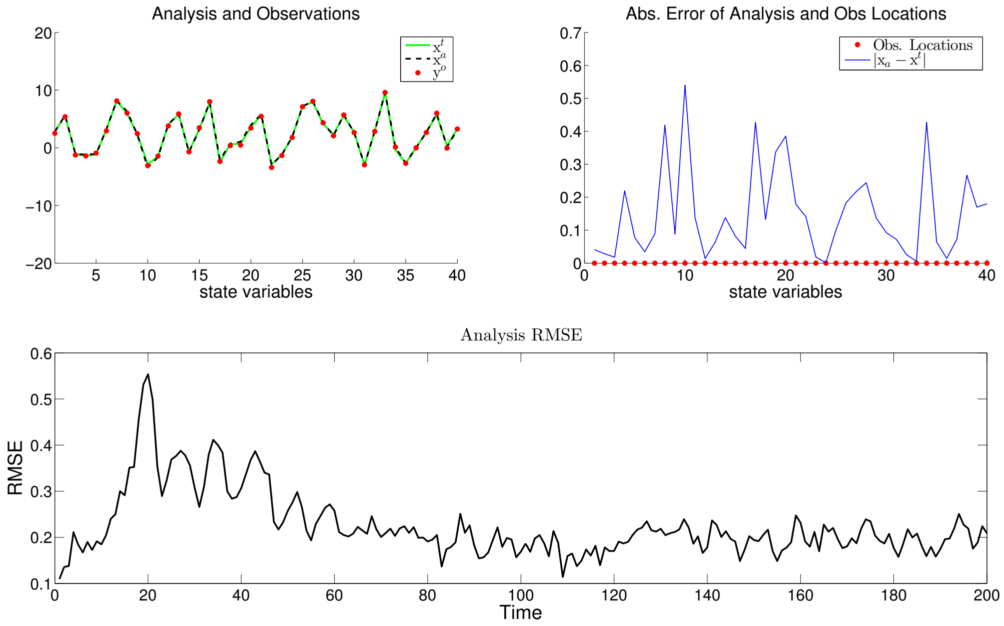


Figure 2.12: 3D-Var for the Lorenz-96 model with observation network 1 and $\sigma_o = 0.20$. The average analysis RMSE is 0.2257.

We continue with the results for the second observation network, in Figures 2.13 and 2.14. As can be seen, the analysis still fits very well the true state, not only at the observed components, but at the unobserved as well.

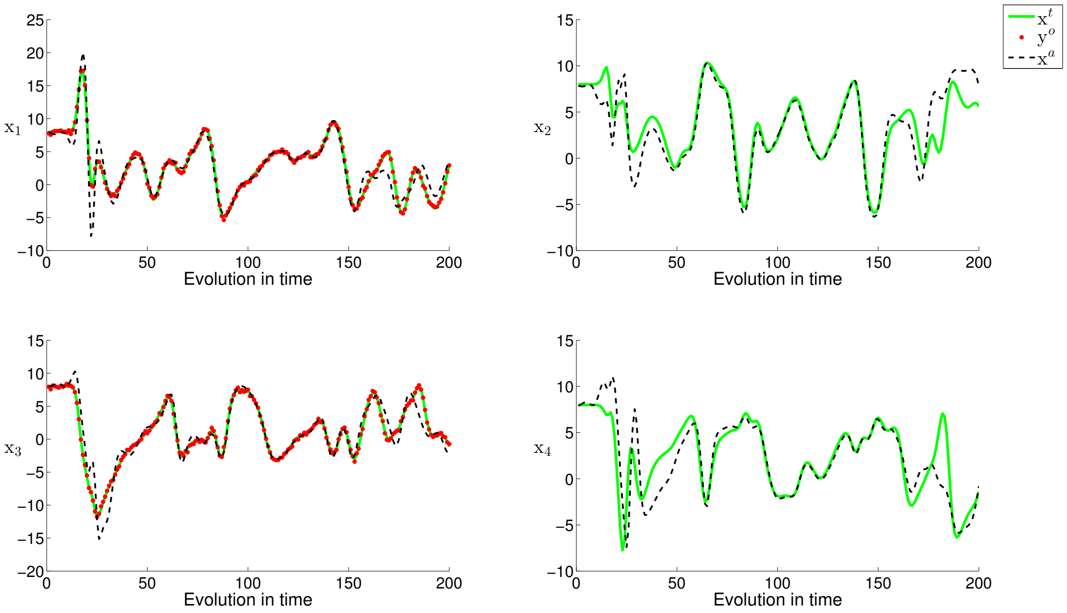


Figure 2.13: 3D-Var for the Lorenz-96 model with observation network 2, $\sigma_o = 0.20$ and assimilation performed at each time step.

Now, if we take a look at the upper right element of Figure 2.14, we observe that the absolute difference between the analysis and the true state is obvious smaller in the last 20 components. This is due to the localization of the background error covariance around these components and moreover, we see that at the observation locations appearing in the same graph, the difference is even smaller than at the intermediate unobserved locations.

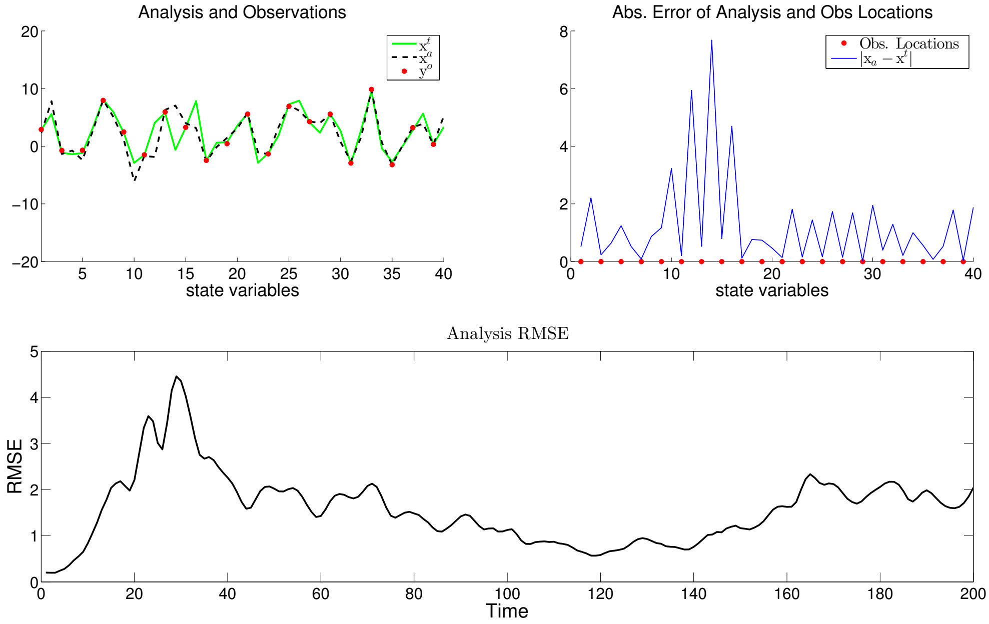


Figure 2.14: 3D-Var for the Lorenz-96 model with observation network 2 and $\sigma_o = 0.20$. The average analysis RMSE is 1.5791.

Assuming now that observations are assimilated into our system every 5 integration steps for the first observation network, we have the results presented in Figures 2.15 and 2.16. The analysis here is a good estimate of the true, having an RMSE 2.7451. Only to compare the results between the use of B6h1 and Bloc, we have to mention that the RMSE in the analogous case using B6h1 was 4.1616.

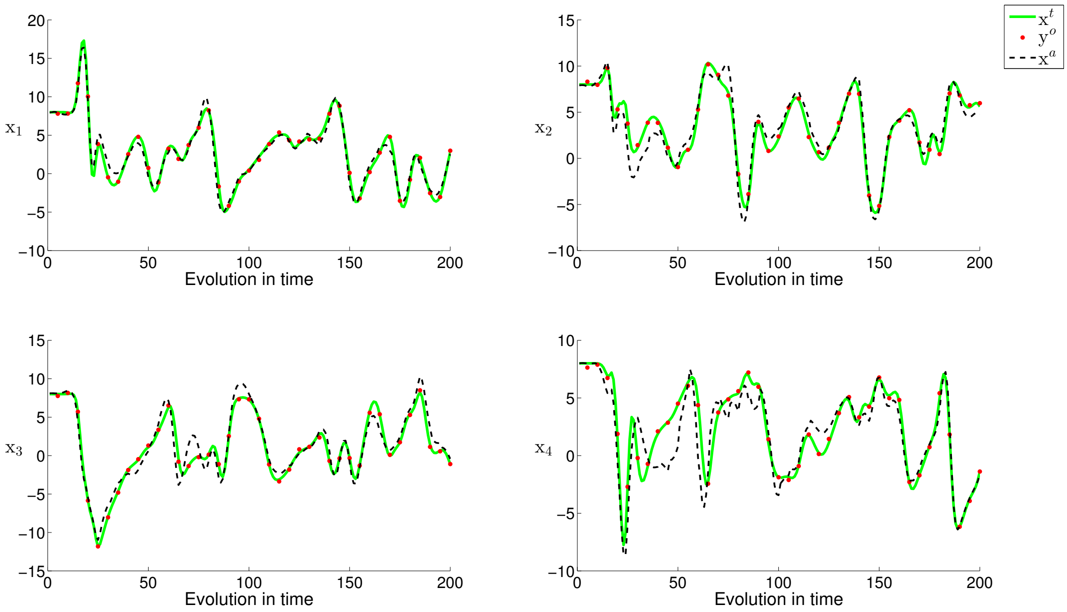


Figure 2.15: 3D-Var for the Lorenz-96 model with observation network 1, $\sigma_o = 0.20$ and assimilation performed every 5 integration steps.

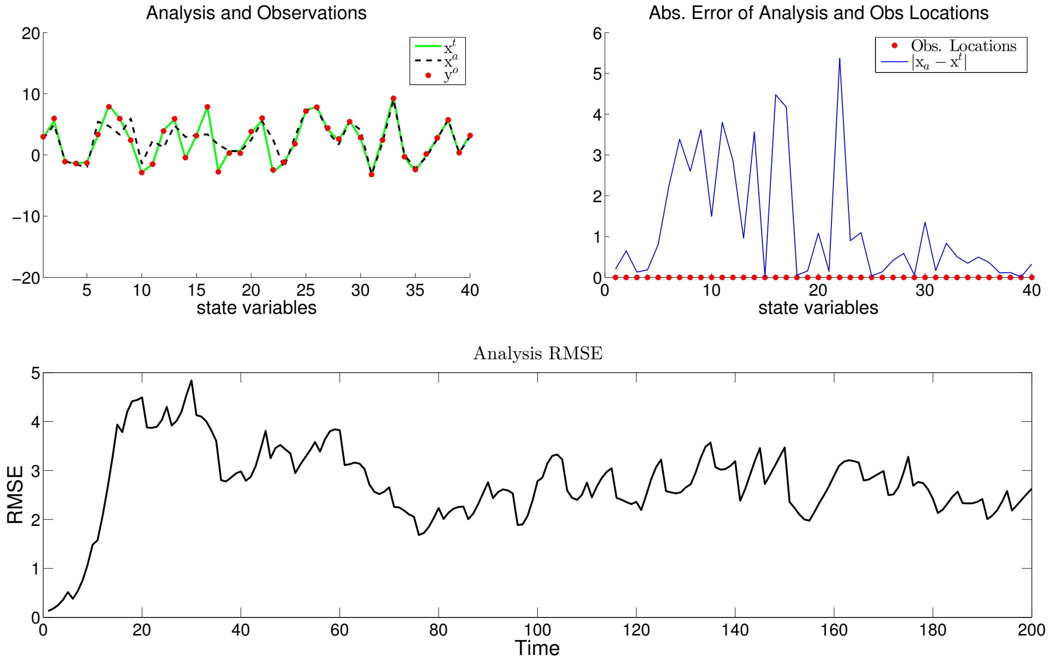


Figure 2.16: 3D-Var for the Lorenz-96 model with observation network 1 and $\sigma_o = 0.20$. The average analysis RMSE is 2.7451.

Considering now the second observation network for assimilation every 5 integration steps, we present the results in Figures 2.17 and 2.18. For the observed components the analysis mimics, in general, the behavior of the true state, while for the unobserved components the difference is quite bigger.

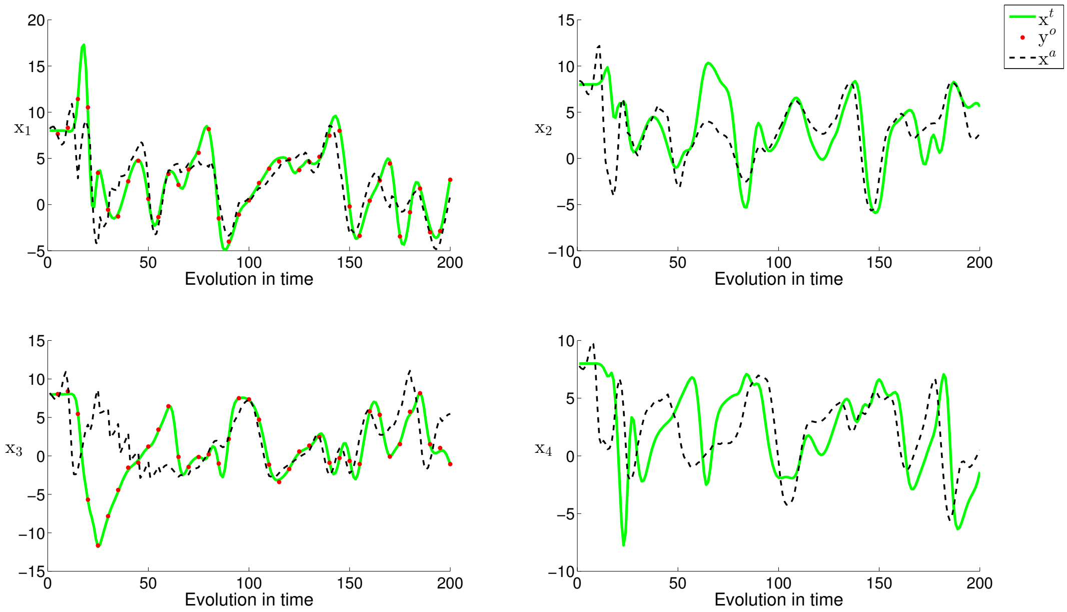


Figure 2.17: 3D-Var for the Lorenz-96 model with observation network 2, $\sigma_o = 0.20$ and assimilation performed every 5 integration steps.

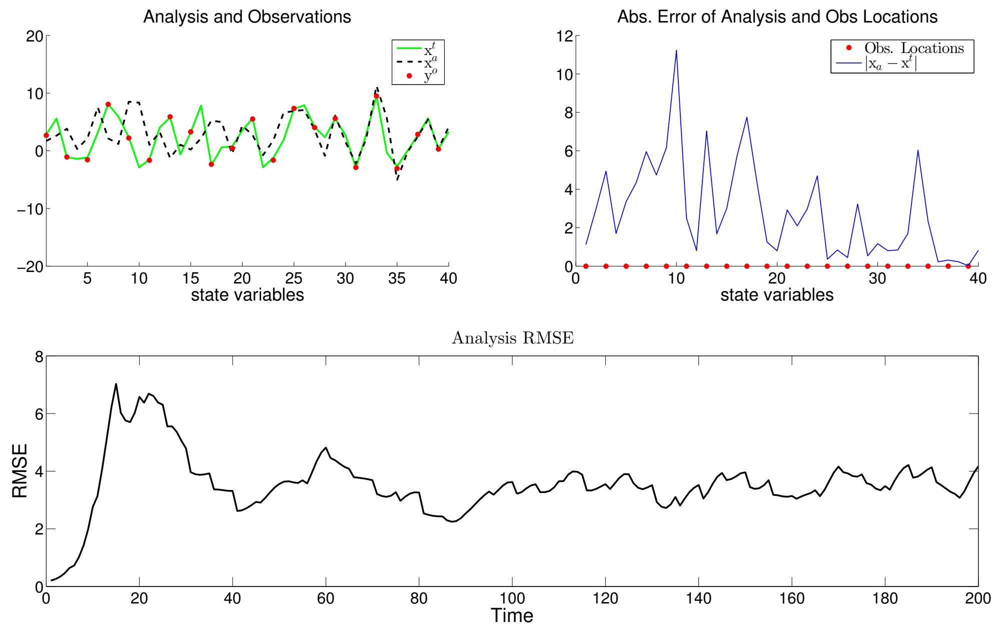


Figure 2.18: 3D-Var for the Lorenz-96 model with observation network 2 and $\sigma_o = 0.20$. The average analysis RMSE is 3.5467.

The conclusion of the examples presented in this section is that 3D-Var has limited ability in finding an optimal analysis estimate. In fact, the choice of the background error covariance is of great importance for the improvement of the 3D-Var estimates.

2.3 Four-Dimensional Variational Assimilation

The idea in 3D-Var was to rewrite the least-squares problem as a minimization of a cost function. The Four-dimensional variational assimilation or 4D-Var is a simple generalization of 3D-Var for observations that are distributed in time. This method seeks the initial condition such that the forecast best fits the observations within the assimilation window.

The 4D-Var cost function includes a term measuring the distance to the background at the beginning of the time interval, together with a sum accounting for the observations collected over a k-hour time window:

$$\begin{aligned} J\left[\mathbf{x}(t_0)\right] &= \frac{1}{2}\left[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)\right]^T \mathbf{B}_0^{-1}\left[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)\right] \\ &+ \frac{1}{2}\sum_{i=0}^{N}\left[H(\mathbf{x}_i) - \mathbf{y}_i^o\right]^T \mathbf{R}_i^{-1}\left[H(\mathbf{x}_i) - \mathbf{y}_i^o\right]. \end{aligned} \tag{2.62}$$

The control variable (i.e., the variable with respect to which the cost function is minimized) is the *initial* state of the model with the time interval $\mathbf{x}(t_0)$ and the analysis at the end of the interval is given by the *model integration* from the solution $\mathbf{x}(t_n) = M_n[\mathbf{x}(t_0)]$. This means that the analysis has to satisfy the model equations.

The 4D-Var functional can be written in the form $J = J_b + J_o$. Then, we proceed to the minimization of the two individual functions. The first part,

$$J_b\left[\mathbf{x}(t_0)\right] = \frac{1}{2}\left[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)\right]^T \mathbf{B}_0^{-1}\left[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)\right], \tag{2.63}$$

is the background component of the cost function and its gradient with respect to $\mathbf{x}(t_0)$ is

$$\frac{\partial J_b}{\partial \mathbf{x}(t_0)} = \mathbf{B}_0^{-1}\left[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)\right]. \tag{2.64}$$

The second part is the observation component

$$J_o\left[\mathbf{x}(t_0)\right] = \frac{1}{2}\sum_{i=0}^{N}\left[H(\mathbf{x}_i) - \mathbf{y}_i^o\right]^T \mathbf{R}_i^{-1}\left[H(\mathbf{x}_i) - \mathbf{y}_i^o\right] \tag{2.65}$$

and it is more complicated because of the nonlinear $\mathbf{x}_i = M_i[\mathbf{x}(t_0)]$.
As we saw previously in Remark 2.2.1, given a symmetric matrix $\mathbf{A}$ and a quadratic functional $J(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x}$, its gradient is given by $\partial J/\partial \mathbf{x} = \mathbf{A}\mathbf{x}$. If the functional is $J(\mathbf{x}) = \mathbf{z}^T\mathbf{A}\mathbf{z}$ and $\mathbf{z} = \mathbf{z}(\mathbf{x})$, then the gradient is given by

$$\frac{\partial J}{\partial \mathbf{x}} = \left[\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right]^T \mathbf{A}\mathbf{z}, \qquad \text{where} \quad \left[\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right]_{k,l} = \frac{\partial z_k}{\partial x_l}. \tag{2.66}$$

In our case, we have to compute the gradient of $\mathbf{z} = H(\mathbf{x}_i) - \mathbf{y}_i^o$ with respect to $\mathbf{x}(t_0)$. We begin from the nonlinear $\mathbf{x}_i = M_i[\mathbf{x}(t_0)]$ and we introduce a perturbation to the initial state. Then, the linearized form is $\delta\mathbf{x}_i = \mathbf{L}(t_0, t_i)\ \delta\mathbf{x}_0$, where $\mathbf{L}(t_0, t_i)$ is the *tangent linear model* (see Definition 2.3.1) that advances the initial perturbation from t_0 to t_i.

Definition 2.3.1. A *tangent linear model* (TLM) is obtained by linearizing the model about the nonlinear trajectory of the model between t_{i-1} and t_i. So, if we introduce a perturbation in the initial conditions then, the final perturbation is

$$\mathbf{x}(t_i) + \delta\mathbf{x}(t_i) = M_{i-1}[x(t_{i-1}) + \delta\mathbf{x}(t_{i-1})] = M_{i-1}[\mathbf{x}(t_{i-1})] + \mathbf{L}_{i-1}\delta\mathbf{x}(t_{i-1}) + O(|\delta\mathbf{x}|^2).$$

where $\mathbf{L}_{i-1}$ is the TLM that transforms the initial perturbation at time t_{i-1} to the final time t_i. Therefore, the TLM expression is $\delta\mathbf{x}(t_i) = \mathbf{L}_{i-1}\ \delta\mathbf{x}_{t_{i-1}}$. If there are several steps in a time interval $[t_0, t_i]$, the TLM that advances a perturbation from t_0 to t_i is given by the product of the TLM matrices that advance it over each step:

$$\mathbf{L}(t_0, t_i) = \prod_{j=i-1}^{0}\mathbf{L}(t_j, t_{j+1}) = \prod_{j=i-1}^{0}\mathbf{L}_j = \mathbf{L}_{i-1}\mathbf{L}_{i-2}\cdot\cdots\mathbf{L}_0.$$

Moreover, the *adjoint model* i.e., the transpose of the tangent linear model, is given by

$$\mathbf{L}^T(t_i, t_0) = \prod_{j=0}^{i-1}\mathbf{L}^T(t_{j+1}, t_j) = \prod_{j=0}^{i-1}\mathbf{L}_j^T = \mathbf{L}_0^T\mathbf{L}_1^T\cdots\mathbf{L}_{i-1}^T,$$

which means that the adjoint model "advances" a perturbation backward in time, from the final to the initial time.

Therefore, the gradient of $H(\mathbf{x}_i) - \mathbf{y}_i^o$ with respect to $\mathbf{x}(t_0)$ can be written as

$$\frac{\partial[H(\mathbf{x}_i) - \mathbf{y}_i^o]}{\partial \mathbf{x}(t_0)} = \frac{\partial H}{\partial \mathbf{x}_i}\frac{\partial M}{\partial \mathbf{x}_0} = \mathbf{H}_i\ \mathbf{L}(t_0, t_i) = \mathbf{H}_i\prod_{j=i-1}^{0}\mathbf{L}(t_j, t_{j+1}), \tag{2.67}$$

where $\mathbf{H}_i$ and $\mathbf{L}_i$ are the linearized Jacobian matrices $\partial H/\partial x_i$ and $\partial M/\partial x_0$, respectively. Hence, using (2.66) and (2.67), the gradient of the observation component is

$$\frac{\partial J_o}{\partial \mathbf{x}(t_0)} = \sum_{i=0}^{N} \mathbf{L}^T(t_i, t_0)\mathbf{H}_i^T \mathbf{R}_i^{-1}[H(\mathbf{x}_i) - \mathbf{y}_i^o]. \tag{2.68}$$

Finally, the total gradient of the 4D-Var functional is given by

$$\frac{\partial J}{\partial \mathbf{x}(t_0)} = \mathbf{B}_0^{-1}\left[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)\right] + \sum_{i=0}^{N} \mathbf{L}^T(t_i, t_0)\mathbf{H}_i^T \mathbf{R}_i^{-1}[H(\mathbf{x}_i) - \mathbf{y}_i^o]. \tag{2.69}$$

We wish to find the minimum of the cost function and as we have mentioned before, this can be done using an iterative minimization algorithm. At this point, we must note that the gradient of the observation component in (2.68) shows that every iteration of the 4D-Var minimization requires the computation of the gradient. Thus, every iteration implies that we must compute the increments $[H(\mathbf{x}_i) - \mathbf{y}_i^o]$ at the observation times t_i during a forward integration. Then, we have to multiply them by $\mathbf{H}_i^T \mathbf{R}_i$, and finally, integrate the resulting weighted increments back to the initial time, using the adjoint model.

Chapter 3

Kalman Filtering

Kalman Filtering (KF) is a sequential technique for estimating the state of a dynamical system from a set of noisy measurements. The measurements need not be those of the state variables themselves, but must be related to them through a functional that can be linearized. The filter is named after Rudolf E. Kalman, who, in 1960, published his now famous article describing a recursive solution to the discrete-data linear filtering problem [17].
The Kalman filter is optimal, in the sense that it minimizes the variance of the estimate's error, for linear models with additive independent white noise in both the model and the measurement systems. Although it was designed for linear problems, a new version was soon discovered, known as the *Extended Kalman Filter* (EKF), which has been used ever since for nonlinear applications of Kalman filtering.
In meteorology, Kalman filters are widely used to improve the prediction of the variables of interest. Actually, most of the times in NWP models both the observation operator H and the forward model M are nonlinear. Therefore, we focus on the Extended version. As seen in the sequel, these operators can be linearized using Taylor series expansions around the current predicted state.

