Diss. ETH No. 24468

ENSEMBLE KALMAN PARTICLE FILTERS FOR HIGH-DIMENSIONAL DATA ASSIMILATION

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZÜRICH
(Dr. sc. ETH Zürich)

presented by

SYLVAIN ROBERT

MSc in Statistics ETH Zürich
born on 06.01.1987
citizen of Fribourg

accepted on the recommendation of

Prof. Peter Bühlmann, examiner
Prof. Hans R. Künsch, co-examiner
Dr. Daniel Leuenberger, co-examiner

2017

Instead of fearing wrong predictions, we look eagerly for them; it is only when predictions based on our present knowledge fail that probability theory leads us to fundamental new knowledge.

– Edwin T. Jaynes

Men fear thought as they fear nothing else on earth – more than ruin, more even than death. Thought is subversive and revolutionary, destructive and terrible, thought is merciless to privilege, established institutions, and comfortable habits; thought is anarchic and lawless, indifferent to authority, careless of the well-tried wisdom of the ages. Thought looks into the pit of hell and is not afraid... Thought is great and swift and free, the light of the world, and the chief glory of man.

– Bertrand Russell

Acknowledgments

My special thanks go to all the people without whom I would never have been able to write this thesis and accomplish my PhD. First, I would like to express my immense gratitude to Prof. Hansruedi Künsch, my "Doktorvater", who guided me along this path with his experience and wisdom. He taught me to express my thoughts clearly and mathematically, to carefully check results, and dominate the impetus of my youth. He gave me a glimpse into the world of scientific research and rigorous mathematical thinking, and for that and many other things I will be forever grateful.

I would also like to deeply thank Dr. Daniel Leuenberger, without whom I would never have been able to apply the methods that we developed to any interesting problem. He introduced me to the arcane of supercomputing, made it look easy to run complex experiments on hundreds of cores and interpret their results. I cannot count how many times he saved me when I was lost in the midst of all these scripts and files, and did not see the point of what we were doing.

Thank you also to Prof. Peter Bühlmann, who accepted to play the role of my official supervisor, but above all whose good humor and cheerful spirit always permeated through the whole SfS. Speaking of the latter, I would like to thank all of my colleagues, past and present, who made our little corner of ETH such a lively place to work in. In particular, thank you to the secretaries, Susan, Cecilia and Letizia, who made all the administrative clutter look like a bliss. Thank you as well to all my PhD colleagues with whom I shared this experience, and to the consulting and WBL team as well, for the serious and the less serious times.

Finally, I would like to thank my friends and family who always supported me and provided me with such a nourishing and positive environment. In no particular order, thank you Olivier, Ivan, Simon, Elena, Matteo, Sarah, Carlotta, Alan, Ruben, Loris, Père, Mère, Antoine, and so many more. Last but not least, thank you my dear Sarah for always believing in me and for making my life beautiful.

Contents

Abstract

Data assimilation consists in estimating the state of a system, for example the atmosphere in numerical weather prediction (NWP), by combining information coming from the dynamical laws of the system with a stream of observations. Because of the presence of observational noise and uncertainty in the initial conditions, a probabilistic instead of a deterministic approach is to be preferred. The goal is thus to estimate the time evolution of the distribution of the system state conditioned on all the past observations. Ensemble data assimilation methods, such as the ensemble Kalman filter (EnKF), solve this problem by representing the distribution of the state with a finite sample of particles which follow the dynamical laws of the system.

What makes data assimilation for geophysical applications particularly challenging is that the dimension of the state to estimate is extremely high (order of 10^8), while the ensemble size is limited to less than 100 due to heavy computational costs. At the same time, the increasing resolution of the physical models makes Gaussian assumptions, on which the EnKF relies, less and less valid. In the present thesis we propose extensions to the ensemble Kalman particle filter (EnKPF), a hybrid algorithm which relaxes some of the Gaussian assumptions by combining the EnKF with the particle filter (PF). The goal of these extensions is to make the EnKPF suitable for very high-dimensional applications.

The first contribution consists in proposing two localized versions of the algorithm: the naive-LEnKPF and the block-LEnKPF. The naive-LEnKPF, similar to the local EnKF (LEnKF), works by assimilating data in local windows and then patching the results together. It has the advantage to be simple and efficient, but it does not address the issue of discontinuities introduced by the PF part of the algorithm. The block-LEnKPF, on the

other hand, assimilates the observations by blocks and limits their influence to a local area while smoothing out the introduced discontinuities. Both local EnKPFs are applied to an artificial model of cumulus convection of medium dimensionality. The results of the numerical experiments show that the new algorithms perform at a similar level to the LEnKF, and bring some noticeable improvements for non-Gaussian variables such as the precipitation field.