3.1 Extended Kalman Filter

In the sequential data assimilation at time t_i we have at our disposal the outcome of a previous forecast, denoted by $\mathbf{x}^f(t_i)$. Therefore, $\mathbf{x}^f(t_i)$ is the analogue of the background $\mathbf{x}^b$ found in Optimal Interpolation. At time t_i we collect a set of observations which are arranged into the vector $\mathbf{y}_i^o$. Given the forecast $\mathbf{x}^f(t_i)$ and the observations $\mathbf{y}_i^o$, we perform an analysis to obtain the state estimate $\mathbf{x}^a(t_i)$. Then, we advance $\mathbf{x}^a(t_i)$ from time t_i to t_{i+1} using the model dynamics and create the new forecast at time t_{i+1}, denoted by $\mathbf{x}^f(t_{i+1})$. The last one will serve as the background in the next cycle, and we continue the same process iteratively.
Although, the Kalman filter algorithm is very similar to Optimal Interpolation, there is a main difference: in OI we assume that the background error covariance $\mathbf{B}$ is a constant matrix, while in KF we update the forecast error covariance $\mathbf{P}^f(t_i)$ at every forecast step using the model.

The forecast state is advanced from the previous analysis time t_{i-1} to the current time t_i, through the nonlinear forecast model

$$\mathbf{x}^f(t_i) = M_{i-1}[\mathbf{x}^a(t_{i-1})]. \tag{3.1}$$

In the sequel of Kalman Filtering, our basic assumption is that the errors involved in the estimates and the observations are following zero-mean normal distributions.

The model may not be perfect, therefore we assume that the true state of the atmosphere is given by

$$\mathbf{x}^t(t_i) = M_{i-1}[\mathbf{x}^t(t_{i-1})] + \eta(t_{i-1}), \tag{3.2}$$

where $\eta(t_{i-1})$ is a zero-mean noise process with covariance matrix $\mathbf{Q}_{i-1} = E\{\eta_{i-1}\eta_{i-1}^T\}$, where $\eta_{i-1} = \eta(t_{i-1})$.

The forecast error covariance can be obtained by linearizing the model about the nonlinear trajectory, between two consecutive time-steps t_{i-1} and t_i. Therefore, we introduce a perturbation $\delta\mathbf{x}(t_i)$ in the initial conditions and we have

$$\begin{aligned} \mathbf{x}(t_i) + \delta\mathbf{x}(t_i) &= M_{i-1}[\mathbf{x}(t_{i-1}) + \delta\mathbf{x}(t_{i-1})] \\ &= M_{i-1}[\mathbf{x}(t_{i-1})] + \mathbf{L}_{i-1}\delta\mathbf{x}(t_{i-1}) + O(|\delta\mathbf{x}|^2), \end{aligned} \tag{3.3}$$

where $\mathbf{L}$ is the tangent linear model, i.e., the matrix which transforms the initial perturbation at time t_{i-1} to the final perturbation at time t_i.
Once again, we assume that the observations contain errors with zero mean and error covariance matrix $\mathbf{R}_i = E\{\varepsilon_i^o \varepsilon_i^{oT}\}$ and are given by

$$\mathbf{y}_i^o = H[\mathbf{x}^t(t_i)] + \varepsilon_i^o, \tag{3.4}$$

where H is the (nonlinear) observation operator.

The error over a forecast period depends on the initial error and the errors arising from the forecast model. Using (3.1) and (3.2), we have

$$\begin{aligned} \varepsilon_i^f &= \mathbf{x}^t(t_i) - \mathbf{x}^f(t_i) \\ &= M_{i-1}[\mathbf{x}^t(t_{i-1})] + \eta(t_{i-1}) - M_{i-1}[\mathbf{x}^a(t_{i-1})] \\ &= M_{i-1}[\mathbf{x}^a(t_{i-1}) + \mathbf{x}^t(t_{i-1}) - \mathbf{x}^a(t_{i-1})] + \eta(t_{i-1}) - M_{i-1}[\mathbf{x}^a(t_{i-1})] \\ &\approx \cancel{M_{i-1}[\mathbf{x}^a(t_{i-1})]} + \mathbf{L}_{i-1}\varepsilon_{i-1}^a + \eta(t_{i-1}) - \cancel{M_{i-1}[\mathbf{x}^a(t_{i-1})]}, \end{aligned}$$

where the last expression is the result of a first-order Taylor series expansion around $\mathbf{x}^a(t_{i-1})$ neglecting higher order terms. Therefore, we arrive in the equation

$$\varepsilon_i^f \approx \mathbf{L}_{i-1}\varepsilon_{i-1}^a + \eta_{i-1}. \tag{3.5}$$

Assuming that the forecast error ε_i^f has mean zero, we continue with the derivation of the forecast error covariance matrix, which is given by:

$$\begin{aligned} \mathbf{P}^f(t_i) &= E\left\{\varepsilon_i^f \varepsilon_i^{fT}\right\} \\ &= E\left\{(\mathbf{L}_{i-1}\varepsilon_{i-1}^a + \eta_{i-1})(\mathbf{L}_{i-1}\varepsilon_{i-1}^a + \eta_{i-1})^T\right\} \\ &= E\left\{(\mathbf{L}_{i-1}\varepsilon_{i-1}^a + \eta_{i-1})(\varepsilon_{i-1}^{a\,T}\mathbf{L}_{i-1}^T + \eta_{i-1}^T)\right\} \\ &= E\left\{\mathbf{L}_{i-1}\varepsilon_{i-1}^a\varepsilon_{i-1}^{a\,T}\ \mathbf{L}_{i-1}^T + \mathbf{L}_{i-1}\varepsilon_{i-1}^a\eta_{i-1}^T + \eta_{i-1}\varepsilon_{i-1}^{a\,T}\ \mathbf{L}_{i-1}^T + \eta_{i-1}\eta_{i-1}^T\right\} \\ &= \mathbf{L}_{i-1}\ E\{\varepsilon_{i-1}^a\varepsilon_{i-1}^{a\,T}\}\ \mathbf{L}_{i-1}^T + \mathbf{L}_{i-1}\ E\{\varepsilon_{i-1}^a\eta_{i-1}^T\} + E\{\eta_{i-1}\varepsilon_{i-1}^{a\,T}\}\ \mathbf{L}_{i-1}^T + E\{\eta_{i-1}\eta_{i-1}^T\}. \end{aligned}$$

The terms that contain the expected value of the model and the analysis errors are zero since they are uncorrelated. Moreover, the term $E\{\varepsilon_{i-1}^a\varepsilon_{i-1}^{a\,T}\}$ is actually the analysis error covariance matrix $\mathbf{P}^a(t_{i-1})$ at time t_{i-1} (as defined in (2.43)) and can be derived as in OI.

Therefore,

$$\mathbf{P}^f(t_i) = \mathbf{L}_{i-1}\ \mathbf{P}^a(t_{i-1})\ \mathbf{L}_{i-1}^T + \mathbf{Q}_{i-1}. \tag{3.6}$$

After completing the forecast step at time t_i, the innovation vector is

$$\begin{aligned}\mathbf{d}_i &= \mathbf{y}_i^o - H[\mathbf{x}^f(t_i)]\\ &= \mathbf{y}_i^o - H[\mathbf{x}^t(t_i) + \mathbf{x}^f(t_i) - \mathbf{x}^t(t_i)]\\ &= \mathbf{y}_i^o - H[\mathbf{x}^t(t_i)] - \mathbf{H}_i[\mathbf{x}^f(t_i) - \mathbf{x}^t(t_i)]\\ &= \boldsymbol{\varepsilon}_i^o - \mathbf{H}_i\boldsymbol{\varepsilon}_i^f.\end{aligned} \tag{3.7}$$

Moreover, the optimal weight matrix or *Kalman gain* $\mathbf{K}_i$, that minimizes the analysis error covariance $\mathbf{P}_i^a$, can be found following the same process as in OI for the computation of (2.46).

$$\begin{aligned}\mathbf{K} &= E\left\{(\mathbf{x}^t - \mathbf{x}^f)[\mathbf{y}^o - H(\mathbf{x}^f)]^T\right\}\left[E\left\{[\mathbf{y}^o - H(\mathbf{x}^f)][\mathbf{y}^o - H(\mathbf{x}^f)]^T\right\}\right]^{-1}\\ &= E\left\{(-\boldsymbol{\varepsilon}^f)[\boldsymbol{\varepsilon}^o - \mathbf{H}\boldsymbol{\varepsilon}^f]^T\right\}\left[E\left\{[\boldsymbol{\varepsilon}^o - \mathbf{H}\boldsymbol{\varepsilon}^f][\boldsymbol{\varepsilon}^o - \mathbf{H}\boldsymbol{\varepsilon}^f]^T\right\}\right]^{-1}\\ &= E\left\{-\boldsymbol{\varepsilon}^f(\boldsymbol{\varepsilon}^o)^T + \boldsymbol{\varepsilon}^f(\boldsymbol{\varepsilon}^f)^T\mathbf{H}^T\right\}\left[E\left\{\boldsymbol{\varepsilon}^o(\boldsymbol{\varepsilon}^o)^T - \boldsymbol{\varepsilon}^o(\boldsymbol{\varepsilon}^f)^T\mathbf{H}^T - \mathbf{H}\boldsymbol{\varepsilon}^f(\boldsymbol{\varepsilon}^o)^T + \mathbf{H}\boldsymbol{\varepsilon}^f(\boldsymbol{\varepsilon}^f)^T\mathbf{H}^T\right\}\right]^{-1}\\ &= E\left\{\boldsymbol{\varepsilon}^f(\boldsymbol{\varepsilon}^f)^T\right\}\ \mathbf{H}^T + \left[E\left\{\boldsymbol{\varepsilon}^o(\boldsymbol{\varepsilon}^o)^T\right\} + \mathbf{H}\ E\left\{\boldsymbol{\varepsilon}^f(\boldsymbol{\varepsilon}^f)^T\right\}\mathbf{H}^T\right]^{-1}\\ &= \mathbf{P}^f\ \mathbf{H}^T\left[\mathbf{R} + \mathbf{H}\ \mathbf{P}^f\ \mathbf{H}^T\right]^{-1}.\end{aligned}$$

Thus, the Kalman gain, after completing the forecast step i, is given by

$$\mathbf{K}_i = \mathbf{P}^f(t_i)\ \mathbf{H}_i^T\left[\mathbf{R}_i + \mathbf{H}_i\ \mathbf{P}^f(t_i)\ \mathbf{H}_i^T\right]^{-1}. \tag{3.8}$$

The analysis state and its error covariance can be written as in OI, using the calculated $\mathbf{P}^f(t_i)$ and $\mathbf{K}_i$ matrices, instead of $\mathbf{B}$ and $\mathbf{W}$, respectively.

Therefore, we have that

$$\mathbf{x}^a(t_i) = \mathbf{x}^f(t_i) + \mathbf{K}_i\left[\mathbf{y}_i^o - H[\mathbf{x}^f(t_i)]\right], \tag{3.9}$$

$$\mathbf{P}^a(t_i) = (\mathbf{I} - \mathbf{K}_i\mathbf{H}_i)\mathbf{P}^f(t_i). \tag{3.10}$$

Under the assumption of normally distributed errors, $\mathbf{x}^a$ given by (3.9) is the mean value of the distribution of the true state $\mathbf{x}^t$ at time t_i, which means that it is the optimal estimate of the state.

We summarize the recursive algorithm of the Extended Kalman Filter consisting of two steps: the "*forecast step*" that advances the forecast state and its error covariance matrix and the "*analysis step*" that updates the analysis state and the corresponding error covariance.

Extended Kalman Filter Algorithm

1. Input

   System state $\mathbf{x}^a(t_0) = \mathbf{x}_0$ and error covariance matrix $\mathbf{P}^a(t_0) = \mathbf{P}_0$.

2. For $i = 1, 2, \ldots$

   - *Forecast Step:*

   $$\begin{aligned} \mathbf{x}^f(t_i) &= M_{i-1}[\mathbf{x}^a(t_{i-1})] \\ \mathbf{P}^f(t_i) &= \mathbf{L}_{i-1}\mathbf{P}^a(t_{i-1})\mathbf{L}_{i-1}^T + \mathbf{Q}_{i-1} \end{aligned}$$

   - *Analysis Step:*

   $$\begin{aligned} \mathbf{K}_i &= \mathbf{P}^f(t_i)\mathbf{H}_i^T \left[\mathbf{R}_i + \mathbf{H}_i\ \mathbf{P}^f(t_i)\ \mathbf{H}^T\right]^{-1} \\ \mathbf{x}^a(t_i) &= \mathbf{x}^f(t_i) + \mathbf{K}_i \left[\mathbf{y}_i^o - H[\mathbf{x}^f(t_i)]\right] \\ \mathbf{P}^a(t_i) &= [\mathbf{I} - \mathbf{K}_i\mathbf{H}_i]\,\mathbf{P}^f(t_i) \end{aligned}$$

A great property of the Extended Kalman Filter is that, even if a system starts with a poor initial guess of the state of the atmosphere, the EKF may go through a transient period, after which it should provide the best linear unbiased estimate of the state and its error covariance. However, there are limitations in its application. The error propagation is approximated by the tangent linear model between two analysis steps. Therefore, if the time-step between two consecutive updates is long enough and in combination with infrequent observations, it may lead to a coarse approximation of the forecast error covariance and hence, to the divergence of the filter. Another problem is the computational cost of the EKF in high-dimensional problems. The TLM matrix $\mathbf{L}_{i-1}$ has size n, i.e., the number of degrees of freedom of the model (in modern models it is more than 10^6) and the update of the error covariance is equivalent to performing $\mathcal{O}(n)$ model integrations.

3.2 Ensemble Kalman Filtering

The propagation of the covariance information is the main feature of the Kalman Filter. It is also its main challenge since the matrices $\mathbf{P}^f$ and $\mathbf{P}^a$ have dimension $n \times n$, which for large n, their computation time and storage may not be feasible. Therefore, the key is to find a reasonable approximation of these covariance matrices with less computational cost. This leads to the *Ensemble Kalman Filter* (EnKF), a simplification of the standard Kalman filtering, first introduced by Evensen in [7]. It has gained popularity because of its simple implementation. It does not require the derivation of the tangent linear operator or integrations backward in time. It is an approximation of the EKF which avoids evolving the error covariance matrix at every time step. Instead, an *ensemble* of K data assimilation cycles is used to estimate the forecast uncertainty. The goal of an EnKF is to generate an analysis ensemble which reflects both an estimate of the true atmospheric state (through its mean) and its uncertainty (through its spread).

Each ensemble member evolves independently from the others according to the forecast model. When new observations become available, the entire ensemble is adjusted in order to take into account both the new state estimate and the uncertainty dictated by the observations. This adjustment indicates the analysis step of the algorithm. Therefore, the analysis ensemble mean is formed as a weighted average of the forecast (or background) ensemble mean and the observations, where the weights are determined from the background and observation uncertainties.

We start with an ensemble $\{\mathbf{x}_k^a,\ k = 1, \ldots, K\}$ consisting of K members at time t_{i-1}, where each member is an n-dimensional model state vector. Evolving each ensemble member according to the nonlinear forecast model

$$\mathbf{x}_k^f(t_i) = M_{i-1}[\mathbf{x}_k^a(t_{i-1})], \qquad k = 1, \ldots, K, \tag{3.11}$$

we obtain the forecast ensemble $\mathbf{x}_k^f(t_i)$ at time t_i.

The analysis assumes that the best available estimate to the system state, before the observations are taken into account, is the background ensemble mean

$$\overline{\mathbf{x}}^f = \frac{1}{K}\sum_{k=1}^{K} \mathbf{x}_k^f. \tag{3.12}$$

We define the background ensemble perturbations matrix $\mathbf{X}^f$, whose k-th column is defined as $(\mathbf{x}_k^f - \overline{\mathbf{x}}^f)/\sqrt{K-1},\ k = 1, \ldots, K$, i.e.,

$$\mathbf{X}^f = \frac{1}{\sqrt{K-1}}\left(\mathbf{x}_1^f - \overline{\mathbf{x}}^f, \ldots, \mathbf{x}_K^f - \overline{\mathbf{x}}^f\right). \tag{3.13}$$

Then, the uncertainty in the state estimate is described by the background error covariance matrix

$$\mathbf{P}^f = \mathbf{X}^f(\mathbf{X}^f)^T. \tag{3.14}$$

The analysis must determine a state estimate $\overline{\mathbf{x}}^a$, an error covariance matrix $\mathbf{P}^a$ and an ensemble $\{\mathbf{x}_k^a,\ k = 1, \ldots, K\}$ with sample mean

$$\overline{\mathbf{x}}^a = \frac{1}{K}\sum_{k=1}^{K} \mathbf{x}_k^a. \tag{3.15}$$

and covariance matrix

$$\mathbf{P}^a = \mathbf{X}^a(\mathbf{X}^a)^T, \tag{3.16}$$

where $\mathbf{X}^a$ is the $n \times K$ matrix of the analysis ensemble perturbations, defined as

$$\mathbf{X}^a = \frac{1}{\sqrt{K-1}}\left(\mathbf{x}_1^a - \overline{\mathbf{x}}^a, \ldots, \mathbf{x}_K^a - \overline{\mathbf{x}}^a\right). \tag{3.17}$$

Filters that use perturbed observation ensembles (i.e., the perturbed observations ensemble is created by adding random vectors to the actual observations for each member) such as the EnKF, are known as Stochastic filters, whereas those that do not use perturbed observations are known as Deterministic filters. In the sequel, we consider the deterministic filter, also called *Ensemble Square Root Kalman Filter* (EnSRF).

Based on the analysis step (equations (3.8) to (3.10)) of the standard Kalman Filter, the analysis ensemble mean should satisfy the equation

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{K}[\mathbf{y}^o - H(\overline{\mathbf{x}}^f)], \tag{3.18}$$

where $\mathbf{K}$ is the *ensemble Kalman Gain.* This is because the background ensemble error covariance matrix $\mathbf{P}^f$ is an approximation of the full background error covariance, defined as

$$\mathbf{K} = \mathbf{X}^f(\mathbf{X}^f)^T\mathbf{H}^T\left[\mathbf{R} + \mathbf{H}\mathbf{X}^f(\mathbf{X}^f)^T\mathbf{H}^T\right]^{-1}. \tag{3.19}$$

Despite the fact that EnKF has a simpler implementation than the EKF algorithm, a main question is how to update the ensemble in the analysis step. We wish to determine an ensemble which approximates the analysis error covariance $\mathbf{P}^a$, given an ensemble which approximates the background error covariance $\mathbf{P}^f$. Substituting the equations (3.14) and (3.19) into the standard formula (3.10) for the analysis error covariance, we have

$$\begin{aligned}
\mathbf{P}^a &= [\mathbf{I} - \mathbf{KH}]\,\mathbf{P}^f && (3.20)\\
&= \left[\mathbf{I} - \mathbf{X}^f(\mathbf{X}^f)^T\mathbf{H}^T\left[\mathbf{R} + \mathbf{HX}^f(\mathbf{X}^f)^T\mathbf{H}^T\right]^{-1}\mathbf{H}\right]\mathbf{X}^f(\mathbf{X}^f)^T && (3.21)\\
&= \mathbf{X}^f\left\{\mathbf{I} - (\mathbf{X}^f)^T\mathbf{H}^T\left[\mathbf{R} + \mathbf{HX}^f(\mathbf{X}^f)^T\mathbf{H}^T\right]^{-1}\mathbf{HX}^f\right\}(\mathbf{X}^f)^T. && (3.22)
\end{aligned}$$

Let $\mathbf{T}$ denote the term in the curly brackets, i.e.,

$$\mathbf{T} = \mathbf{I} - (\mathbf{X}^f)^T\mathbf{H}^T\left[\mathbf{R} + \mathbf{HX}^f(\mathbf{X}^f)^T\mathbf{H}^T\right]^{-1}\mathbf{HX}^f. \qquad (3.23)$$

$\mathbf{T}$ is a Hermitian positive-definite matrix therefore, there exists a unique matrix $\mathbf{S}$, such that $\mathbf{T} = \mathbf{SS}^T$. Using the decomposition of $\mathbf{T}$, we write the analysis error covariance as

$$\mathbf{P}^a = \mathbf{X}^f\mathbf{T}(\mathbf{X}^f)^T = \mathbf{X}^f\mathbf{SS}^T(\mathbf{X}^f)^T, \qquad (3.24)$$

which leads to the conclusion that the analysis ensemble is given by

$$\mathbf{X}^a = \mathbf{X}^f\mathbf{S}, \qquad (3.25)$$

since it is the square root matrix of $\mathbf{P}^a$, as defined in (3.16). This formulation is the so called Ensemble Square Root Kalman Filter.

The approximation of the error covariance matrix can be based on an arbitrary chosen ensemble of vectors in the model space. It can be shown, however, that the optimal error for this approximation can be achieved if the centered ensemble (i.e, the ensemble from which its mean value is subtracted) spans the space of the $K-1$ maximal eigenvectors of the given background covariance $\mathbf{P}^f$. A detailed discussion on the error estimate of the analysis covariance approximation, dependent on the background covariance, is given in section 8 of [10].

3.2.1 Application to the Lorenz-96 model

We have implemented the Ensemble Square Root Kalman Filter for the Lorenz-96 model. We are going to examine some EnSRF cases by considering two of the observation networks defined in the examples in section 2.2.1. Therefore, the observation networks are: (1) observe all and (2) observe every 2 sites. In order to understand why data assimilation methods are so beneficial, we begin by presenting a no-assimilation run in Figure 3.1. For this particular example, we have used the forcing term $F = 8$ in the model and the true state. 40 ensemble members are integrated from the same initial conditions for 100 time steps. As already explained in the introduction, in section 1.2, the forcing term $F = 8$ is a value causing chaotic behavior of the model.

In Figure 3.1, both the truth state (green line) and the ensemble members (black lines) are propagated in time from the same initial conditions. After a short number of steps the evolution of the ensemble members is chaotic. This is due to the fact that the system is chaotic for $F = 8$.

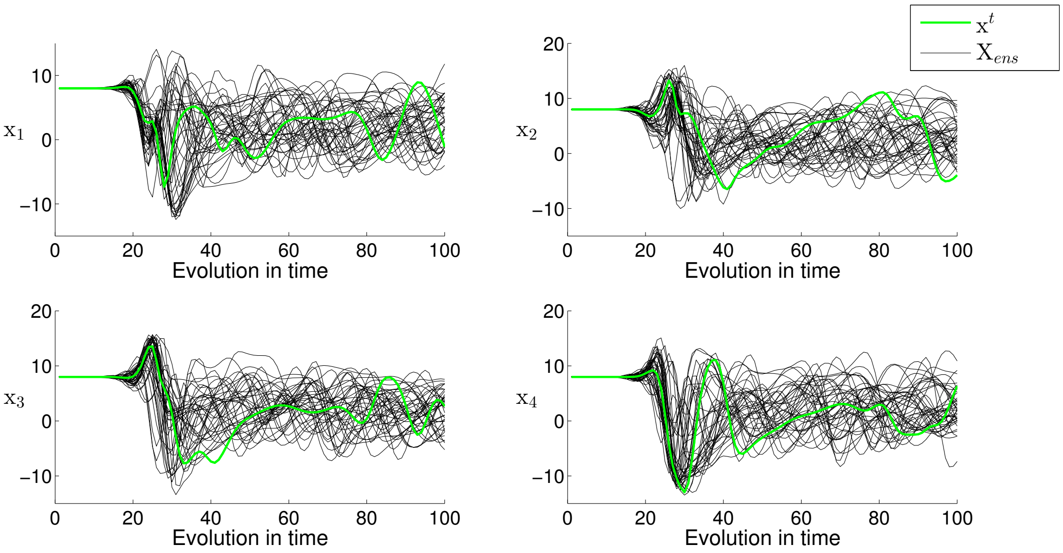


Figure 3.1: No-assimilation run (integration in time) of the Lorenz-96 model.

Now, an assimilation cycle is carried out in Figure 3.2. Each member evolves independently, until the observation becomes available. Then, the EnSRF adjusts the ensemble in order to reflect both the new state estimate and the uncertainty dictated by the observation. After the observation has been assimilated, the ensemble is integrated for 50 more time steps and we can see that the ensemble members gradually spread around the true state. Thus, the effect of the assimilation is limited to a short period after the incorporation of the observations.

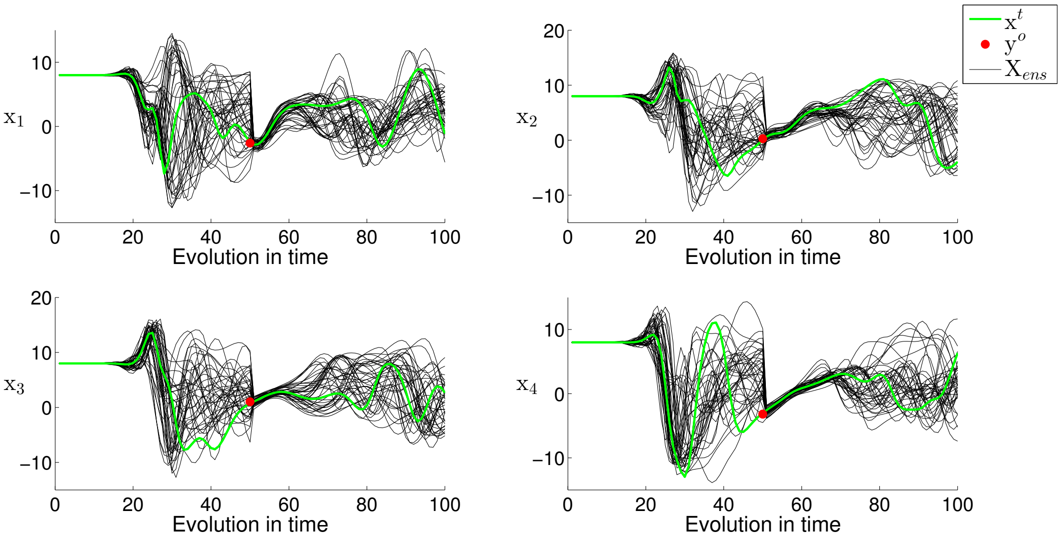


Figure 3.2: A single EnSRF analysis step. The truth state (green line) and the ensemble members (black lines) are propagated in time from the same initial condition. As soon as the observation (red dot) becomes available, the ensemble is adjusted to fit the data and give a better estimate of the true state.

In Figure 3.3, we have carried out more assimilation cycles. When observations are available more often, we observe that the EnSRF produces an analysis estimate that is closer to the truth. Therefore, we have to find an appropriate number of time steps so that the assimilation is performed as often as needed to keep the analysis close to the truth without forcing the solution.

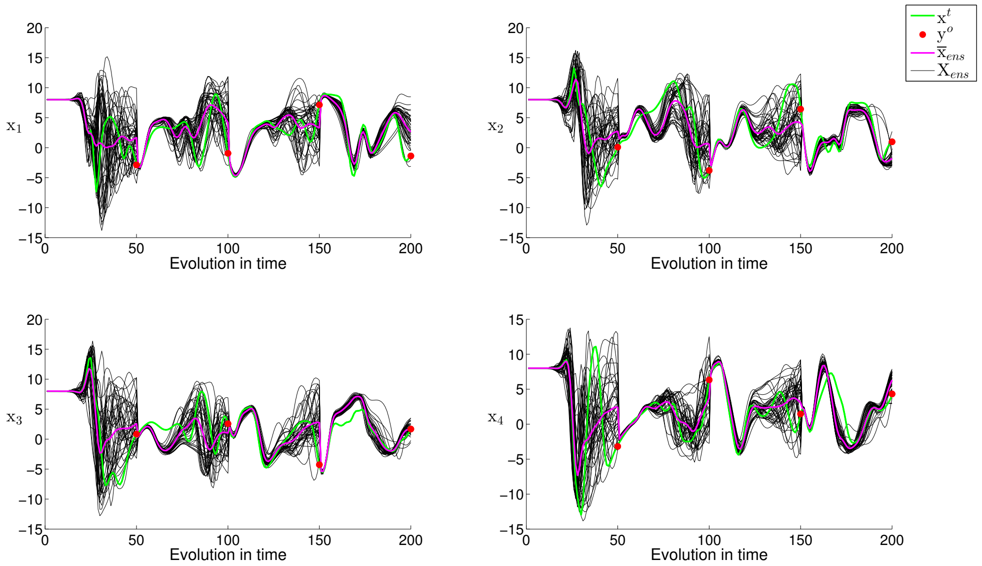


Figure 3.3: Four EnSRF cycles have been completed with 50 intermediate integration steps between the assimilations.

3.2.1.1 Case study I

In our first study case, we consider an ensemble of 10 members and observations at each site, available every 50 time-steps with observation error $\varepsilon_o \sim N(0, \sigma_o^2)$ having $\sigma_o = 0.20$. In Figure 3.4 we present the evolution in time of the first four components of the true state, the ensemble members and their mean, as well as the available observations.

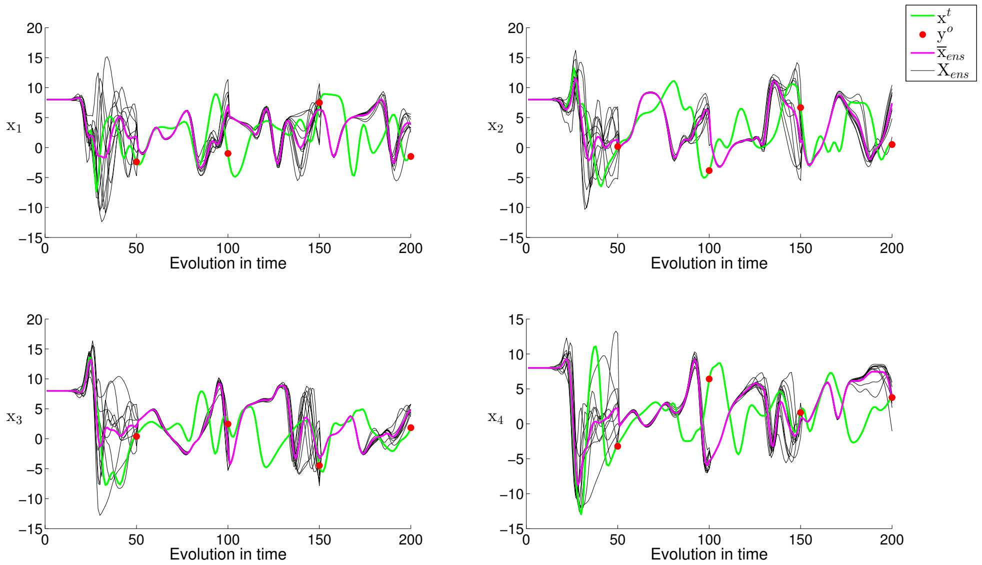


Figure 3.4: EnSRF for the Lorenz-96 with $K = 10$, observation network 1 and $\sigma_o = 0.20$.

The size of the ensemble in these assimilation methods is crucial and must adequately represent the model state. It has been shown that when the number of the ensemble members is small compared to the size of the model state, the forecast and analysis estimates are less accurate. The accuracy of the forecast estimates can be analyzed by calculating the root mean square error between the true state and the forecast or analysis ensemble mean estimates.

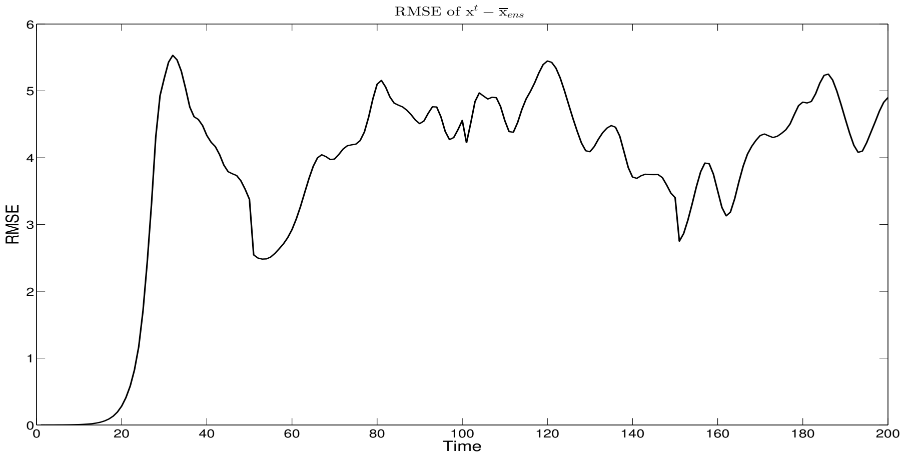


Figure 3.5: EnSRF analysis RMSE results for $K = 10$, observation network 1 and $\sigma_o = 0.20$. The mean value of the RMSE is 3.7362.

We consider now the same test case for an ensemble of 40 members to examine how the ensemble size effects the analysis estimate. In Figure 3.6 we observe that the ensemble mean is a better estimate of the true state compared to the results in Figure 3.4.

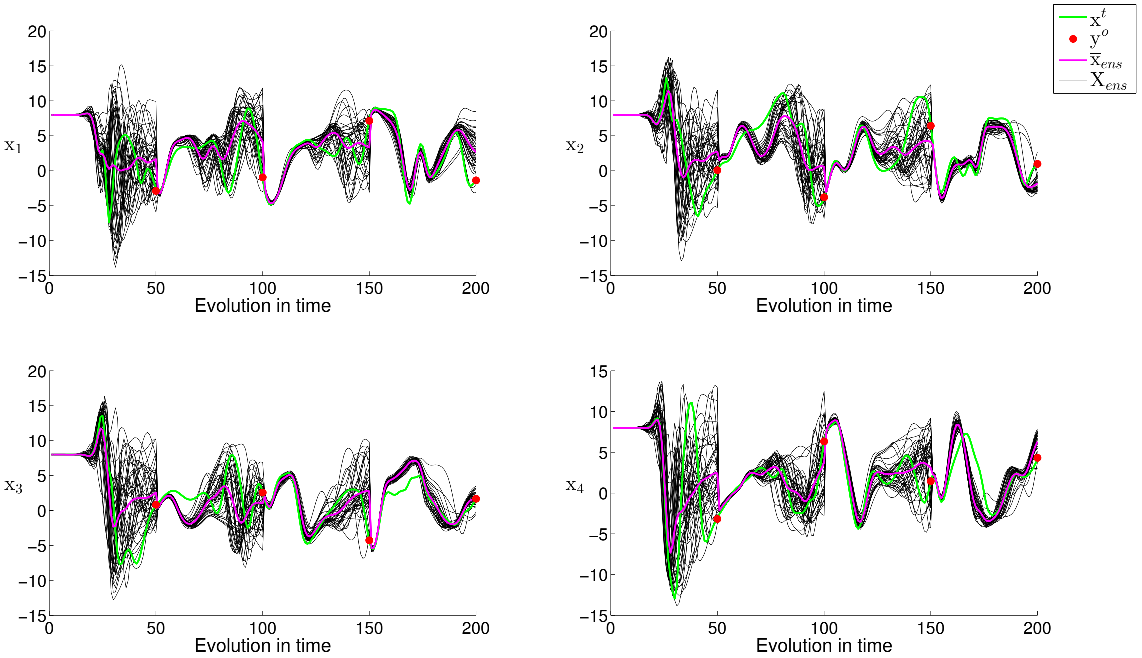


Figure 3.6: EnSRF for the Lorenz-96 with $K = 40$, observation network 1 and $\sigma_o = 0.20$.

Comparing now the Figures 3.5 and 3.7, we see that the RMSE is smaller in the latter case, as expected from the theory. Moreover, when each observation is incorporated into the estimate, we have a significant reduction of the error.

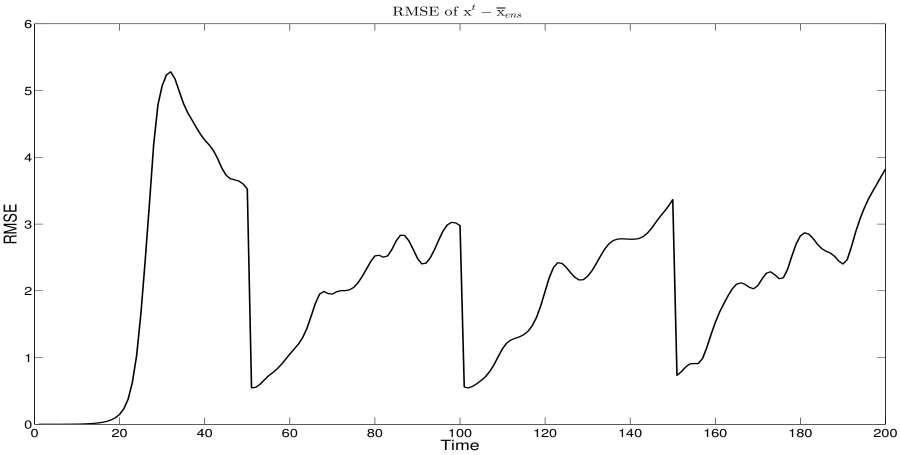


Figure 3.7: EnSRF analysis RMSE results for $K = 40$, observation network 1 and $\sigma_o = 0.20$. The mean value of the RMSE is 2.1170.

We can further improve our estimates by applying covariance inflation. This technique is used for correcting the undersampling problem (a common issue of the ensemble Kalman filtering), which appears when the size of the ensemble is so small that it cannot be statistically representative of the true model state.

The idea is to increase the forecast error covariances by inflating the deviation of the background error from the ensemble mean by a percentage. Before the assimilation of a new observation, we can multiply the analysis error covariance by a factor γ. This is known as the *multiplicative error covariance inflation factor* and is normally chosen to be slightly greater that 1.0. The optimal multiplicative inflation factor may vary according to the ensemble size and after some experiments it has been found in [39] to be 7% of the ensemble size for the EnKF algorithm and 3% for the EnSRF.

Except from the multiplicative inflation factor γ, it is also possible to use an *additive forecast error covariance inflation factor*. In forecast step, when forming the forecast error covariance matrix $\mathbf{P}^f = \mathbf{X}\mathbf{X}^T$, we can add the identity matrix multiplied by a small positive constant ℓ, i.e., $\mathbf{P}^f = \mathbf{X}\mathbf{X}^T + \ell\,\mathbf{I}$, where $\mathbf{I}$ has the same dimension as $\mathbf{P}^f$.

If we do not use some kind of inflation, the spread of the ensemble will be too small to take into account the difference between the truth and the ensemble. In our test cases we use both multiplicative and additive covariance inflation, with values $\gamma = 1.2$ and $\ell = 0.05$, and we have the following results.

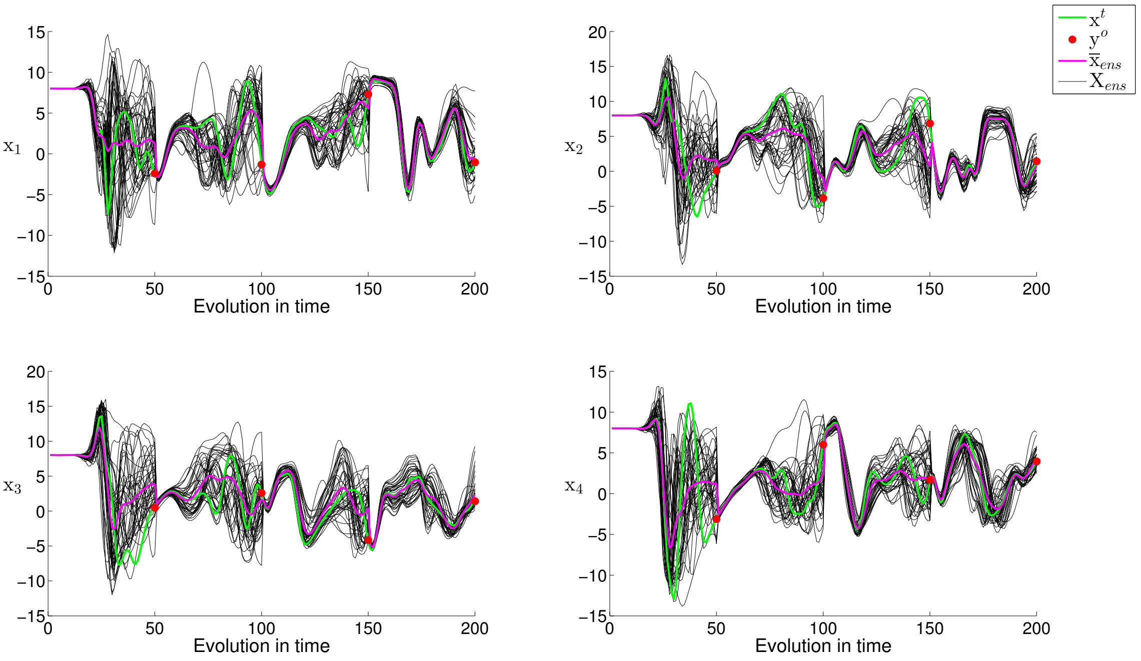


Figure 3.8: EnSRF for the Lorenz-96 with $K = 40$, observation network 1, $\sigma_o = 0.20$, inflation factors $\gamma = 1.2$ and $\ell = 0.05$.

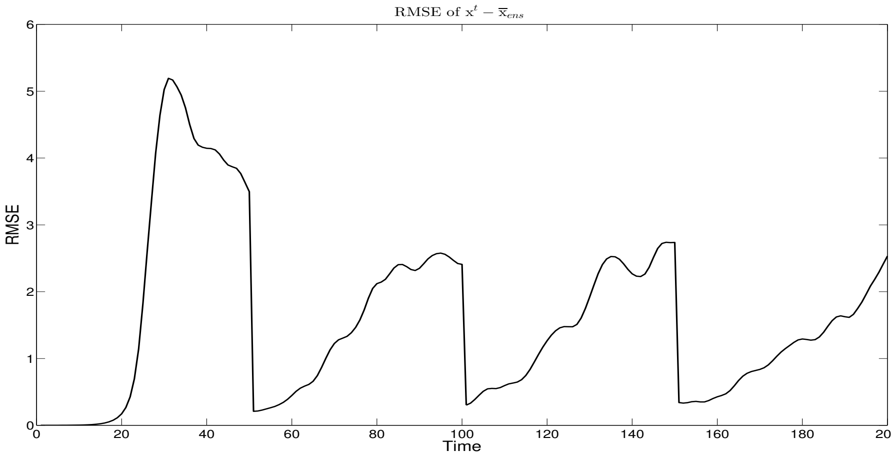


Figure 3.9: EnSRF analysis RMSE results for $K = 40$, observation network 1, $\sigma_o = 0.20$ and inflation factors $\gamma = 1.2$ and $\ell = 0.05$. The mean value of the RMSE is 1.5941.

Observing the Figures 3.6 and 3.9, we conclude that the ensemble mean is a better estimate of the true state when covariance inflation is applied, causing the RMSE to be reduced from 2.1170 to 1.5941.

3.2.1.2 Case study II

In the previous examples, we assumed that both the true and the ensemble are propagating in time through the same model dynamics, i.e., we assumed a perfect forecast model. In general this is not the case, so we are going to assume now that the forecast model is slightly different from the true. We assume that the true model has a forcing term $F = 8$, while the forecast model has $F = 8.2$. Considering an ensemble of 40 members and observations available at each location with $\sigma_o = 0.20$, we compare the results obtained with and without the use of inflation.

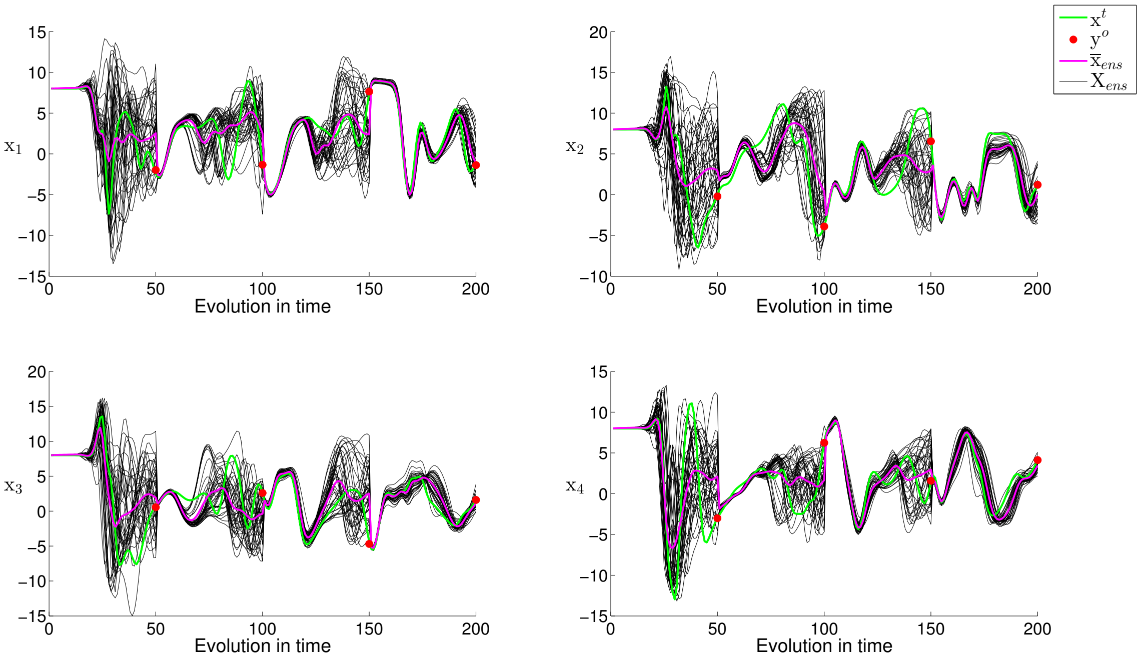


Figure 3.10: EnSRF for the Lorenz-96 with $K = 40$, observation network 1, $\sigma_o = 0.20$ and forecast model with forcing term $F = 8.2$.

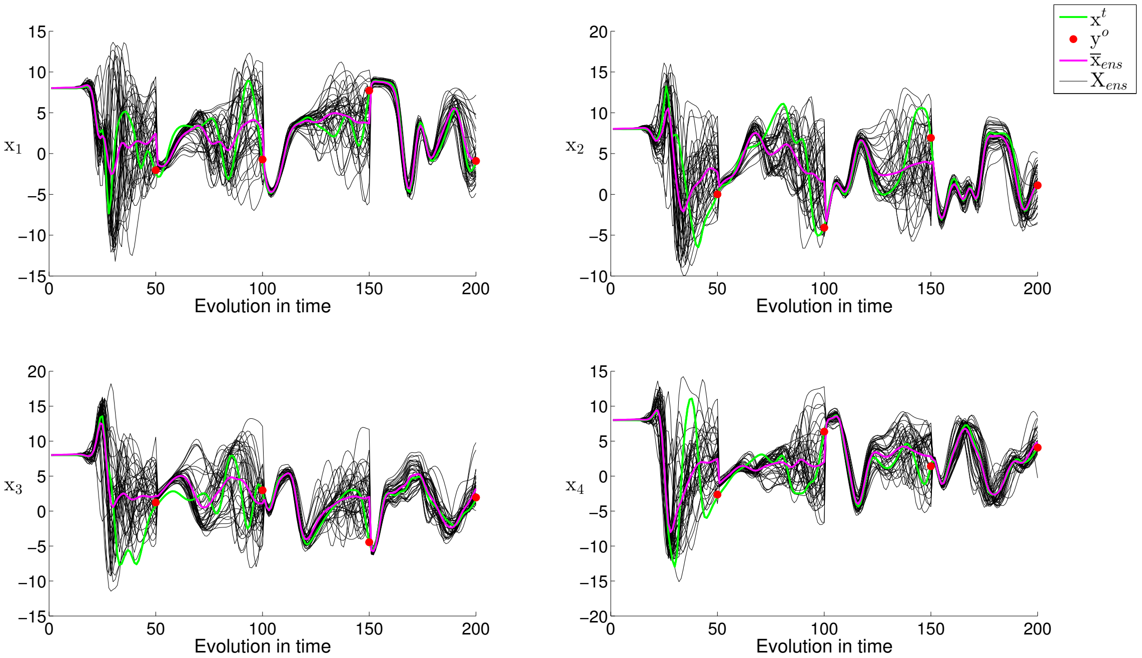


Figure 3.11: EnSRF for the Lorenz-96 with $K = 40$, observation network 1, $\sigma_o = 0.20$, assuming a forecast model with forcing term $F = 8.2$ and inflation factors $\gamma = 1.2$ and $\ell = 0.05$.

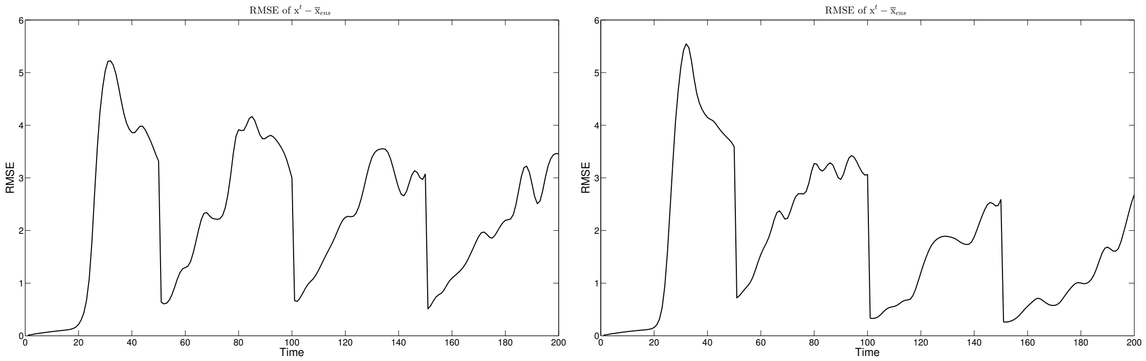


Figure 3.12: EnSRF analysis RMSE results for $K = 40$, observation network 1, $\sigma_o = 0.20$, assuming a forecast model with forcing term $F = 8.2$. In the left plot we do not use inflation, while in the right plot the inflation factors are $\gamma = 1.2$ and $\ell = 0.05$.