The second main contribution of this thesis is to propose a new algorithm, the ensemble transform Kalman particle filter (ETKPF). It is based on a reformulation of the EnKPF in ensemble space, which allows it to be easily and efficiently implemented in an existing full-scale NWP data assimilation framework. Furthermore, the ETKPF replaces the stochastic part of the algorithm with a deterministic scheme, such that it has exact second moment instead of only on expectation. The algorithm was tested on a challenging high-dimensional application at convective scale with COSMO, in a setup similar to the one used operationally at MeteoSwiss. The results of the experiments show the feasibility of the new algorithm in real-world applications and encourage further developments in the direction of localized hybrid particle filters for high-dimensional data assimilation.

Résumé

L'assimilation de données consiste à estimer l'état d'un système, tel que l'atmosphère dans le cas des prédictions météorologiques, en combinant les connaissances des lois qui gouvernent la dynamique du système avec les observations. En général une approche probabilistique plutôt que déterministe est préférée, pour prendre en compte l'incertitude associée aux observations et aux conditions initiales. L'assimilation de données consiste donc à estimer l'évolution temporelle de la distribution de l'état du système d'intérêt, étant données toutes les observations obtenues jusqu'alors. Les methodes d'assimilation d'ensemble, comme par example le *ensemble Kalman filter* (EnKF), accomplissent cette tâche en utilisant un échantillon de différentes realisations possibles du processus étudié pour representer ces distributions.

Ce qui rend ce genre d'applications particulièrement difficile c'est que le nombre de dimensions du système d'intérêt est extremement élevé (de l'ordre de 10^8), alors que la taille des échantillons que l'on peut se permettre est de moins que 100, en raison des larges coûts de calcul associés avec les modèles numériques de l'atmosphère. De plus, les phénomènes non linéaires qui apparaissent dans les modèles à hautes résolution, comme par example la convection, posent de sérieux problèmes aux méthodes qui présupposent des distribution normales, comme le EnKF. Ce travail de thèse propose des modifications du *ensemble Kalman particle filter* (EnKPF), un algorithme hybride entre le EnKF et le *particle filter* (PF), qui dépend moins fortement de la normalité des distributions. La tâche principale de ces modifications est de rendre le EnKPF adapté à des applications à très haute dimensionalité comme on en rencontre en prédictions météorologiques.

La première contribution de cette thèse est de proposer deux versions *loca-*

lisées de l'algorithme en question : le naive-LEnKPF et le block-LEnKPF. Le naive-LEnKPF fonctionne de manière similaire au EnKF local (LEnKF) : il assimile les données dans des régions restreintes et les combine ensuite ensemble pour former l'estimation totale. Il a l'avantage d'être relativement simple à formuler et efficace à calculer, mais il ne prend pas en compte le problème des *cassures* introduites par la partie PF de l'algorithme. Le block-LEnKPF adresse ce problème en assimilant les données par blocs, dont il limite l'influence à un rayon local, et dans un deuxième temps en adoucissant les cassures introduites. Les deux algorithmes sont ensuite testés sur un modèle artificiel convection de dimension moyenne. Les résultats des expériences numériques démontrent que les nouveaux algorithmes ont en général des performances similaires au LEnKF. De plus, ils apportent des améliorations notables pour la prédiction de quantités particulièrement non Gaussiennes, comme le champs de précipitations.

La seconde contribution principale de ce travail de thèse est un nouvel algorithme, le *ensemble transform Kalman particle filter* (ETKPF), qui exploite une reformulation géometrique de l'EnKPF et conduit à une implémentation de l'algorithme particulièrement efficace d'un point de vue computationelle dans le modèle numérique de prédictions météorologiques existant. De plus, l'ETKPF remplace une partie aléatoire de l'EnKPF par un procédé déterministe qui guarantie que l'ensemble a un second moment exacte et non pas seulement en moyenne. Des expériences avec une application difficile en présence de convection ont été effectuées dans un context similaire à celui employé à MeteoSuisse de façon opérationelle. Les résultats obtenus démontrent que le nouvel algorithme peut être appliqué à des situations réalistes, et nous encouragent à continuer le dévelopement de nouvelles méthodes hybrides basées sur le PF pour l'assimilation de données dans les problèmes à haute dimensionalité.