In the presence of covariance inflation, the RMSE appears to have smaller peaks between the assimilations, resulting an error reduction from 2.2908 (no inflation) to 1.7857 (inflation).

3.2.1.3 Case study III

We consider now an example using an ensemble of 40 members but now we assume that the observations become available at every two sites and therefore, we expect that the analysis estimate will be less accurate on the unobserved variables.

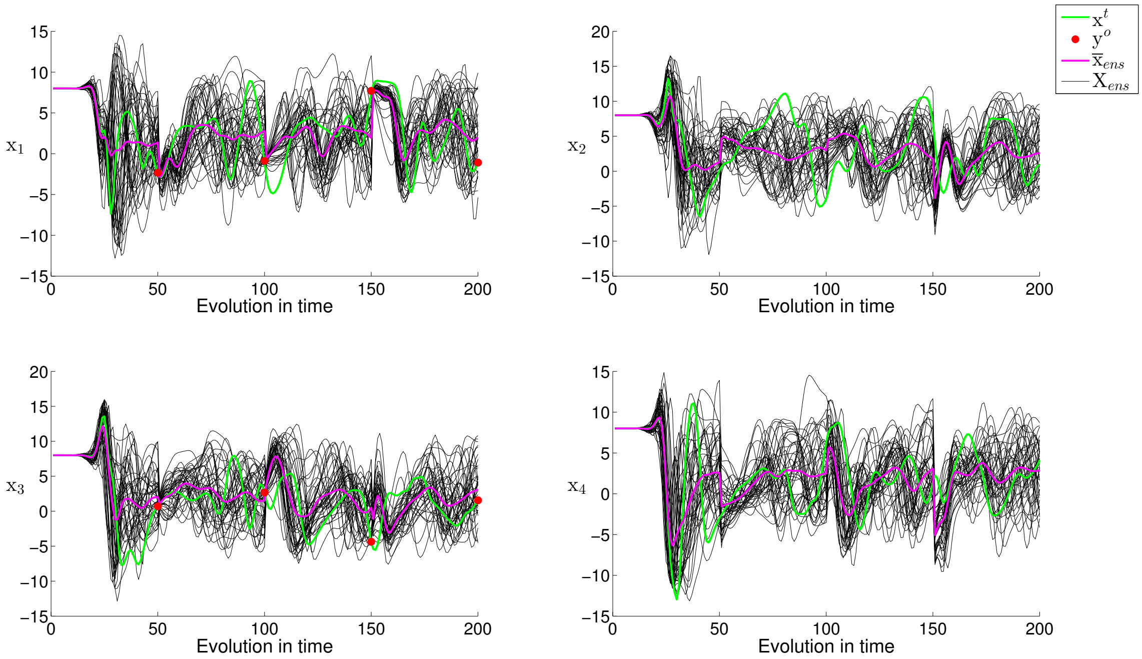


Figure 3.13: EnSRF for the Lorenz-96 with $K = 40$, observation network 2 and $\sigma_o = 0.20$.

In Figure 3.13 we observe that the ensemble members do not follow the true state after the assimilation of each observation. This is due to the fact that the observations are available only on half of the sites. For the unobserved sites, their evolution in time depends only on the model dynamics and thus, their behavior is almost chaotic.

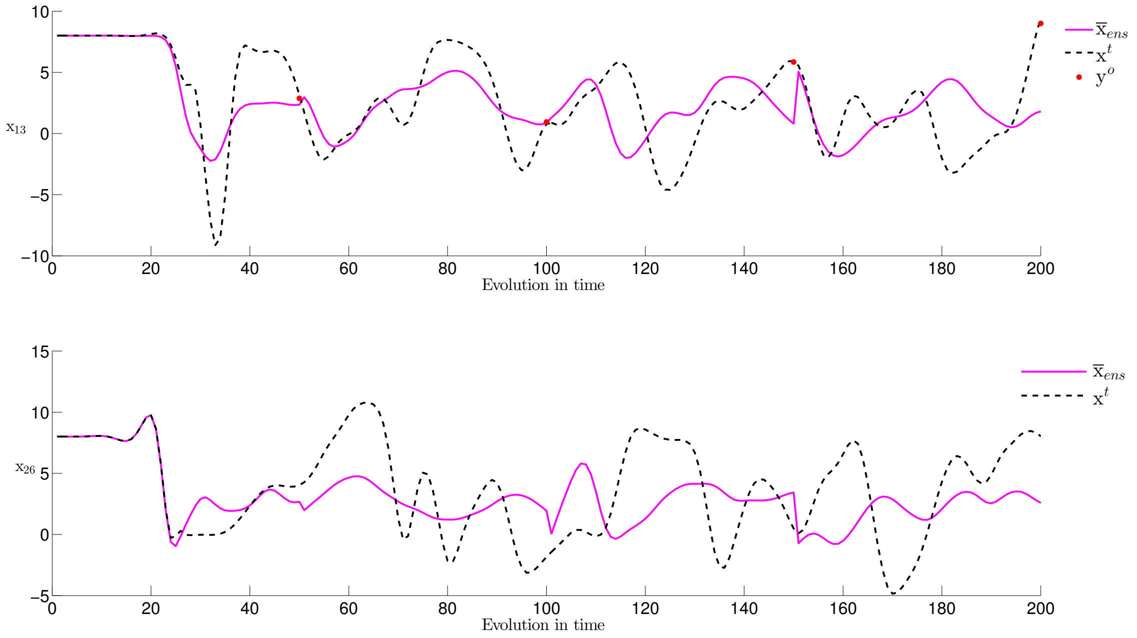


Figure 3.14: EnSRF for the Lorenz-96 with $K = 40$, observation network 2 and $\sigma_o = 0.20$. Evolution in time of the 13th and 26th components, which are observed and unobserved respectively.

The evolution in time of an observed and an unobserved component of the model state is shown in Figure 3.14. As can be seen, the ensemble mean is not a good estimate of the true state, which is an almost expected behavior of the analysis, since we have less information about the behavior of the actual model state. The RMSE of the estimate is plotted against time in the figure below. The error's behavior is quite different from the previous test case, in which, after the assimilation of the observations there was a significant error reduction.

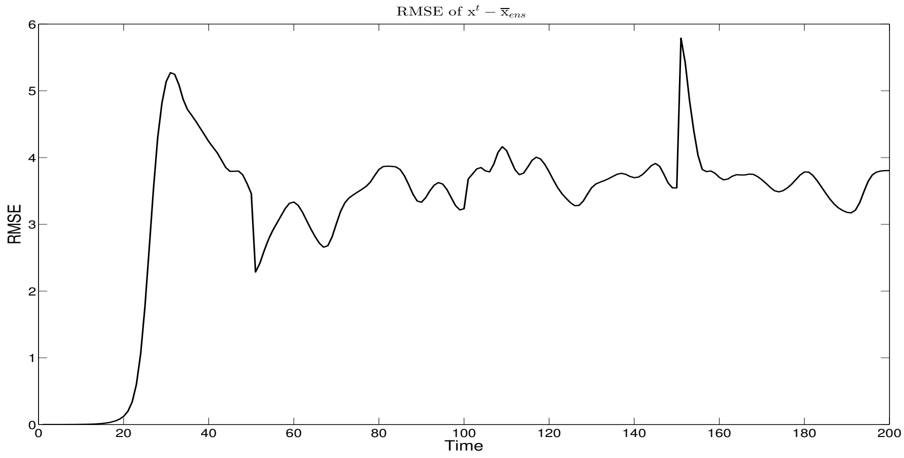


Figure 3.15: EnSRF analysis RMSE results for $K = 40$, observation network 2 and $\sigma_o = 0.20$. The mean value of the RMSE is 3.2396.

So far we have presented different test cases of 3D-Var and EnSRF for the Lorenz-96 model. We conclude that in 3D-Var the constant background error covariance must be tuned properly to receive improved estimates. In the EnSRF, it is important to choose an appropriate ensemble so

that the ensemble covariance will be a good approximation of the error covariance matrix. An important restriction of the Ensemble Kalman filtering is the size of the ensemble, due to computational requirements of maintaining a large ensemble. The success of the ensemble methods depends on how well the ensemble represents the model state. If the ensemble is so small that it cannot statistically represent the system, then we have undersampling and the filter produces poor estimates. A solution to the underestimation problem is the covariance inflation method and the choice of the inflation factor depends on the type of the ensemble filter that we use.
In the appendix there can be found results of additional test-cases concerning these assimilation methods for the Lorenz-96 model.

Chapter 4

Weather Research and Forecasting Model

The *Weather Research and Forecasting Model* (WRF-Model) is a state-of-the-art atmospheric modeling system which serves for both atmospheric research and operational forecasting needs. The development of the WRF began in the latter part of the 1990's as a collaboration among the National Center for Atmospheric Research (NCAR)[1], the National Oceanic and Atmospheric Administration (NOAA), the Air Force Weather Agency (AFWA), the Naval Research Laboratory, the University of Oklahoma and the Federal Aviation Administration.
It offers many options for atmospheric physical processes and is suitable for a broad range of applications, including the following: real-time NWP, idealized atmospheric simulations, data assimilation research, hurricane research, coupled-model applications, regional climate research, forecast search, modeling training.
There are two dynamical solvers available for the resolution of the atmospheric governing equations, and the variants of the model are known as WRF-ARW (Advanced Research WRF) and WRF-NMM (Nonhydrostatic Mesoscale Model). The WRF-ARW is provided by the NCAR Mesoscale and Microscale Meteorology Division (MMM Division). The WRF-NMM is provided by the Developmental Testbed Center (DTC) developed at the National Centers for Environmental Prediction (NCEP).
The main components of the WRF modeling system are the WRF Preprocessing System (WPS), the WRF Data Assimilation system (WRFDA), the WRF core (solver) and the Post-processing and Visualization tools. For a detailed description of each component refer to the official WRF technical report [32].

4.1 WRF Data Assimilation

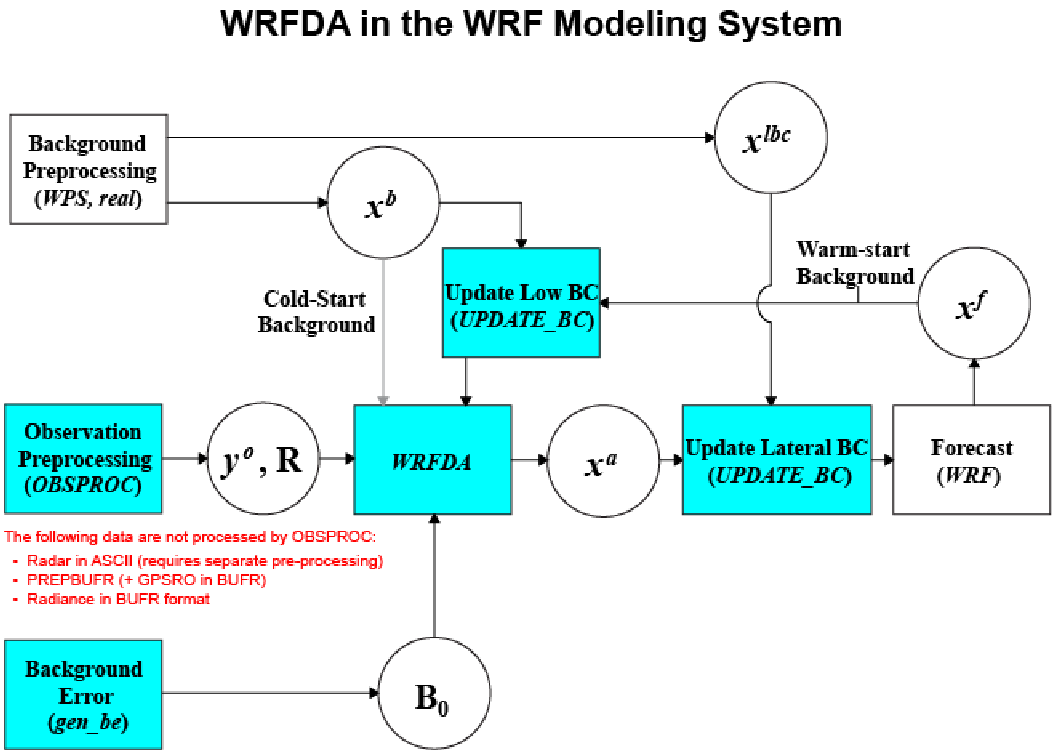
The main objective of the WRF partners in 1999-2001 was to develop a unified community data assimilation system characterized by: robustness, accuracy, computational efficiency, flexibility, equipped with an adequate support and documentation, while at the same time it had to be easy to use. As a starting point, the initial WRF data assimilation system was based on the the community fifth-generation Pennsylvania State University–National Center for Atmospheric Research Mesoscale Model (MM5) 3DVAR system (for a detailed description refer to [3]). Therefore, the first version of the WRF data assimilation component included an incremental 3D-Var algorithm and for that reason, it was named as WRF3DVar. Later, a 4D-Var assimi-

[1] The NCAR was represented by the National Centers for Environmental Prediction (NCEP) and the (then) Forecast Systems Laboratory (FSL).

lation scheme was included and the name was changed to WRF-VAR. Similarly, in 2008 after the release of a hybrid variational/ensemble method the component was renamed to WRFDA, which prevails even today.

WRFDA can be used to ingest observations into the interpolated analyses created by the Preprocessing system but also, it can used to update WRF model's initial conditions when the WRF model is run in cycling mode. It is based on an incremental variational data assimilation technique and supports both 3D-Var and 4D-Var methods. It also has the capability of hybrid data assimilation that combines the benefits of the variational approach with the statistical, flow-dependent error information provided by the ensemble forecasts.

The various components of the WRFDA system are shown in blue in the sketch below, together with their relationship with the rest of the WRF Modeling system. The circles that are directly linked to the WRFDA box represent the datasets involved in the WRFDA process, while the remaining boxes correspond to the algorithms of the ARW system.


WRFDA in the WRF Modeling System
Background Preprocessing (WPS, real)
x^{lbc}
x^b
Warm-start Background
Cold-Start Background
Update Low BC (UPDATE_BC)
x^f
Observation Preprocessing (OBSPROC)
y^o, R
WRFDA
x^a
Update Lateral BC (UPDATE_BC)
Forecast (WRF)
Background Error (gen_be)
B_0

The following data are not processed by OBSPROC:

- Radar in ASCII (requires separate pre-processing)
- PREPBUFR (+ GPSRO in BUFR)
- Radiance in BUFR format

Figure 4.1: Sketch showing the relationship between the components of WRFDA and the rest of the WRF system. Source: This diagram has been taken from the original WRFDA documentation [32].

The variables shown in Figure 4.1 represent:

$\mathbf{x}^b$ the first guess, either from a previous WRF forecast or from WPS/real.exe output.
$\mathbf{x}^{lbc}$ the lateral boundary from WPS/real.exe output.
$\mathbf{x}^a$ the analysis provided by the WRFDA data assimilation system.
$\mathbf{x}^f$ the WRF forecast output.
$\mathbf{y}^o$ the observations processed by the OBSPROC utility.
$\mathbf{B}_0$ the background error statistics from generic BE data (CV3) or gen_be.
$\mathbf{R}$ the observational and representative error statistics.

As can be seen in Figure 4.1, the WRFDA needs three inputs:

A first guess $\mathbf{x}^b$. In *cold-start* mode, this is typically a forecast/analysis from another model provided by the Preprocessing system. Its functions are responsible for defining the simulation domain and interpolating forecast fields from another models to the target domain. In *cycling* or *warm-start* mode, the background processing is not required as the background field $\mathbf{x}^b$ is a short-range ARW forecast provided from the previous step.

Observations $\mathbf{y}^o$. Observation data may be supplied using the Observation Pre-processing "*obsproc*" utility. The purpose of this utility is to remove observations outside the specified time and spacial domains, re-order and merge duplicate data reports, retrieve pressure or height based on observed information using the hydrostatic assumption, perform basic quality control, assign the total observation errors and covariances $\mathbf{R}$ and finally, reformat the observations to the 3D-Var format.

A Background error covariance $\mathbf{B}$. For the appropriate definition of the background error covariance, the code is supplied with the utility "*gen_be*". The specification of the background error covariance is crucial for the final analysis obtained by the assimilation, therefore we discuss the main aspects of this utility in section 4.2.

In Figure 4.1, there are two parts that have not been discussed so far: the Lateral and Low Boundary Conditions update processes, which are both included in the UPDATE_BC utility supplied with WRFDA. When using the WRFDA output (i.e., the analysis $\mathbf{x}^a$) to run a WRF forecast, it is essential to update the WRF lateral boundary conditions $\mathbf{x}^{lbc}$ so that they match to the new WRFDA initial condition (analysis). The lateral BCs update procedure, as seen in the WRFDA Flow Figure 4.1, requires the WRFDA analysis state $\mathbf{x}^a$ and the lateral boundaries $\mathbf{x}^{lbc}$ provided by the WPS/real preprocessing (in the file *wrfbdy_01*). However, in a global run, the lateral boundary conditions are no longer needed since the simulation domain covers the entire globe. In cycling mode (warm-start), the lower boundary conditions in the first guess file also need to be updated based on the information from the WRFINPUT file which is generated by the WPS/real at analysis time.

The WRF 3DVar system

The WRF 3DVar system aims to produce an optimal estimate of the true atmospheric state at analysis time through the minimization of a prescribed cost function:

$$J(\mathbf{x}) = J_b(\mathbf{x}) + J_o(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b) + \frac{1}{2}(\mathbf{y}-\mathbf{y}^o)^T(\mathbf{E}+\mathbf{F})^{-1}(\mathbf{y}-\mathbf{y}^o). \qquad (4.1)$$

The solution $\mathbf{x}$, obtained by the minimization of (4.1), represents the a posteriori maximum likelihood (minimum variance) estimate of the true state of the atmosphere given two sources of a priori data: the first guess $\mathbf{x}^b$ and the observations $\mathbf{y}^o$. The fit to individual data points is weighted by the estimates of their errors: $\mathbf{B}, \mathbf{E}$ and $\mathbf{F}$ are the background, observation and representativity error covariance matrices, respectively. Recalling the 3D-Var functional (2.51), we observe a different formulation of $J_o(\mathbf{x})$. The representativity error is an estimate of inaccuracies introduced in the observation operator H that is used to transform the gridpoint analysis $\mathbf{x}$ to the observation space $\mathbf{y} = H(\mathbf{x})$ for comparison against the observations. Therefore, defining a total observational operator $\mathbf{R} = \mathbf{E} + \mathbf{F}$ the two formulations are, in fact, identical.

The 3D-Var algorithm adopted in WRFDA is a model-space, incremental formulation of the variational problem and the original description can be found in [4]. Assuming a model state $\mathbf{x}$ with n degrees of freedom, the calculation of the full background term $J_b(\mathbf{x})$ requires $\sim \mathcal{O}(n^2)$ calculations. For a typical NWP model with $n^2 \sim 10^{12}$ direct solution is not feasible in the time frame needed for data assimilation in operational applications. A practical solution in this problem is to calculate the $J_b(\mathbf{x})$ in terms of control variables defined as $\mathbf{x}' = \mathbf{U}\mathbf{v}$ where $\mathbf{x}' = \mathbf{x} - \mathbf{x}^b$ is the analysis increment. The transformation $\mathbf{U}$ permits the use of efficient filtering techniques that approximate the full background error covariance matrix. Moreover, if $\mathbf{U}$ is well designed, the product $\mathbf{U}\mathbf{U}^T$ will closely match the full background error covariance matrix $\mathbf{B}$. The 3D-Var functional in terms of analysis increments can be rewritten as

$$J(\mathbf{v}) = J_b + J_o = \frac{1}{2}\mathbf{v}^T\mathbf{v} + \frac{1}{2}(\mathbf{y}^{o\prime} - \mathbf{HUv})^T(\mathbf{E}+\mathbf{F})^{-1}(\mathbf{y}^{o\prime} - \mathbf{HUv}), \tag{4.2}$$

where $\mathbf{y}^{o\prime} = \mathbf{y}^o - H(\mathbf{x}^b)$ is the innovation vector and $\mathbf{H}$ is the linearization of the observation operator H. The minimization of this incremental functional is carried out using a conjugate gradient method yielding the analysis increments $\mathbf{x}'$, which are added to the first guess $\mathbf{x}^b$ to provide an updated analysis.

4.2 Background Error Covariance Estimates

The background error covariances (BE) are an important input to variational data assimilation systems. They influence the analysis fit to the observations and also they define the analysis response away from the observations, e.g., in cases of data-sparse regions.
Unlike the ensemble techniques, the variational DA methods do not explicitly evolve the background error covariances in real-time. Instead, climatological statistics are usually approximated by either using ensemble perturbations or using the "National Meteorological Center (NMC) method" (introduced in [9] and [30], respectively). The NMC method has been adopted by the most NWP centers for estimating the forecast error covariance using the difference $\mathbf{x}'$ between forecasts valid at the same time.

The WRFDA system includes the "*gen_be*"utility that has been designed by NCAR/MMM for the calculation of domain-specific climatological estimates of forecast error covariances, instead of using the default statistics[2] supplied within the release. The background error covariance is defined as

$$\mathbf{B} = E\left\{\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T\right\} \approx E\left\{\mathbf{x}'\mathbf{x}'^T\right\} \tag{4.3}$$

where the expected value indicates an average over time or geographical area. The true background in unknown, therefore it is assumed that it is well represented by the model state perturbation $\mathbf{x}'$. In the NMC method, $\mathbf{x}'$ is the difference between short-range forecasts, e.g., $\mathbf{x}' = \mathbf{x}^f(24h) - \mathbf{x}^f(12h)$ for typical regional applications or $\mathbf{x}' = \mathbf{x}^f(48h) - \mathbf{x}^f(24h)$ for global applications, valid at the same time. Alternatively, for an ensemble-based approach, $\mathbf{x}'_k = \mathbf{x}_k - \overline{\mathbf{x}}$, where the overbar denotes the average over the $k = 1, \ldots, K$ ensemble members. Both methods can be used by *gen_be* utility for the approximation of the background error covariance.

Regarding the 3D-Var formulation within the WRFDA, the background error covariances are specified not in the model space $\mathbf{x}'$ but in a control variable space $\mathbf{v}$, which is related to the

[2] The WRFDA system provides default synoptic-scale climatological forecast error statistics for initial setup, testing and training simulations.

model space through the control variable transform $\mathbf{U}$, i.e.,

$$\mathbf{x}' = \mathbf{U}\mathbf{v} = \mathbf{U}_p\mathbf{U}_v\mathbf{U}_h\mathbf{v}.$$

The expansion $\mathbf{U} = \mathbf{U}_p\mathbf{U}_v\mathbf{U}_h$ represents the various stages of the covariance modeling: horizontal correlations $\mathbf{U}_h$, vertical covariances $\mathbf{U}_v$ and multivariate covariances $\mathbf{U}_p$. The components of $\mathbf{v}$ are chosen so that their error cross-correlations are negligible, permitting the matrix $\mathbf{B}$ to be block-diagonalized.

The *gen_be* code is separated into a number of stages:

Stage 0	Convert model-specific data to standard fields.
Stage 1	Remove time-domain mean from the fields.
Stage 2	Calculate regression coefficients and use them to define unbalanced control variables.
Stage 3	Calculate vertical error covariances for control variables (represented via decomposition into eigenvectors/eigenvalues).
Stage 4	Calculate horizontal error correlations: lengthscales (in regional domains) and power spectra (in global domains).

A detailed description of each stage of the control variable transform can be found in [3] and [4]. The users have two main choices to define the background error covariance when running the *gen_be* utility, called CV3 and CV5. In CV3, the control variables are in physical space while in CV5 they are in eigenvector space. Moreover, in CV3 a BE file (*be.dat*) is supplied with the WRFDA source code, which is a global error covariance that can be used for any regional domain. In CV5, on the other hand, BE is a domain-dependent error covariance, which is generated through the *gen_be* utility (using an empirical orthogonal function (EOF) to represent the vertical covariance), based on forecast or ensemble data from the simulation domain.

4.3 4DVar on WRFDA

The 4D-Var assimilation method as already mentioned, has a number of advantages over the 3D-Var scheme. It allows the observations to be assimilated at the time of their measurement or in a specific time-window. Moreover, it implicitly defines flow-dependent forecast error covariances and almost has the ability to use a forecast model as a constraint leading to an improved analysis estimate. Given these advantages, 4D-Var was included in the capabilities of the WRFDA system. The WRF 4DVar algorithm considers the incremental 4D-Var formulation (refer to [6], [15], [22] and [34]), which is commonly used in operational systems. The incremental 4D-Var approach is designed to find the analysis increment that minimizes a prescribed cost function defined as a function of the analysis increment instead of the analysis itself.

The incremental 4D-Var procedure can be seen as a pair of nested iterations: the *outer* and *inner* loops. In the outer loop, the high-resolution model is run and innovations are computed with respect to the resulting high-resolution trajectory. Then, the inner loop uses a degraded lower-resolution model with simplified physics, to perform most of the integrations required by the minimization of the approximate functional. The resulting lower-resolution innovation is then used to correct the initial conditions of the high-resolution model, which is integrated once more, and the innovations are recomputed with respect to this updated trajectory. In this formulation, the tangent linear and the adjoint models are used in the inner-loop minimization, while the evolution of the background error is estimated with the full forward model. Designing the incremental 4D-Var, requires to decide how many outer loops will be performed and what will be the configuration of the lower-resolution model used in the inner loop.

The WRF 4DVar functional is of the form

$$J = J_b + J_o + J_c, \tag{4.4}$$

which includes quadratic measures of the distance to the background, observation and balanced control. The background cost function term J_b is

$$\begin{aligned}
J_b &= \frac{1}{2}(\mathbf{x}^n - \mathbf{x}^b)^T \mathbf{B}^{-1} (\mathbf{x}^n - \mathbf{x}^b) \\
&= \frac{1}{2}\left[(\mathbf{x}^n - \mathbf{x}^{n-1}) + (\mathbf{x}^{n-1} - \mathbf{x}^b)\right]^T \mathbf{B}^{-1} \left[(\mathbf{x}^n - \mathbf{x}^{n-1}) + (\mathbf{x}^{n-1} - \mathbf{x}^b)\right] \\
&= \frac{1}{2}\left[(\mathbf{x}^n - \mathbf{x}^{n-1}) + \sum_{i-1}^{n-1}(\mathbf{x}^i - \mathbf{x}^{i-1})\right]^T \mathbf{B}^{-1} \left[(\mathbf{x}^n - \mathbf{x}^{n-1}) + \sum_{i-1}^{n-1}(\mathbf{x}^i - \mathbf{x}^{i-1})\right].
\end{aligned}$$

The background error covariance matrix $\mathbf{B}$ is typically a climatological estimate. However, it may also be derived from prior or ensemble based flow-dependent estimates (using the gen_be utility) as seen in the previous section. The background $\mathbf{x}^b$ is usually a short-range forecast created by a previous analysis. The vector $\mathbf{x}^i$ denotes an intermittent analysis after the i-th outer loop with $i = 1, \dots, n$ be the index of the iteration. The final analysis of WRF 4DVar is obtained after the last outer loop denoted as $\mathbf{x}^n$.
The inner loop minimization starts from a guess vector, $\mathbf{x}^{n-1}$, which is the analysis vector from the latest outer loop. In the first outer loop, the background $\mathbf{x}^b$ is normally taken as the first guess vector $\mathbf{x}^0$. Nevertheless, the background and guess vectors should not be mixed in the incremental formulation, as they are the same only during the first outer loop.

The observation cost function J_o in (4.4), represents the quadratic measure of the distance between the analysis $\mathbf{x}^n$ and the observations $\mathbf{y}_k$:

$$\begin{aligned}
J_o &= \frac{1}{2}\sum_{k=1}^{K}\left\{H_k\left[M_k(\mathbf{x}^n)\right] - \mathbf{y}_k\right\}^T \mathbf{R}^{-1} \left\{H_k\left[M_k(\mathbf{x}^n)\right] - \mathbf{y}_k\right\} \\
&= \frac{1}{2}\sum_{k=1}^{K}\left\{H_k\left[M_k(\mathbf{x}^n - \mathbf{x}^{n-1} + \mathbf{x}^{n-1})\right] - \mathbf{y}_k\right\}^T \mathbf{R}^{-1} \left\{H_k\left[M_k(\mathbf{x}^n - \mathbf{x}^{n-1} + \mathbf{x}^{n-1})\right] - \mathbf{y}_k\right\} \\
&\approx \frac{1}{2}\sum_{k=1}^{K}\left\{H_k\left[M_k(\mathbf{x}^{n-1})\right] + \mathbf{H}_k\mathbf{M}_k(\mathbf{x}^n - \mathbf{x}^{n-1}) - \mathbf{y}_k\right\}^T \times \\
&\quad \mathbf{R}^{-1}\left\{H_k\left[M_k(\mathbf{x}^{n-1})\right] + \mathbf{H}_k\mathbf{M}_k(\mathbf{x}^n - \mathbf{x}^{n-1}) - \mathbf{y}_k\right\} \\
&= \frac{1}{2}\sum_{k=1}^{K}\left[\mathbf{H}_k\mathbf{M}_k(\mathbf{x}^n - \mathbf{x}^{n-1}) - \mathbf{d}_k\right]^T \mathbf{R}^{-1} \left[\mathbf{H}_k\mathbf{M}_k(\mathbf{x}^n - \mathbf{x}^{n-1}) - \mathbf{d}_k\right].
\end{aligned}$$

A linear approximation of the operators has been made here. We have the observation operator H_k and its tangent linear model $\mathbf{H}_k$, as well as the model operator M_k and its tangent linear approximation $\mathbf{M}_k$, where k denotes the observation window. Moreover, $\mathbf{d}_k = \mathbf{y}_k - H_k M_k(\mathbf{x}^{n-1})$ is the innovation vector for k-th observation window and $\mathbf{R}$ is the observation error covariance matrix.

Finally, the balance cost function in (4.4) measures the distance between the analysis and a balanced state. The use of unbalanced initial conditions often generates high-frequency oscillations with amplitudes larger than those observed in nature. To overcome this problem, a digital filter is included in WRF 4DVar to remove these high-frequency waves in the analysis.

According to [12], the basic idea of using a digital filter for initialization is to calculate the Fourier transform of the initial noisy forecast values in time domain and then, set the coefficients of high-frequencies to zero. Finally, the filtered values can be obtained by taking the inverse Fourier transform and use them as initial conditions for a new forecast starting at time 0. In WRF 4DVar the digital filter in J_c has the form:

$$\begin{aligned} J_c = \frac{1}{2}\gamma_{df} & \left[\mathbf{M}_{N/2}(\mathbf{x}^n - \mathbf{x}^{n-1}) - \sum_{i=0}^{N} f_i \mathbf{M}_i (\mathbf{x}^n - \mathbf{x}^{n-1}) \right]^T \times \\ \mathbf{C}^{-1} & \left[\mathbf{M}_{N/2}(\mathbf{x}^n - \mathbf{x}^{n-1}) - \sum_{i=0}^{N} f_i \mathbf{M}_i (\mathbf{x}^n - \mathbf{x}^{n-1}) \right] \\ = \frac{1}{2}\gamma_{df} & \left[\sum_{i=0}^{N} g_i \mathbf{M}_i (\mathbf{x}^n - \mathbf{x}^{n-1}) \right]^T \mathbf{C}^{-1} \left[\sum_{i=0}^{N} g_i \mathbf{M}_i (\mathbf{x}^n - \mathbf{x}^{n-1}) \right]. \end{aligned}$$

Here γ_{df} is the weight assigned to the J_c term, f_i is the coefficient for the digital filter and g_i is the modified coefficient with

$$g_i = \begin{cases} -f_i, & i \neq N/2, \\ 1 - f_i, & i = N/2, \end{cases}$$

where N is the total integration steps over the assimilation window (for more information about the use of digital filters refer to [11], [12], [27] and [38]).
Furthermore, $\mathbf{C}$ is a diagonal matrix containing variances of wind, temperature and dry surface pressure having default values $(3\text{m/s})^2, (1\text{K})^2$ and $(10\text{hPa})^2$, respectively.

As defined in (4.4), the full 4D-Var functional is composed as the sum of J_b, J_o and J_c. In order to avoid the calculation of the full background term J_b and accelerate the minimization algorithm, we define a control variable transform (similarly to the incremental 3D-Var case)

$$\mathbf{v}^n = \mathbf{U}^{-1}(\mathbf{x}^n - \mathbf{x}^{n-1}),$$

where $\mathbf{U}$ is defined as $\mathbf{B} = \mathbf{U}\mathbf{U}^T$. Applying this transformation to the 4D-Var cost function we finally obtain its gradient $J'(\mathbf{v}^n) = \nabla J(\mathbf{v}^n)$ with respect to the control variable $\mathbf{v}^n$, which is

$$\begin{aligned} J'(\mathbf{v}^n) = & \sum_{i=1}^{n-1} \mathbf{v}^i + \mathbf{v}^n \\ & + \mathbf{U}^T \sum_{k=1}^{K} \mathbf{M}_k^T \mathbf{H}_k^T \mathbf{R}^{-1} \{ \mathbf{H}_k \mathbf{M}_k \mathbf{U} \mathbf{v}^n - \mathbf{d}_k \} \\ & + \mathbf{U}^T \sum_{i=0}^{N} \mathbf{M}_i^T g_i \gamma_{df} \mathbf{C}^{-1} \left(\sum_{i=0}^{N} g_i \mathbf{M}_i \mathbf{U} \mathbf{v}^n \right). \end{aligned} \tag{4.5}$$

In the last expression, $\mathbf{H}_k^T$ is the adjoint observation operator over the assimilation window k, $\mathbf{M}_k^T$ is the adjoint model, which propagates the analysis residuals $\{\mathbf{H}_k \mathbf{M}_k \mathbf{U} \mathbf{v}^n - \mathbf{d}_k\}$ and the digital filter forcing $\sum_{i=0}^{N} f_i \mathbf{M}_i \mathbf{U} \mathbf{v}^n$, backward in time. A 3D-Var solution can be obtained by setting $K = 1$ and removing the model-related components. According to the theory, the analysis state is obtained when the cost function (4.4) is minimized, or equivalently, when its gradient in (4.5) is equal to zero. As described earlier, the cost function minimization is carried out using the Conjugate Gradient method; the minimization takes place in the inner-loop of the WRF 4DVar algorithm.

Given the background model state $\mathbf{x}^b$, the lateral boundary conditions WRFBDY valid during the analysis time window, the background and the observation error covariance matrices $\mathbf{B}$ and $\mathbf{R}$, respectively, as well as the observations grouped into K time windows, WRF 4DVar will produce the final analysis $\mathbf{x}^n$. The major software components of the WRF 4DVar system are the following:

WRF The WRF ARW model is referred to as WRF_NL (WRF full nonlinear model). The ARW solves the compressible, non-hydrostatic Euler equations, cast in flux form and conserving of both mass and scalar. The ARW model has a terrain vertical coordinate and an Arakawa C-grid staggering in the horizontal. The model uses the Runge-Kutta 2nd and 3rd order time integration schemes, and second to sixth order advection options.

WRFPLUS This component comprises the WRF tangent linear model (WRF_TL) and its adjoint model (WRF_AD). Both models are calculated based on a simplified version of the full WRF model (WRF_SN), which includes some simple physical processes such as vertical diffusion and large-scale condensation.

VAR The VAR module contains all the components of WRF 3DVar extended to include four-dimensional enhancements such as grouping of the observations $\mathbf{y}$ into $\mathbf{y}_k$ and replacing $H, \mathbf{H}, \mathbf{H}^T$ by $H_k, \mathbf{H}_k$ and $\mathbf{H}_k^T$, respectively, in an observation window k. This function calls the WRF_NL, WRF_TL and WRF_AD models and the grid/variable transform operators.

COM The aforementioned components are separate coded, therefore COM manages the communication among them. Its implementation is hidden from the other three components and allows the transmission of data either through disk I/O, or through memory for maximum efficiency.

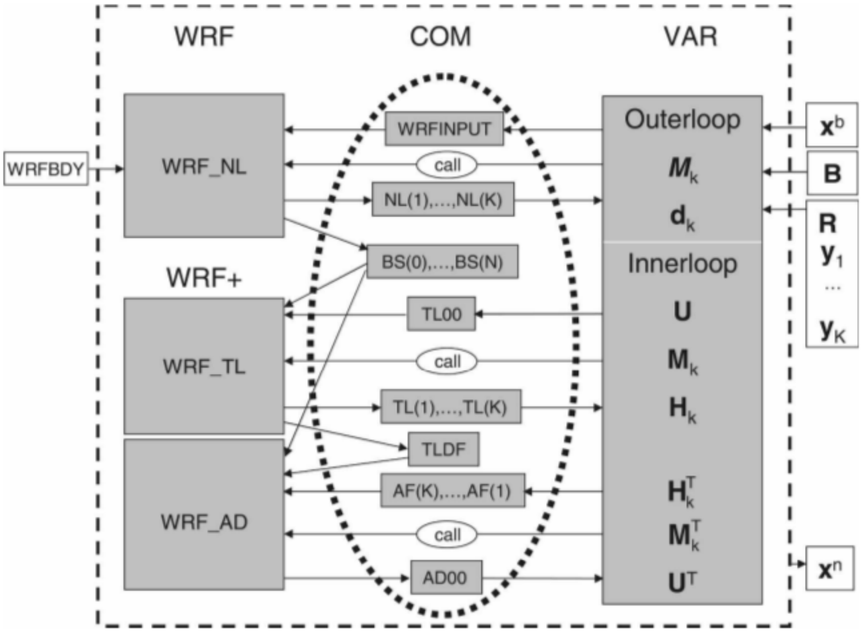


Figure 4.2: Data flow and program structure of WRF 4DVar. Source: This diagram has been taken from [15], which is the main reference used for the description of WRF 4DVar.

Figure 4.2 depicts the data flow and the structure of WRF 4DVar. The files used for the communication between VAR, WRF and WRFPLUS are shown in the middle part of the diagram and are described below:

WRFINPUT — The full model state at the beginning of each outer loop, written out by VAR and read in by WRF as an initial model state.

NL(1), ..., NL(K) — K model states (one for each observation window) produced by WRF and read in VAR before computing the innovation vector $\mathbf{d}_k$.

BS(0), ..., BS(N) — Here $N+1$ model states (one for each time step) produced by WRF, read in by WRFPLUS as basic states.

TL00 — The initial model state for the tangent linear model, written out by VAR after the U transform and read in by WRFPLUS.

TL(1), ..., TL(K) — K tangent linear model states (one for each observation window) produced by WRFPLUS during the tangent linear integration, read in VAR before computing the adjoint forcing (AF), as defined below.

TLDF — This files is responsible for the communication between the tangent linear and adjoint models, separately coded in WRFPLUS. The digital filter forcing $\sum_{i=0}^{N} f_i \mathbf{M}_i \mathbf{U} \mathbf{v}^n$ computed by WRF_TL at the end of the tangent linear integration is then read in by WRF_AD in the beginning of the adjoint integration.

AF(K), ..., AF(1) — K files containing the adjoint forcings $\mathbf{H}_k^T \mathbf{R}^{-1}\{\mathbf{H}_k \mathbf{M}_k \mathbf{U} \mathbf{v}^n - \mathbf{d}_k\}$ (one for each observation window k) are written by VAR and read in by WRFPLUS during the adjoint integration.

AD00 — The output of WRFPLUS after the adjoint integration, read in by VAR before the $\mathbf{U}^T$ transform.

As in the case of the WRF 3DVar assimilation, WRF 4DVar can also run both in cold-start and cycling modes. The advantage of WRF 4DVar analysis compared to the 3D-Var is its flow-dependent structure in the increments, in contrast with 3DVar, which uses the static covariance in the analysis and thus, the resulting increments correspond exactly to the structure of the background error covariance.

4.4 Hybrid Variational-Ensemble

WRF 3DVar assimilation system assumes that the background error covariance is static and nearly homogeneous and isotropic. In reality, the background error covariance depends on the errors of the day. WRF 4DVar implicitly includes a time-involving background error covariance through the evolution of initial errors introduced by the tangent linear dynamics. However, the flow-dependent covariance may be limited due to the assumption of the static background error covariance in the beginning of each 4D-Var assimilation cycle. It is possible to use ensemble information in the variational data assimilation framework in order to improve the analysis estimate.

Hamill and Snyder [13] suggested a Hybrid EnKF-3DVar scheme in which, the background error covariance can be represented by a weighted sum of the static $\mathbf{B}_s$ and the ensemble covariance $\mathbf{P}_e$, i.e.,

$$\mathbf{P}_b = (1-a)\mathbf{P}_e + a\mathbf{B}_s,$$

where a is a tunable parameter between 0 (pure EnKF) and 1 (pure 3D-Var).
This strategy has been demonstrated on simple models, but for large NWP models it is difficult to implement. Lorenc in [23] discussed how an ensemble based covariance model could be adapted conveniently to the variational framework by extending the control variables. The two different methods, eventually, proved to be theoretically equivalent to each other in [37].

In WRFDA the Hybrid DA scheme is based on the existing 3D-Var system, following Lorenc's method. The ensemble mean is updated by the hybrid scheme using the extended control variable to incorporate ensemble covariance information. The ensemble perturbations are generated by the Ensemble Transform Kalman filter (ETKF, for a detailed description refer to [36]), which is less expensive than the EnKF since it updates the ensemble perturbations in the low-dimensional ensemble subspace. Finally, this scheme is called as the hybrid ETKF-3DVAR system of WRF.

Starting with an ensemble of K background forecasts at time t_0, there are the following four steps in each Hybrid ETKF-3DVAR assimilation cycle:

1. update the ensemble mean by the hybrid ensemble-3DVAR method,
2. update the forecast perturbations using ETKF,
3. add the updated ensemble perturbations to the updated mean to generate K initial ensemble members,
4. make K forecasts starting from the K initial ensemble members forward to the next analysis time.

The first two steps are essential for the efficient design of the hybrid method and are described in the sequel. In the WRF hybrid ETKF-3DVAR, the flow-dependent ensemble covariances are incorporated in the variational minimization by extending control variables. The analysis increment of the hybrid is a sum of two terms:

$$\mathbf{x}' = \mathbf{x}'_1 + \sum_{k=1}^{K}(\mathbf{a}_k \circ \mathbf{x}^e_k), \tag{4.6}$$

where $\mathbf{x}'_1$ is the increment associated with the WRF 3DVar static background covariance and the sum corresponds to the increment associated with the flow-dependent ensemble covariance. In this term, $\mathbf{x}^e_k$ is the k-th normalized ensemble perturbation defined as $\mathbf{x}^e_k = (\mathbf{x}_k - \overline{\mathbf{x}})/\sqrt{K-1}$ where K is the ensemble size, $\mathbf{x}_k$ is the k-th ensemble forecast and $\overline{\mathbf{x}}$ is the forecast ensemble mean. The vectors $\mathbf{a}_k$ denote the extended control variances for each ensemble member and the symbol " $\circ$ " denotes the Schur product (element by element) of the vectors $\mathbf{a}_k$ and $\mathbf{x}^e_k$. The coefficient $\mathbf{a}_k$ fro each member determines the ensemble covariance localization scale.

The analysis increments $\mathbf{x}'$ can be obtained by minimization of the hybrid cost function:

$$\begin{aligned} J(\mathbf{x}'_1, \mathbf{a}) &= \beta_1 J_1 \ + \ \beta_2 J_e \ + \ J_o \\ &= \beta_1 \frac{1}{2}(\mathbf{x}'_1)^T \mathbf{B}^{-1}(\mathbf{x}'_1) \ + \ \beta_2 \frac{1}{2}(\mathbf{a})^T \mathbf{A}^{-1}(\mathbf{a}) \\ &\quad + \frac{1}{2}(\mathbf{y}^{o\prime} - \mathbf{H}\mathbf{x}')^T \mathbf{R}^{-1}(\mathbf{y}^{o\prime} - \mathbf{H}\mathbf{x}') \end{aligned} \tag{4.7}$$

Comparing (4.7) to the original WRF 3DVAR cost function, the usual background term has been replaced a weighted sum of J_1 and J_e. The term J_1 is the WRF 3DVAR background term associated with the static background error covariance $\mathbf{B}$. In the term J_e, $\mathbf{a}$ is formed as $\mathbf{a}^T = (\mathbf{a}_1^T, \mathbf{a}_2^T, \dots, \mathbf{a}_K^T)$ and is constrained by the block-diagonal (spatial) covariance matrix $\mathbf{A}$, defined as

$$A = \begin{bmatrix} \mathbf{S} & & & \\ & \mathbf{S} & & \\ & & \ddots & \\ & & & \mathbf{S} \end{bmatrix},$$

where each block $\mathbf{S}$ is the prescribed correlation matrix that constrains the spatial variation of $\mathbf{a}_k$. Moreover, J_o is the observation term, in which $\mathbf{y}^{o\prime} = \mathbf{y}^o - H(\mathbf{x}^b)$ is the innovation vector, where in the ETKF case the background forecast $\mathbf{x}^b$ is the ensemble mean forecast.
Moreover, in (4.7), β_1 and β_2 are the weights assigned to the static and the ensemble covariances, respectively. In order to conserve the total background-error variance, the weights are constrained by

$$\frac{1}{\beta_1} + \frac{1}{\beta_2} = 1.$$

In [37] it is proved that the solution of the minimization of the cost function (4.7) is equivalent to the solution obtained by the minimization of a cost function where the background error covariance is explicitly defined as a sum of the static and the ensemble covariances with localization applied through the Schur product:

$$J(\mathbf{x}'_1, \mathbf{a}) = \ \frac{1}{2}\mathbf{x}'^T \left(\frac{1}{\beta_1}\mathbf{B} + \frac{1}{\beta_2}\mathbf{P}^e \circ \mathbf{S}\right)^{-1} \mathbf{x}' + \frac{1}{2}(\mathbf{y}^{o\prime} - \mathbf{H}\mathbf{x}')^T \mathbf{R}^{-1}(\mathbf{y}^{o\prime} - \mathbf{H}\mathbf{x}'), \tag{4.8}$$

with $\mathbf{P}^e$ be the ensemble covariance matrix defined as

$$\mathbf{P}^e = \sum_{k=1}^{K} \mathbf{x}^e_k \mathbf{x}^{eT}_k.$$

Given the covariance $E\left\{\mathbf{a}_k\mathbf{a}_k^T\right\} = \mathbf{S}$ for $k = 1, \ldots, K$, the covariance of the sum in (4.6) satisfies

$$E\left\{\sum_{k=1}^{K}(\mathbf{a}_k \circ \mathbf{x}_k^e)\left[\sum_{k=1}^{K}(\mathbf{a}_k \circ \mathbf{x}_k^e)\right]^T\right\} = \mathbf{P}^e \circ \mathbf{S}.$$

Similarly to the WRF 3DVAR formulation, the terms J_1 and J_e, which correspond to the background, are preconditioned in order to accelerate the minimization:

J_1 is preconditioned by a control variable transform $\mathbf{U}_1$, relating the control variables $\mathbf{v}_1$ and the model space increments $\mathbf{x}_1'$, that is, $\mathbf{x}_1' = \mathbf{U}_1\mathbf{v}_1$, where $\mathbf{U}_1$ approximates the square root of the static covariance $\mathbf{B}$, i.e., $\mathbf{U}_1 \approx \mathbf{B}^{1/2}$.

J_e is preconditioned by a transform $\mathbf{a} = \mathbf{U}_2\mathbf{v}_2$, where $\mathbf{U}_2$ approximates the square root of the correlation matrix $\mathbf{A}$, i.e., $\mathbf{U}_2 \approx \mathbf{A}^{1/2}$.

$\mathbf{U}_1$ here is the same as in the WRF 3DVAR, while $\mathbf{U}_2$ approximates the square root of the correlation matrix $\mathbf{A}$ and is modeled using the simple horizontal recursive filter provided in the WRF hybrid DA system (described in detail in [3]).

Having updated the ensemble mean by the hybrid ensemble-3DVAr, the next part is to generate ensemble perturbations around the updated mean state. The ETKF is used to update the forecast ensemble perturbations in order to produce the analysis perturbations. Consider $\mathbf{X}^e$ to be the matrix whose columns contain the K ensemble member perturbations from the mean, and let $\mathbf{X}^a$ denote the analysis perturbations matrix. The ETKF updates $\mathbf{X}^e$ into $\mathbf{X}^a$ using a transformation matrix, which is derived within the ensemble perturbation subspace. The ETKF is formulated as:

$$\mathbf{X}^a = \Pi\mathbf{X}^e\boldsymbol{C}(\rho\Gamma + \mathbf{I})^{-1/2}\boldsymbol{C}^T, \tag{4.9}$$

where $\mathbf{I}$ the identity matrix and $\boldsymbol{C}$ contains the eigenvectors and Γ the eigenvalues of the $K \times K$ matrix

$$\frac{1}{K-1}(\mathbf{X}^e)^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{X}^e.$$

In (4.9), Π is an inflation factor and ρ is a scalar accounting for the fraction of the forecast error variance projected onto the ensemble subspace. These factors should be adaptively estimated for each data assimilation cycle using the innovation statistics.

In conclusion, the aim is to ensure that, on average, the background error variance estimate from the spread of ensembles about the ensemble mean is consistent with the background error variance estimated from the differences between the ensemble mean and the observations, i.e.,

$$\begin{aligned}&\left\{\mathbf{R}^{1/2}[\mathbf{y} - H(\overline{\mathbf{x}})]\right\}^T \left\{\mathbf{R}^{1/2}[\mathbf{y} - H(\overline{\mathbf{x}})]\right\} \\ &\approx \operatorname{tr}\left(\sum_{k=1}^{K}\left\{\mathbf{R}^{-1/2}[H(\mathbf{x}_k) - H(\overline{\mathbf{x}})]\right\} \times \left\{\mathbf{R}^{-1/2}[H(\mathbf{x}_k) - H(\overline{\mathbf{x}})]\right\}^T/(K-1) + \mathbf{I}\right),\end{aligned}$$

where $\operatorname{tr}(\cdot)$ denotes the trace of a matrix. Details on the derivation of the last equation can be found in [37].

The Hybrid Variational-Ensemble data assimilation combines the advantages of traditional variational and ensemble approaches in order to produce an analysis that is superior to that produced by each one of the pure 3D-Var and Ensemble schemes.

Chapter 5

Appendix

5.1 3D-Var for the Lorenz-96 model

In this section, 3D-Var experiments are presented, assuming a perfect model and two different observation networks. Each one of them was tested for observational error $\boldsymbol{\varepsilon}_o \sim N(0, \sigma_o^2)$, where $\sigma_o = 0.50$.

In each case, we have two different types of figures. In the first type of figure we present the evolution in time of the first four components of the true state, the analysis state and the available observations.

In the second type of figure, we have three elements: In the upper left corner, we present the analysis state compared to the true state, as well as the observation locations. In the upper right corner, we have the absolute value of the analysis error plotted against the locations at which we have observations. Finally, in the lower part, we present the evolution in time of the analysis RMSE.

5.1.1 Using the matrix B6h1

Initially, we assume that B6h1 is the constant background error covariance matrix and we present the cases shown in the following table.

Analysis RMSE for 3D-Var using B6h1		
Assimilation: Network	each time step	every 5 time steps
1. observe all	4.4403	5.1105
2. observe every 2	4.3193	4.8730

Table 5.1: 3D-Var analysis RMSE results using B6h1 matrix and assuming $\boldsymbol{\varepsilon}_o \sim N(0, \sigma_o^2)$ with $\sigma_o = 0.50$.

In the first example, we consider the first observation network, for observations which are assimilated into our system at each integration step. In Figure 5.1 we present the evolution in time of the first four components of the true state (green line), the analysis estimate (dashed black line), as well as the available observations (red dots).

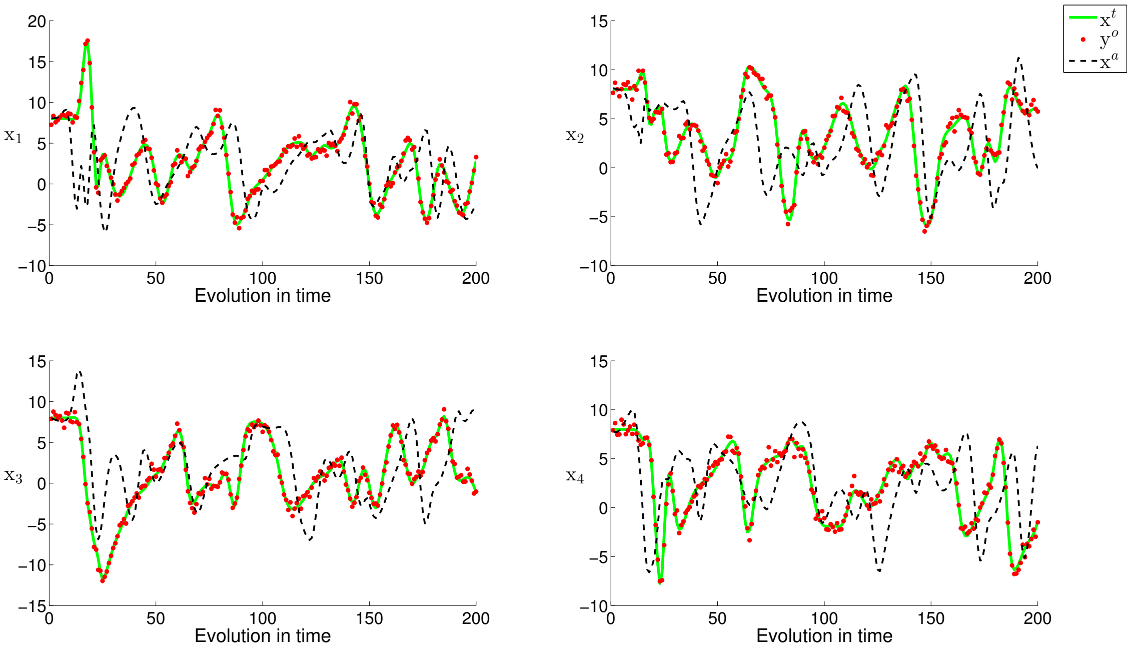


Figure 5.1: 3D-Var for the Lorenz-96 model with observation network 1, $\sigma_o = 0.50$ and assimilation performed at each time step.

For observational errors with standard deviation $\sigma_o = 0.50$, the analysis "captures" the general pattern of the true state but there is no good quality in the estimate.

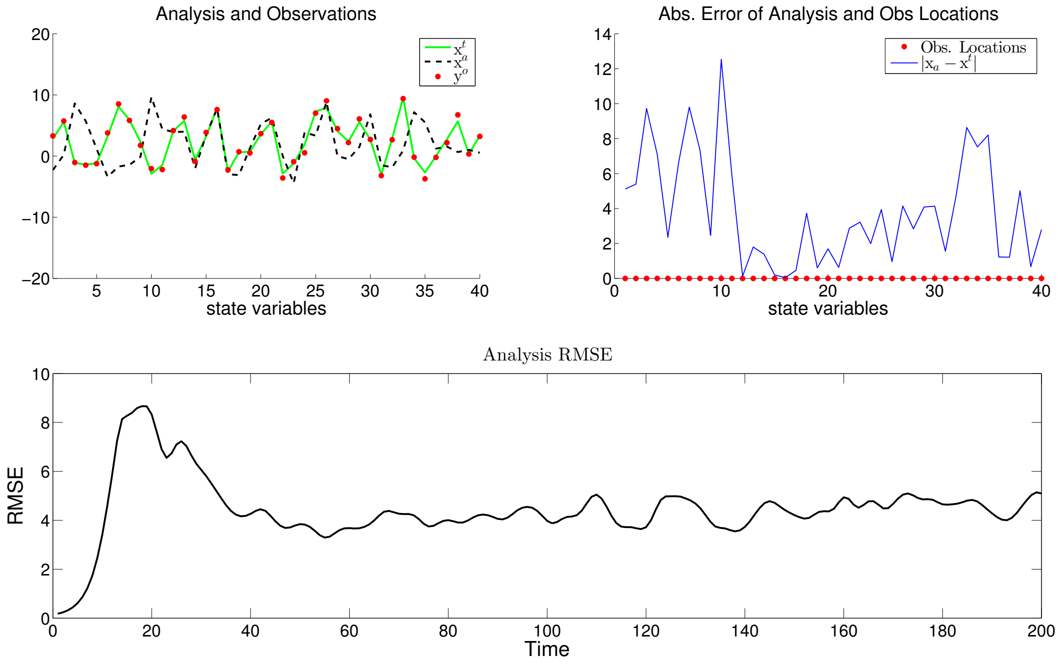


Figure 5.2: 3D-Var for the Lorenz-96 model with observation network 1 and $\sigma_o = 0.50$. The average analysis RMSE is 4.4403.

Figures 5.3 and 5.4 are for the second observation network. The analysis is trying to fit the true state in the observed components, but still the RMSE of the estimate is quite big.

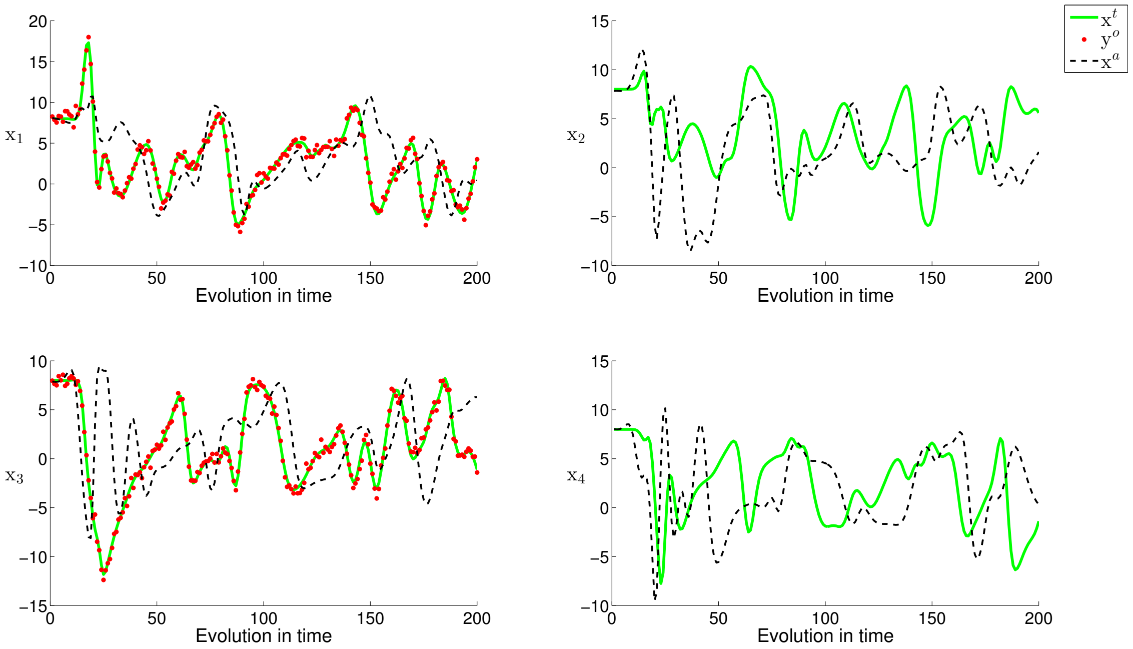


Figure 5.3: 3D-Var for the Lorenz-96 model with observation network 2, $\sigma_o = 0.50$ and assimilation performed at each time step.

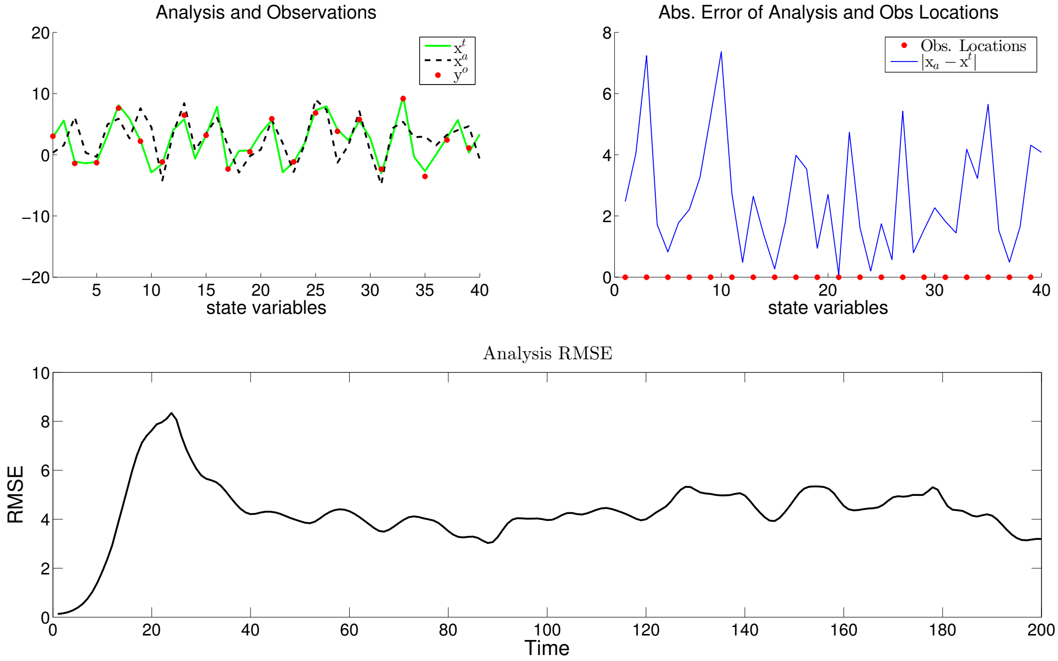


Figure 5.4: 3D-Var for the Lorenz-96 model with observation network 2 and $\sigma_o = 0.50$. The average analysis RMSE is 4.3193.

We consider now that observations are assimilated into our system every 5 integration steps. In Figures 5.5 and 5.6 we have the results for the first observation network, while in Figures 5.7 and 5.8 the results for the second network. In both cases, the analysis produced is not close to the true state, causing high analysis RMSE.

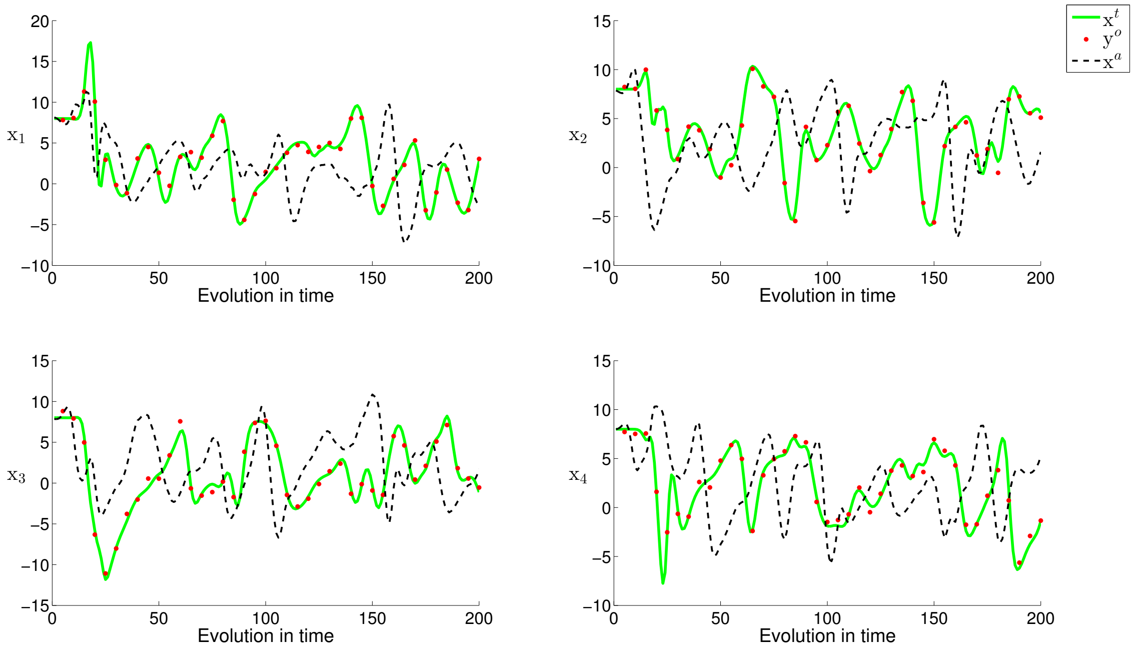


Figure 5.5: 3D-Var for the Lorenz-96 model with observation network 1, $\sigma_o = 0.50$ and assimilation performed every 5 integration steps.

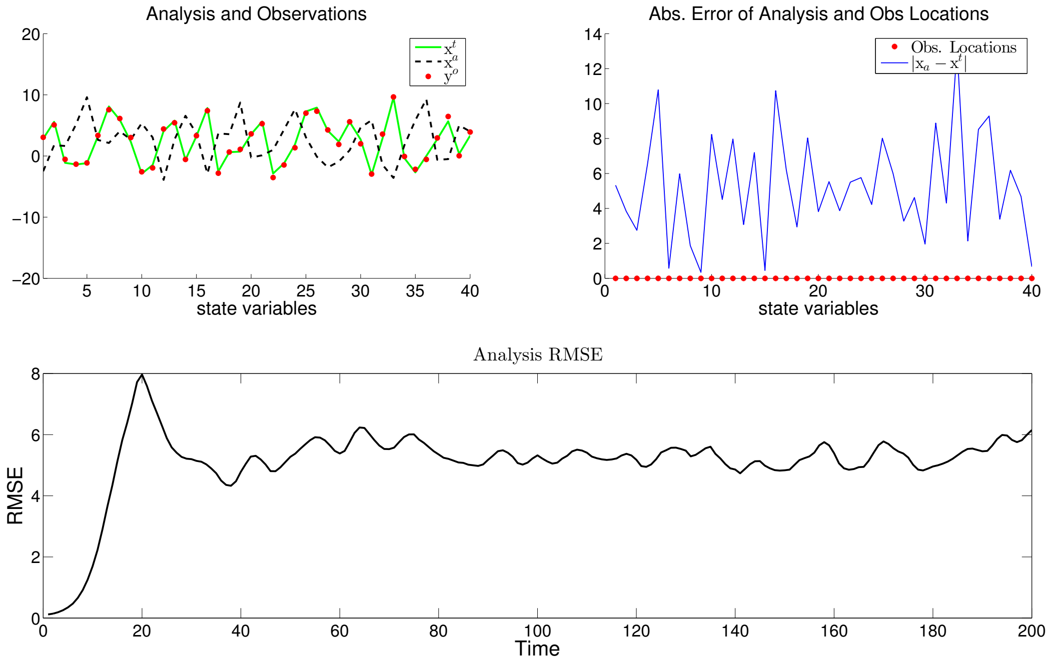


Figure 5.6: 3D-Var for the Lorenz-96 model with observation network 1 and $\sigma_o = 0.50$. The average analysis RMSE is 5.1105.

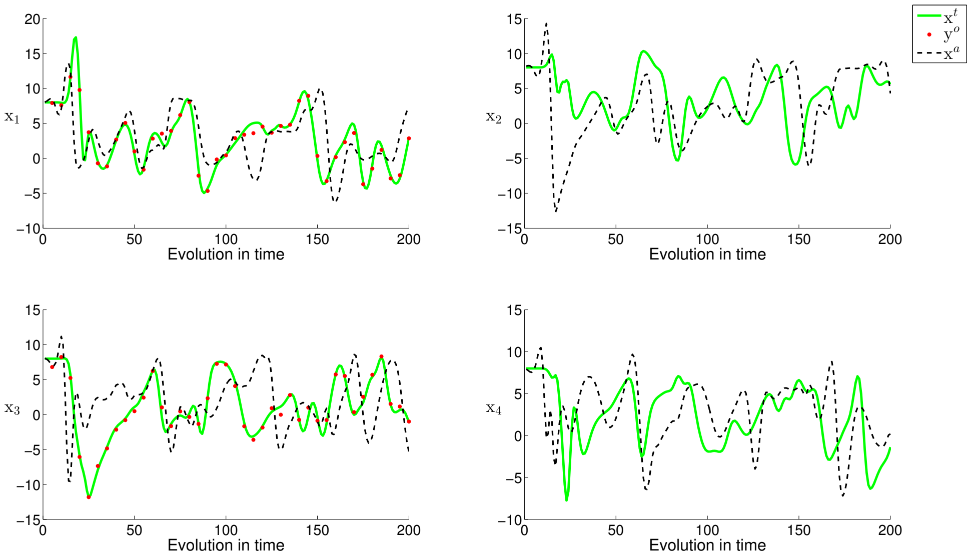


Figure 5.7: 3D-Var for the Lorenz-96 model with observation network 2, $\sigma_o = 0.50$ and assimilation performed every 5 integration steps.

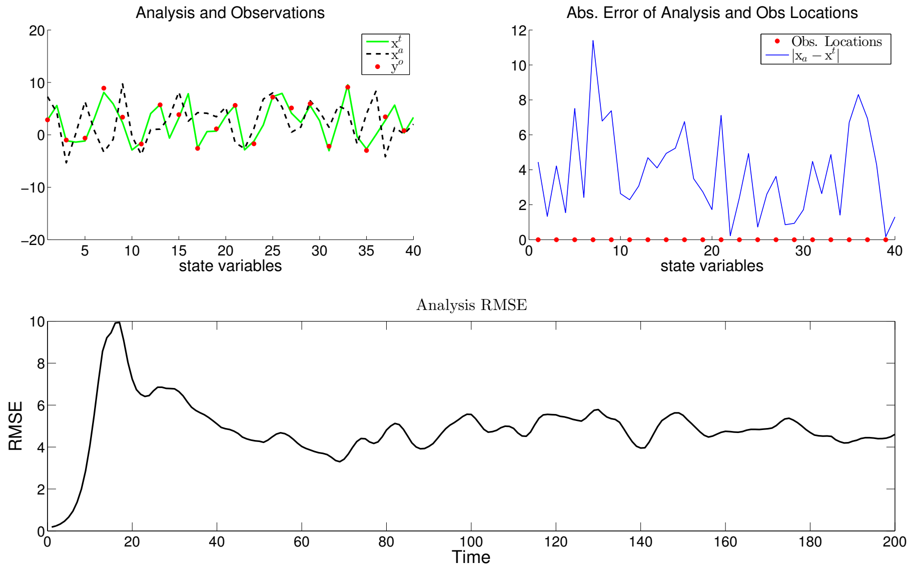


Figure 5.8: 3D-Var for the Lorenz-96 model with observation network 2 and $\sigma_o = 0.50$. The average analysis RMSE is 4.8730.

5.1.2 Using the matrix Bloc

We assume now that Bloc is the background error covariance matrix. The results presenting in the sequel are summarized in the following table.

Analysis RMSE for 3D-Var using Bloc		
Assimilation: Network	each time step	every 5 time steps
1. observe all	1.6891	3.9178
2. observe every 2	3.3563	4.4278

Table 5.2: 3D-Var analysis RMSE results using Bloc matrix and assuming $\varepsilon_o \sim N(0, \sigma_o^2)$ with $\sigma_o = 0.50$.

In Figures 5.9 and 5.10, where observations are available at each location and we have used the localized background error covariance, 3D-Var gives a pretty small RMSE and the analysis estimate is very close to the true state.

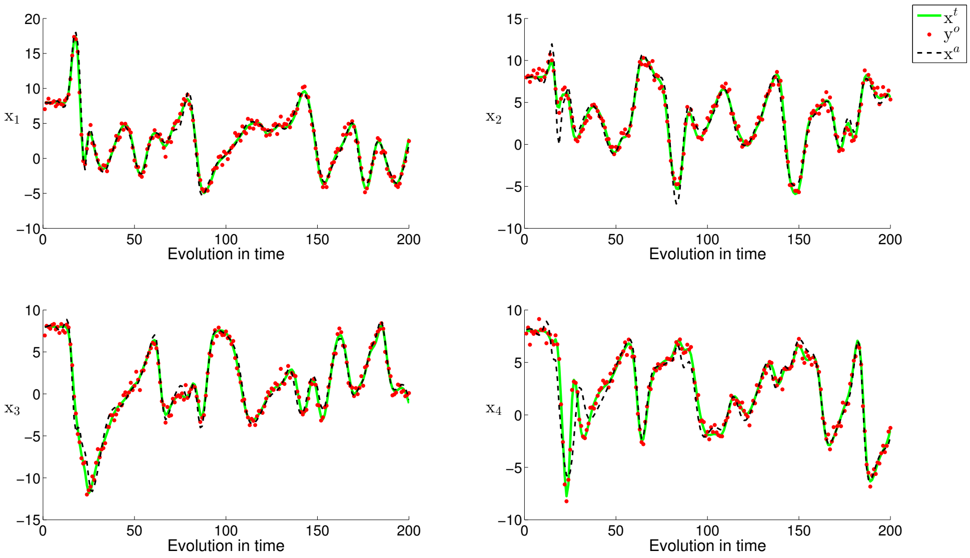


Figure 5.9: 3D-Var for the Lorenz-96 model with observation network 1, $\sigma_o = 0.50$ and assimilation performed at each time step.

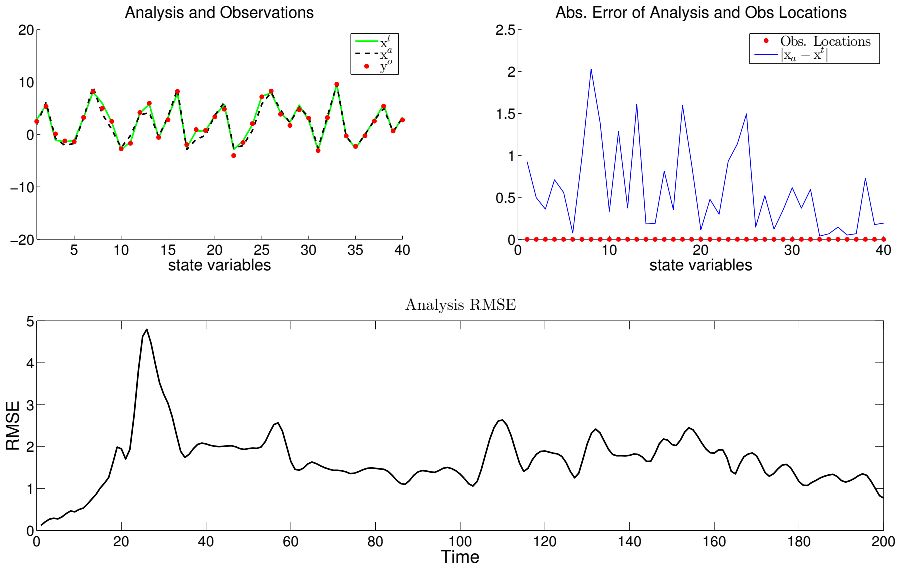


Figure 5.10: 3D-Var for the Lorenz-96 model with observation network 1 and $\sigma_o = 0.50$. The average analysis RMSE is 1.6891.

We continue with the results for the second observation network, in Figures 5.11 and 5.12. The analysis is a relatively good estimate of the true state at the observed components (except the first time steps), while at the unobserved components we see that is also close to the truth in some points.

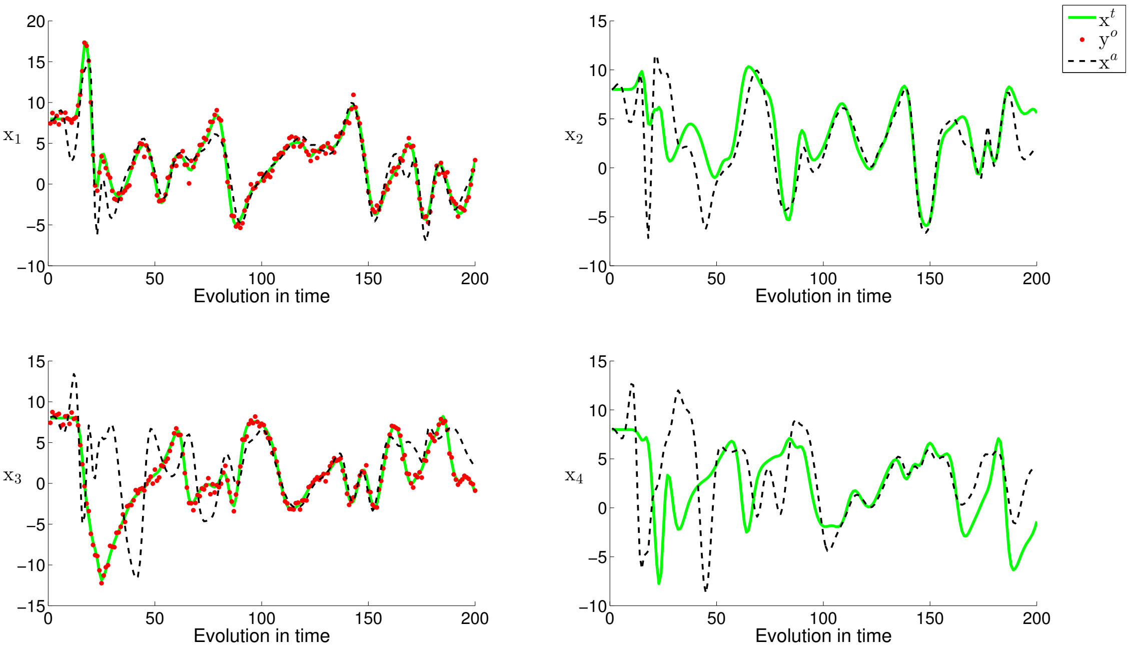


Figure 5.11: 3D-Var for the Lorenz-96 model with observation network 2, $\sigma_o = 0.50$ and assimilation performed at each time step.

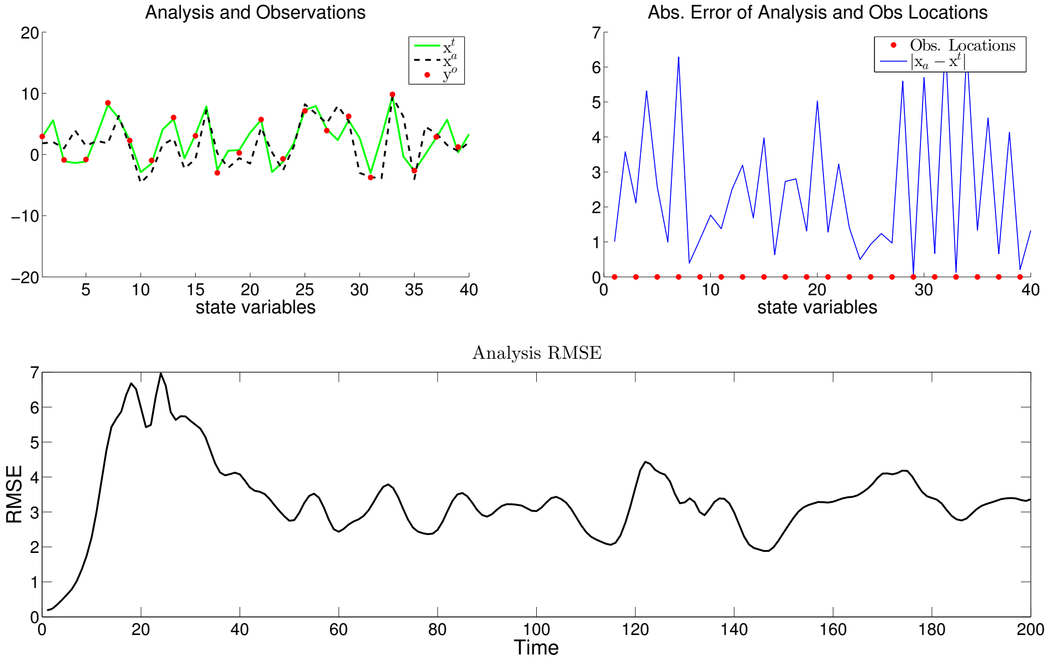


Figure 5.12: 3D-Var for the Lorenz-96 model with observation network 2 and $\sigma_o = 0.50$. The average analysis RMSE is 3.3563.

Assuming now that observations are assimilated into our system every 5 integration steps for the first observation network, we have the results in Figures 5.13 and 5.14.

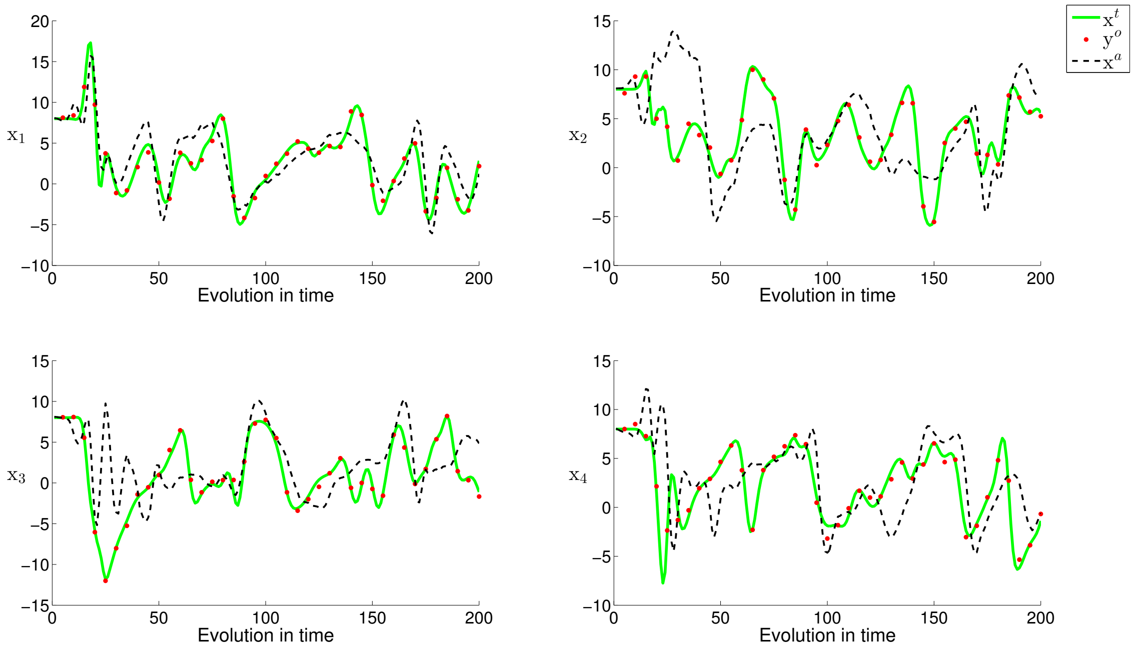


Figure 5.13: 3D-Var for the Lorenz-96 model with observation network 1, $\sigma_o = 0.50$ and assimilation performed every 5 integration steps.

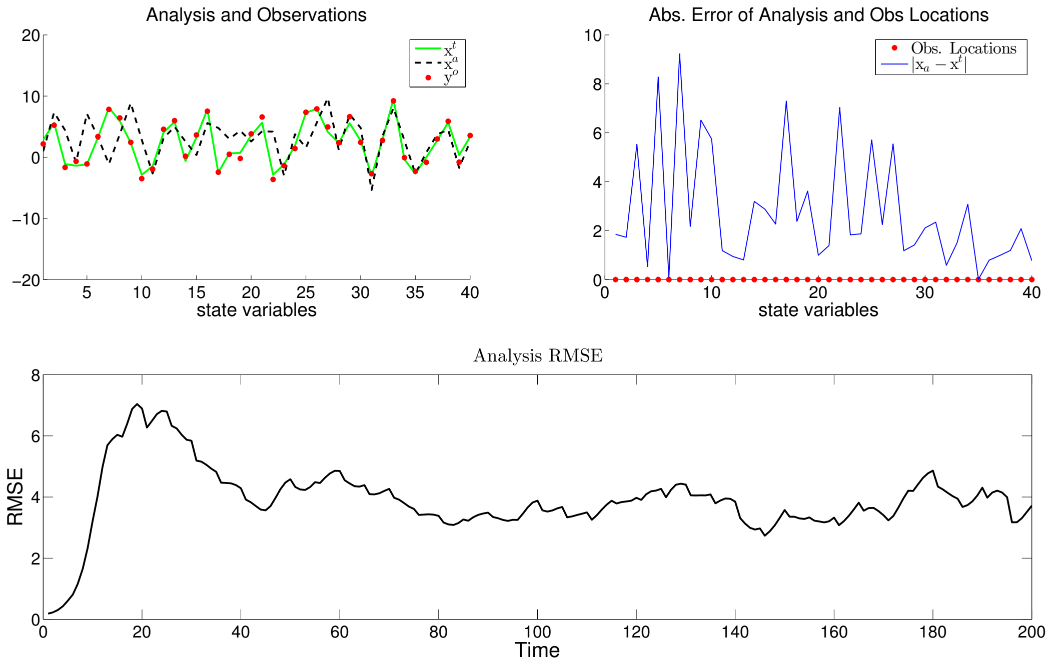


Figure 5.14: 3D-Var for the Lorenz-96 model with observation network 1 and $\sigma_o = 0.50$. The average analysis RMSE is 3.9178.

Considering now the second observation network for assimilation every 5 integration steps, we present the results in Figures 5.15 and 5.16.

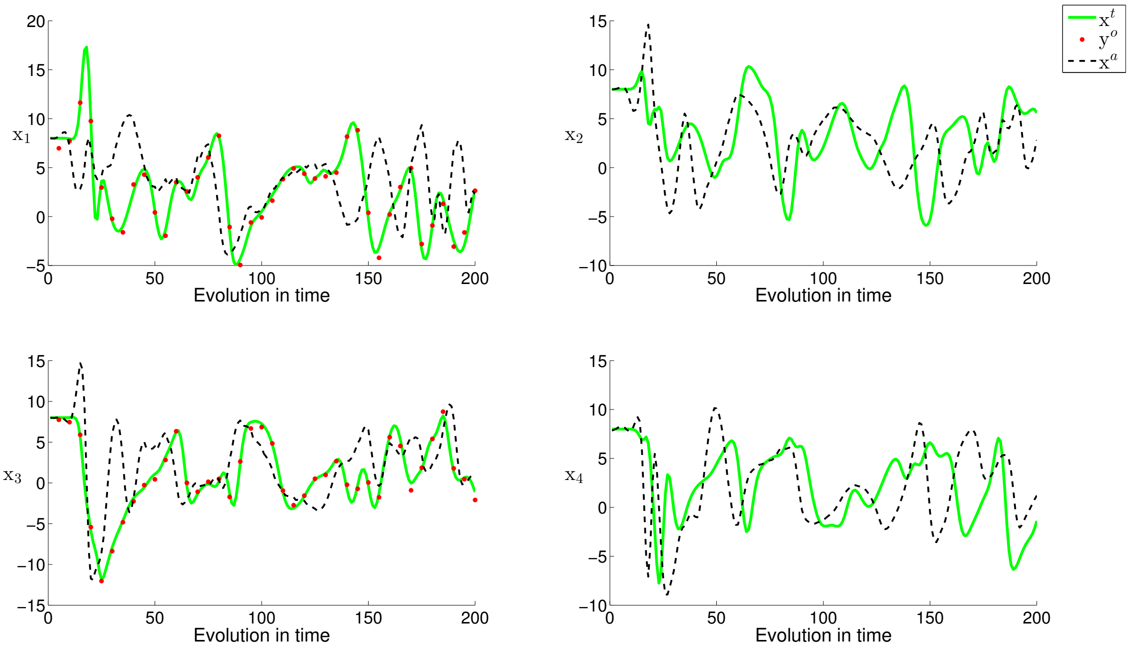


Figure 5.15: 3D-Var for the Lorenz-96 model with observation network 2, $\sigma_o = 0.50$ and assimilation performed every 5 integration steps.

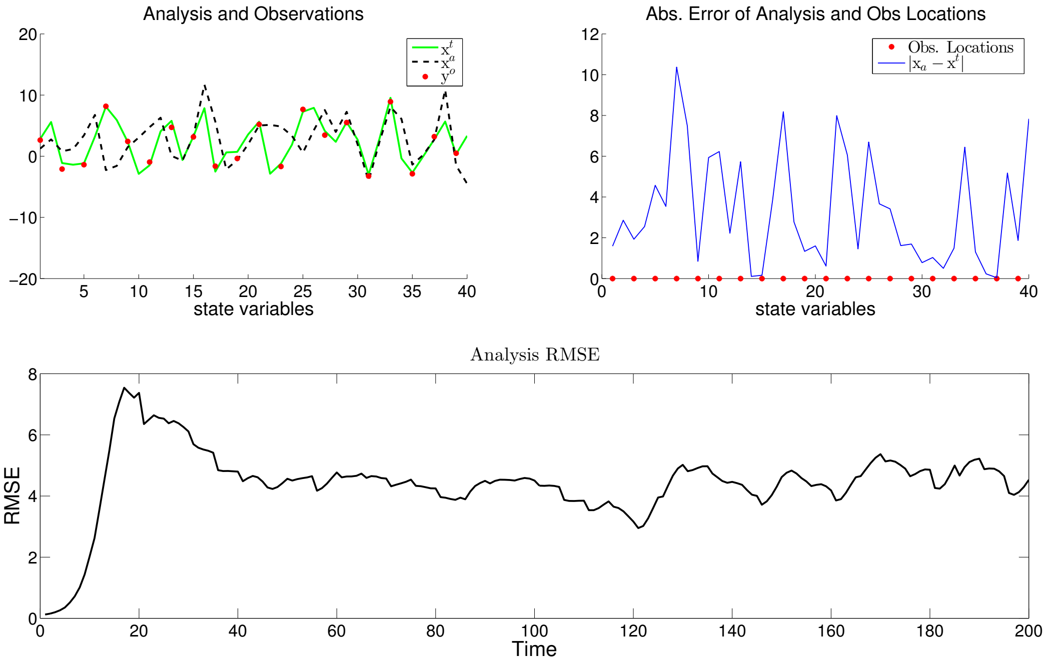


Figure 5.16: 3D-Var for the Lorenz-96 model with observation network 2 and $\sigma_o = 0.50$. The average analysis RMSE is 4.4278.

5.2 EnSRF for the Lorenz-96 model

We present additional results of the EnSRF for the Lorenz-96, considering the same cases as in section 3.2.1 for an observational error $\varepsilon_o \sim N(0, \sigma_o^2)$, for a standard deviation $\sigma_o = 0.50$.

5.2.1 Case study I

Considering an ensemble of 10 members and observations at each site, which become available every 50 time-steps having an observation error $\varepsilon_o \sim N(0, \sigma_o^2)$. In Figure 5.17 we present the time evolution of the first four components of the true state, the ensemble members and their mean, as well as the available observations. Then, in Figure 5.18 we present the RMSE of the analysis estimate as a function of time.

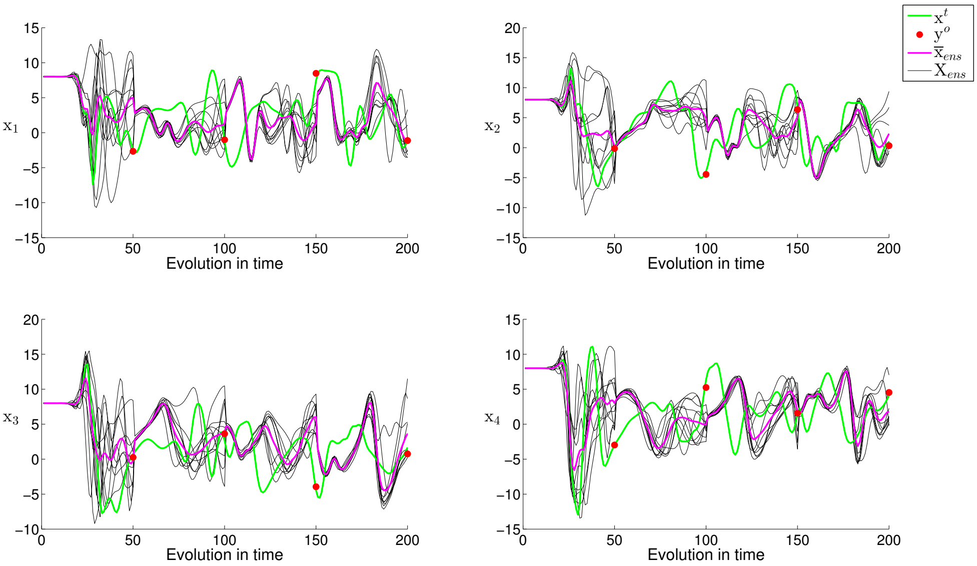


Figure 5.17: EnSRF for the Lorenz-96 with $K = 10$, observation network 1 and $\sigma_o = 0.50$.

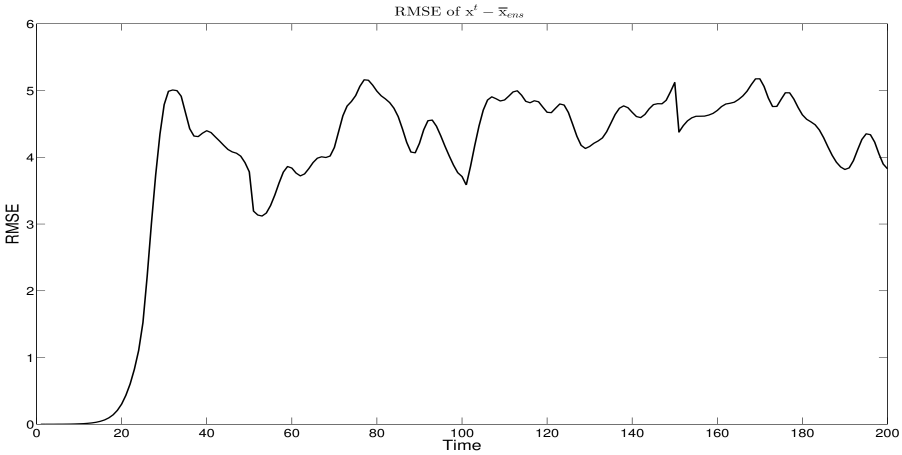


Figure 5.18: EnSRF analysis RMSE results for $K = 10$, observation network 1 and $\sigma_o = 0.50$. The mean value of the RMSE is 3.8939.

The same test-case for an ensemble of 40 members gives the results presented in Figures 5.19 and 5.20.

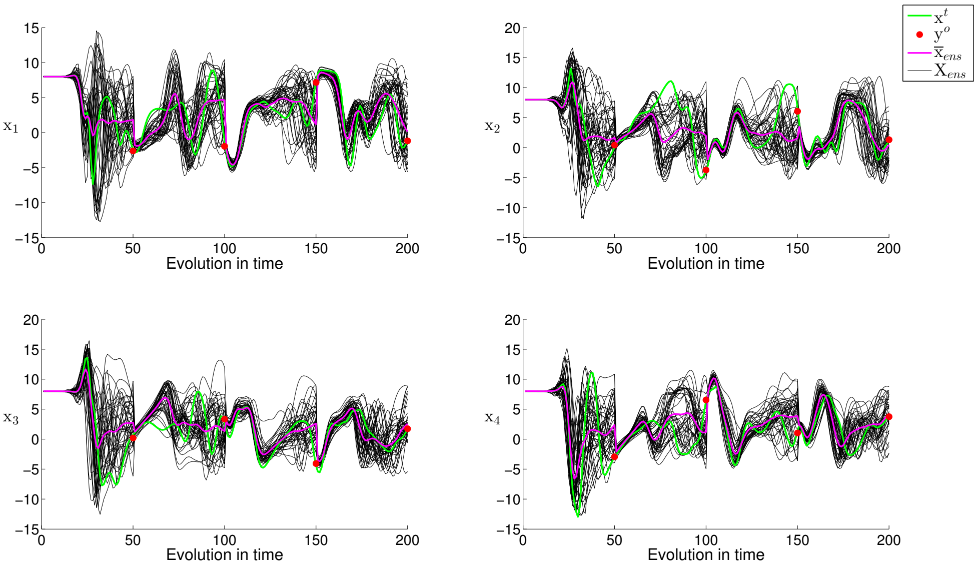


Figure 5.19: EnSRF for the Lorenz-96 with $K = 40$, observation network 1 and $\sigma_o = 0.50$.

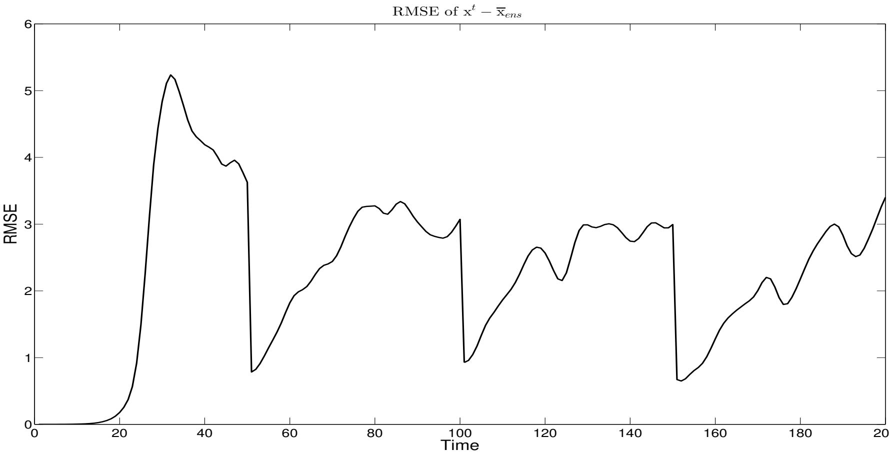


Figure 5.20: EnSRF analysis RMSE results for $K = 40$, observation network 1 and $\sigma_o = 0.50$. The mean value of the RMSE is 2.2899.

It is obvious that for an ensemble with a greater size we obtain a better analysis estimate and therefore, a smaller analysis error.

If we use covariance inflation (both multiplicative and additive) with parameters $\gamma = 1.2$ and $\ell = 0.05$ we get the results showing at Figures 5.21 and 5.18.

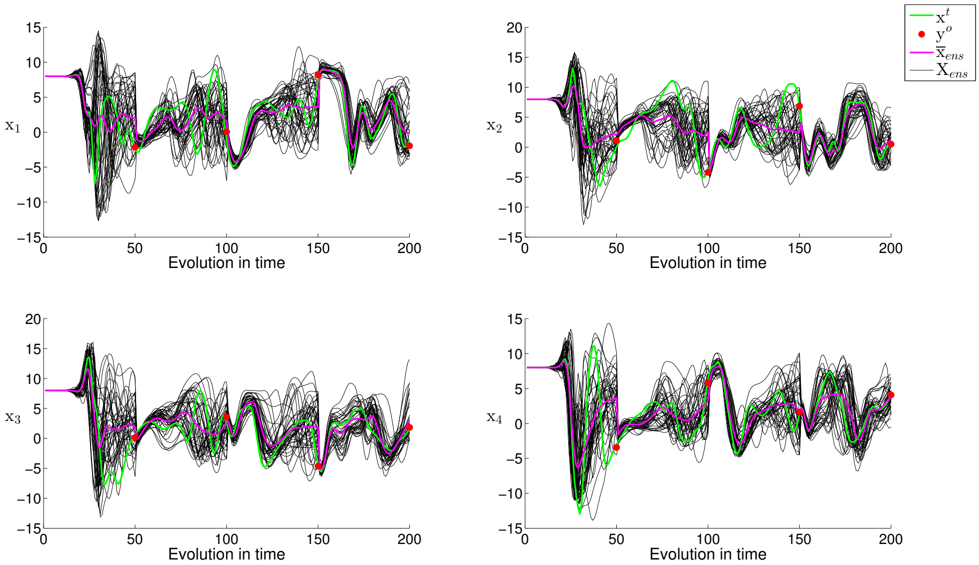


Figure 5.21: EnSRF for the Lorenz-96 with $K = 40$, observation network 1, $\sigma_o = 0.50$, inflation factors $\gamma = 1.2$ and $\ell = 0.05$.

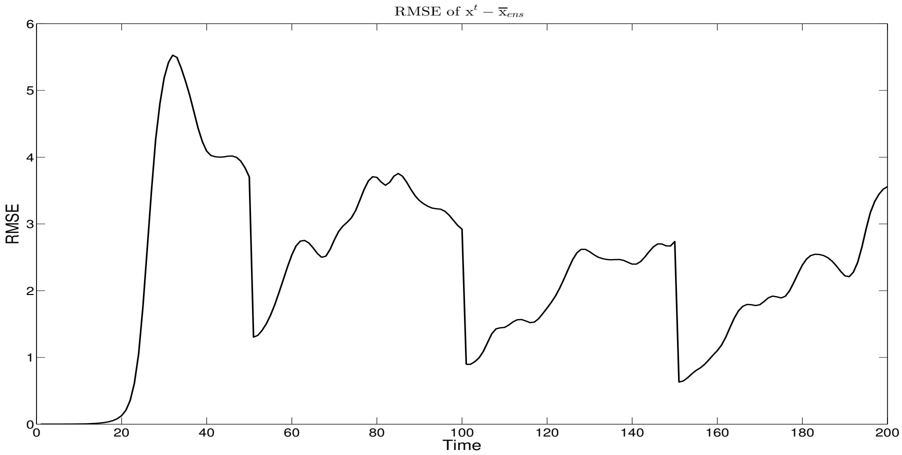


Figure 5.22: EnSRF analysis RMSE results for $K = 40$, observation network 1 and $\sigma_o = 0.50$. The mean value of the RMSE is 2.2923.

Comparing the Figures 5.20 and 5.22, we obtain that the ensemble mean is a better estimate of the true state when covariance inflation is applied, causing the RMSE to be reduced from 2.6365 to 2.0629.

5.2.2 Case study II

In the previous examples, we assumed that both the true and the ensemble are propagating in time through the same model dynamics, i.e., we assumed a perfect forecast model. In general this is not the case, so we are going to assume now that the forecast model is slightly different from the true. We assume that the true model has a forcing term $F = 8$, while the forecast model has $F = 8.2$. Considering an ensemble of 40 members and observations available at each location with $\sigma_o = 0.50$, we compare the results obtained with and without the use of inflation.

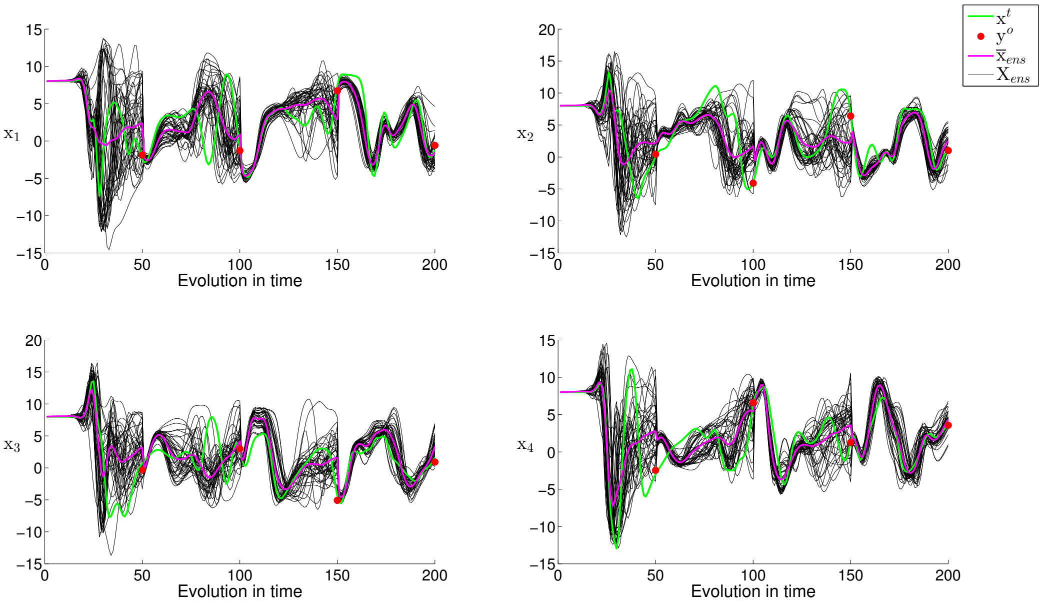


Figure 5.23: EnSRF for the Lorenz-96 with $K = 40$, observation network 1, $\sigma_o = 0.50$ and forecast model with forcing term $F = 8.2$.

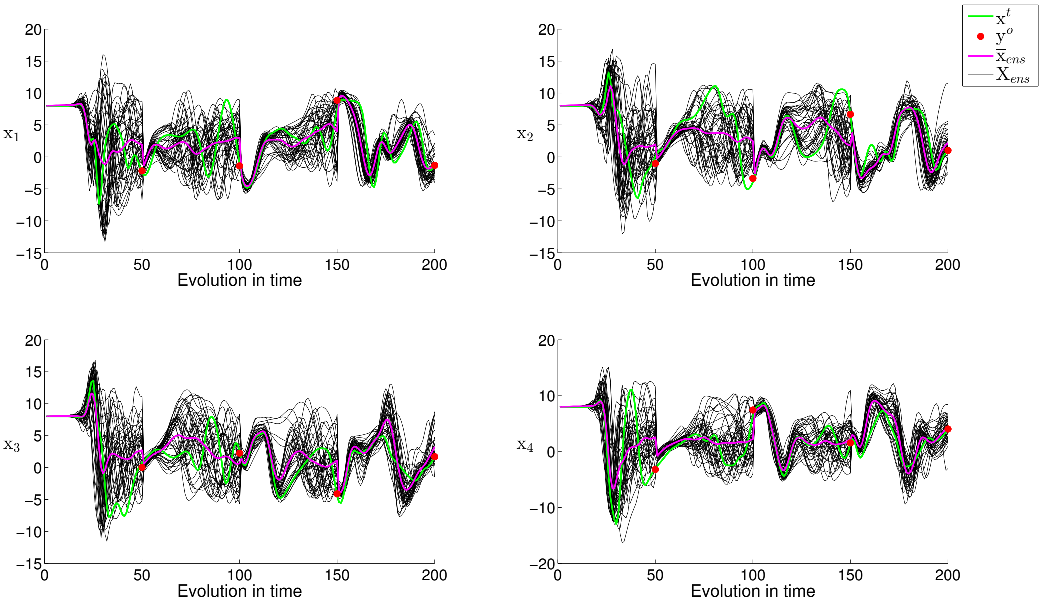


Figure 5.24: EnSRF for the Lorenz-96 with $K = 40$, observation network 1, $\sigma_o = 0.50$, assuming a forecast model with forcing term $F = 8.2$ and inflation factors $\gamma = 1.2$ and $\ell = 0.05$.

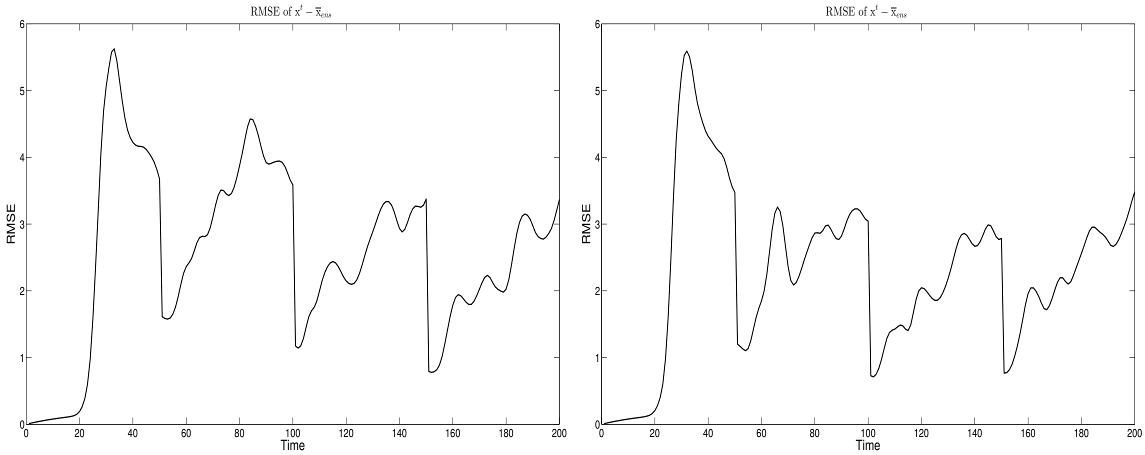


Figure 5.25: EnSRF analysis RMSE results for $K = 40$, observation network 1, $\sigma_o = 0.50$, assuming a forecast model with forcing term $F = 8.2$. In the left plot we do not use inflation, while in the right plot the inflation factors are $\gamma = 1.2$ and $\ell = 0.05$.

In the presence of covariance inflation, the RMSE appears to have smaller peaks between the assimilations, resulting an error reduction from 2.5574 (no inflation) to 2.2674 (inflation).

5.2.3 Case study III

We consider now an example using an ensemble of 40 members but now we assume that the observations become available at every two sites and therefore, we expect that the analysis estimate will be less accurate on the unobserved variables.

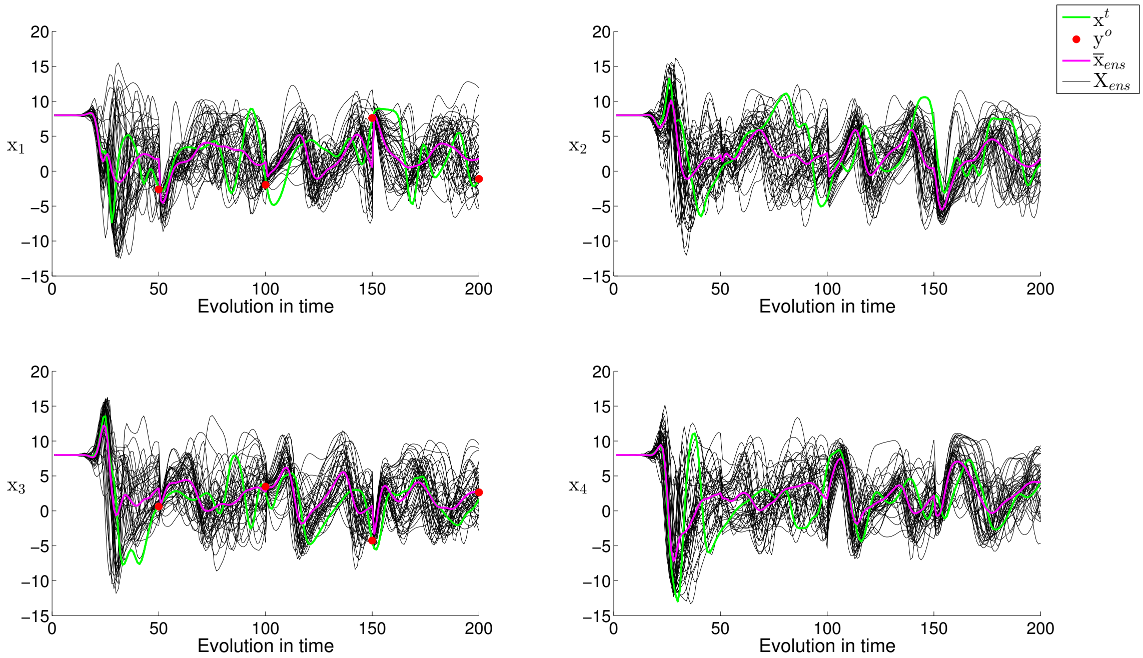


Figure 5.26: EnSRF for the Lorenz-96 with $K = 40$, observation network 2 and $\sigma_o = 0.50$.

In Figure 5.26, we observe that after the assimilation of each observation, the ensemble does not concentrate around the true state. This is due to the fact that the observations are available only on half of the sites. For the unobserved sites, their evolution in time depends only on the model dynamics and thus, their behavior is almost chaotic. In Figure 5.27 we have the evolution in time of an observed and an unobserved component of the model state. As can be seen, the ensemble mean is not a good estimate of the true state, since there is less information available.

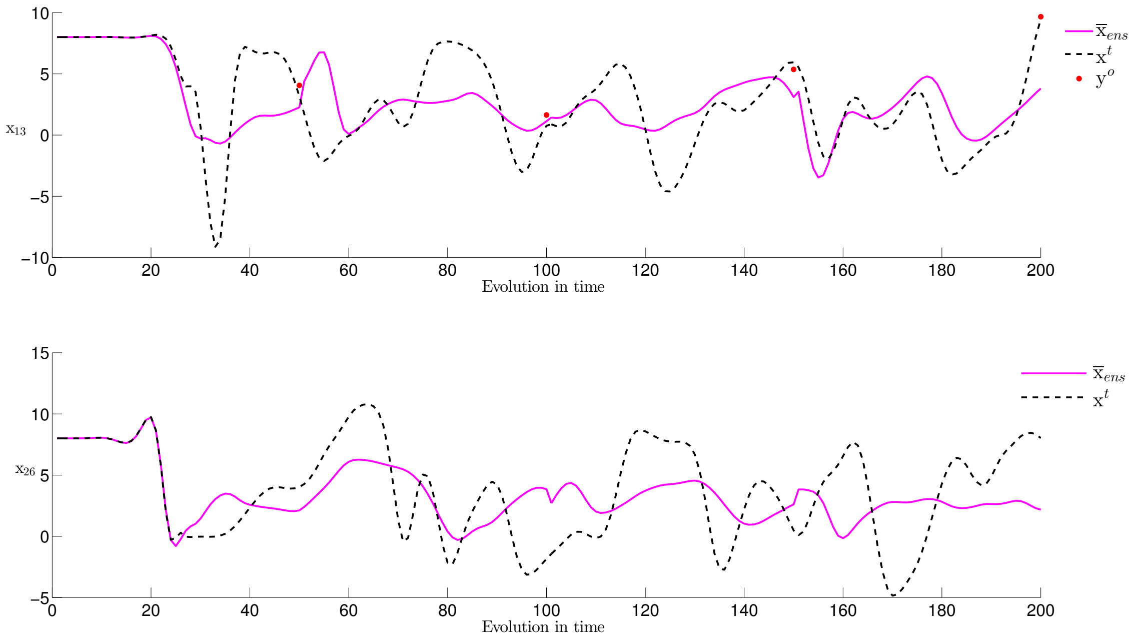


Figure 5.27: EnSRF for the Lorenz-96 with $K = 40$, observation network 2 and $\sigma_o = 0.50$. Evolution in time of the 13th and 26th components, which are observed and unobserved respectively.

The RMSE of the estimate is plotted against time in the figure below. The error's behavior is quite different from the previous test case (Figure 5.25), in which, after each observation was assimilated there was a significant error reduction.

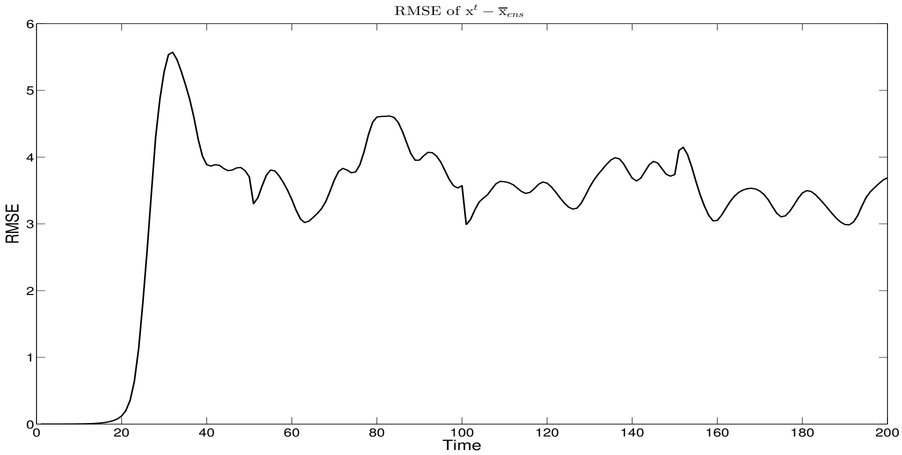


Figure 5.28: EnSRF analysis RMSE results for $K = 40$, observation network 2 and $\sigma_o = 0.50$. The mean value of the RMSE is 3.2518.

Bibliography

[1] Georgios D. Akrivis and Vassilios A. Dougalis. *Numerical Methods for Ordinary Differential Equations*. Crete University Press, Heraklion, 2006.

[2] D. Barker, X. Y. Huang, Z. Liu, T. Auligné, X. Zhang, S. Rugg, R. Ajjaji, A. Bourgeois, J. Bray, Y. Chen, M. Demirtas, Y. R. Guo, T. Henderson, W. Huang, H. C. Lin, J. Michalakes, S. Rizvi, and X. Zhang. The Weather Research and Forecasting Model's Community Variational/Ensemble Data Assimilation System: WRFDA. *Bulletin of the American Meteorological Society*, 93(6):831–843, 2012.

[3] D. M. Barker, W. Huang, Y.-R. Guo, and A. J. Bourgeois. A three-dimensional variational (3DVAR) data assimilation system for use with MM5. Technical report, NCAR. NCAR/TN-453 + STR, 68pp. [Available from UCAR Communications, P.O. Box 3000, Boulder, CO, 80307.], 2003.

[4] D. M. Barker, W. Huang, Y.-R. Guo, A. J. Bourgeois, and Q. N. Xiao. A Three-Dimensional Variational Data Assimilation System for MM5: Implementation and Initial Results. *Monthly Weather Review*, 132(4):897–914, 2004.

[5] Gerrit Burgers, Peter van Leeuwen, and Geir Evensen. Analysis scheme in the ensemble Kalman filter. *Monthly weather review*, 126(6):1719–1724, 1998.

[6] P. Courtier, J.-N. Thépaut, and A. Hollingsworth. A strategy for operational implementation of 4D-Var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120(519):1367–1387, 1994.

[7] Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994.

[8] Geir Evensen. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367, 2003.

[9] Mike Fisher. Background error covariance modelling. In *Seminar on Recent Development in Data Assimilation for Atmosphere and Ocean*, pages 45–63, 2003.

[10] Melina A. Freitag and Roland W. E. Potthast. Synergy of inverse problems and data assimilation techniques. In Mike Cullen, Melina A Freitag, Stefan Kindermann, and Robert Scheichl, editors, *Large Scale Inverse Problems*, volume 13 of *Radon Series on Computational and Applied Mathematics*. Walter de Gruyter, Berlin, 2013.

[11] Pierre Gauthier and Jean-Noel Thepaut. Impact of the digital filter as a weak constraint in the preoperational 4DVAR assimilation system of Météo-France. *Monthly weather review*, 129(8):2089–2102, 2001.

[12] N. Gustafsson. Use of a digital filter as weak constraint in variational data assimilation. In *Proceedings of a workshop on variational assimilation, with special emphasis on three-dimensional aspects*, 1992.

[13] Thomas M. Hamill and Chris Snyder. A hybrid ensemble Kalman filter-3D variational analysis scheme. *Monthly Weather Review*, 128(8):2905–2919, 2000.

[14] Peter L. Houtekamer and Herschel L. Mitchell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1):123–137, 2001.

[15] X. Y. Huang, Q. Xiao, D. M. Barker, X. Zhang, J. Michalakes, W. Huang, T. Henderson, J. Bray, Y. Chen, Z. Ma, J. Dudhia, Y. Guo, X. Zhang, D. J. Won, H. C. Lin, and Y. H. Kuo. Four-Dimensional Variational Data Assimilation for WRF: Formulation and Preliminary Results. *Monthly Weather Review*, 137(1):299–314, 2009.

[16] Brian R. Hunt, Eric J. Kostelich, and Istvan Szunyogh. Efficient data assimilation for spatiotemporal chaos: A Local Ensemble Transform Kalman Filter. *Physica D: Nonlinear Phenomena*, 230(1):112–126, 2007.

[17] Rudolph 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.

[18] Eugenia Kalnay. *Atmospheric modeling, data assimilation, and predictability.* Cambridge University Press, 2003.

[19] Martin Leutbecher. A data assimilation tutorial based on the Lorenz-95 system. ECMWF, May 5, 2004.

[20] John M. Lewis, S. Lakshmivarahan, and Sudarshan Dhall. *Dynamic Data Assimilation: A Least Squares Approach*, volume 13. Cambridge University Press, 2006.

[21] Andrew C. Lorenc. Analysis Methods for Numerical Weather Prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474):1177–1194, 1986.

[22] Andrew C. Lorenc. Modelling of error covariances by 4D-Var data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 129(595):3167–3182, 2003.

[23] Andrew C. Lorenc. The potential of the ensemble Kalman filter for NWP - a comparison with 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 129(595):3183–3203, 2003.

[24] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

[25] Edward N. Lorenz. Predictability: A problem partly solved. In *Seminar on Predictability*, volume 1, pages 1–18, Reading, Berkshire, UK, 1996. ECMWF.

[26] Edward N. Lorenz and Kerry A. Emanuel. Optimal Sites for Supplementary Weather Observations: Simulation with a Small Model. *Journal of the Atmospheric Sciences*, 55(3):399–414, 1998.

[27] Peter Lynch and Xiang-Yu Huang. Initialization of the HIRLAM Model Using a Digital Filter. *Monthly Weather Review*, 120(6):1019–1034, 1992.

[28] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. 1979.

[29] Edward Ott, Brian R. Hunt, Istvan Szunyogh, Aleksey V. Zimin, Eric J. Kostelich, Matteo Corazza, Eugenia Kalnay, DJ Patil, and James A. Yorke. A Local Ensemble Kalman Filter for Atmospheric Data Assimilation. *Tellus A*, 56(5):415–428, 2004.

[30] David F. Parrish and John C. Derber. The National Meteorological Center's Spectral Statistical-Interpolation Analysis System. *Monthly Weather Review*, 120(8):1747–1763, 1992.

[31] Hendrik Reich, Daniel Leuenberger, and Michael Würsch. The Lorenz95 system. DWD-HErZ Winter school on Data Assimilation, Offenbach, 13-17 February, 2012.

[32] J. B. Klemp J. Dudhia D. O. Gill D. M. Barker M. G Duda X.-Y. Huang W. Wang Skamarock, W. C. and J. G. Powers. A Description of the Advanced Research WRF Version 3. NCAR Tech. Note NCAR/TN-475+STR, 113 pp. Technical report, National Center for Atmospheric Research.

[33] Olivier Talagrand. Assimilation of Observations, an Introduction. *Journal of the Meteorological Society of Japan. Ser. II*, 75:81–99, 1997.

[34] F. Veersé and J.-N. Thépaut. Multiple-truncation incremental approach for four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 124(550):1889–1908, 1998.

[35] X. Wang, D. M. Barker, C. Snyder, and T. M. Hamill. A Hybrid ETKF–3DVAR Data Assimilation Scheme for the WRF Model. Part I: Observing System Simulation Experiment. *Monthly Weather Review*, 136(12):5116–5131, 2008.

[36] X. Wang and C. H. Bishop. A comparison of breeding and ensemble transform Kalman filter ensemble forecast schemes. *Journal of the atmospheric sciences*, 60(9):1140–1158, 2003.

[37] X. Wang, T. M. Hamill, J. S. Whitaker, and C. H. Bishop. A comparison of hybrid ensemble transform Kalman filter-optimum interpolation and ensemble square root filter analysis schemes. *Monthly weather review*, 135(3):1055–1076, 2007.

[38] T.-K. Wee and Y.-H. Kuo. Impact of a digital filter as a weak constraint in MM5 4DVAR: An observing system simulation experiment. *Monthly weather review*, 132(2):543–559, 2004.

[39] Jeffrey S. Whitaker and Thomas M. Hamill. Ensemble Data Assimilation without Perturbed Observations. *Monthly Weather Review*, 130(7):1913–1924, 2002.