Chapter 1

Introduction

Numerical weather prediction (NWP) is the product of a large enterprise which has been very successful in the past 40 years (Bauer et al., 2015). This success depends on multiple actors – academia, governments and the private sector – and on multiple factors – increase in computing power, data collection and sharing, etc. One key development was the realization that deterministic forecasts are of limited power because of the chaotic nature of the dynamical laws governing the atmosphere (Lorenz, 1963). To address this limitation, ensemble methods have thus been developed and are used operationally at weather centers since the 1990s (Lewis, 2005). Instead of one very accurate prediction, an ensemble of 50 to 100 different realizations of the forecast is generated, from which one can draw a *probabilistic* instead of a deterministic forecast. Working with such probabilistic forecasts is not only a necessity due to the chaotic nature of the atmosphere, but it is also very useful for many applications of interest, such as evaluating weather-related risks (e.g. floods and hurricanes) or managing renewable energy production (e.g. solar panels and wind farms).

In general, to be able to produce a probabilistic forecast one needs to incorporate all the different sources of uncertainty into the model. This is an extremely challenging task, as there are many of them: There are errors associated with the discretization of the 3-dimensional fields representing the atmosphere, errors associated with the numerical integration schemes of the dynamical laws governing the evolution of the system, errors due to the parameterization of phenomena happening at unresolved scales (e.g. turbulence or cloud formation), unknown parameters, errors in the obser-

vations and so on. Producing probabilistic forecasts is therefore a complex task with many components (Gneiting and Raftery, 2005): representation of model errors, bias correction, post-processing of ensembles, etc.

In the present work we focus on one of those components, called *data assimilation*, which is the determination of the initial conditions of the ensemble from which to start the forecast. Ideally, the initial ensemble should incorporate all the information coming from the past observations, taking into account their associated uncertainty. To do so in a sequential manner one alternates between a prediction step, where the ensemble is propagated through the dynamical laws of the system, and an analysis step, where the new observations are incorporated in the ensemble by using Bayes' theorem.

Due to the high computational cost associated with the integration of the dynamical model, the size of ensembles run at operational centers is very small (less than 100) compared to the dimension of the state of interest (around 10^8 for full-scale NWP) and the number of observations assimilated at each step (around 10^6). In practice, therefore, one looks for approximate but efficient methods rather than aim for the best theoretical solution. The state-of-the-art ensemble data assimilation method is the Ensemble Kalman Filter (EnKF) (Evensen, 1994, 2003), which is essentially a Monte Carlo approximation of the Kalman filter (Kalman, 1960; Kalman and Bucy, 1961). One limitation of the EnKF, however, is that it is consistent only in the case where the distribution of the ensemble is Gaussian. This assumption is acceptable with coarse model grids and "classical" observations, but with high-resolution models and observations such as radar data, non-Gaussianity becomes the norm. Indeed, with high-resolution models (horizontal resolution of the order of 1 km), non-linear phenomena such as convection are represented, which lead to strong non-Gaussianity in the ensemble, for example a bimodal distribution where one mode has a cloud somewhere and one has none. Therefore, there is an urgent need for new data assimilation methods able to handle non-Gaussian distributions while remaining computationally feasible (Bauer et al., 2015). The main goal of this work is to contribute towards a solution to this challenging task.

It should be noted that the methods discussed and developed here are not limited to data assimilation for NWP, but can be equally well used for other large-scale geophysical applications such as reservoir modeling in oil industry, underground water resources estimation, ocean circulation, climate models and many more. We limit ourselves to weather prediction

as it is a task complex enough to require all of our attention.

In Sections 1.1 and 1.2 we present in more detail the type of problems that we are addressing, and in Section 1.3 the existing algorithms to tackle them. In Section 1.4 we discuss the particular difficulties associated with the application of data assimilation algorithms to high-dimensional systems, while in Section 1.5 we present localization, the typical technique used to address those difficulties. In Section 1.6 we discuss the ensemble space formulation of the analysis. In Section 1.7 we briefly mention the various models on which we have tested our algorithms. Finally, in Section 1.8 we give an outline of the remainder of this thesis.

1.1 State space models

The problem of studying a dynamical system through partial and noisy observations fits nicely into the framework of general state space models (see for example Künsch (2001)). In such a framework the system of interest is characterized by the state, $x \in R^q$, a Markov process which describes the time evolution of the state, and an observation process which describes the relation between the state and the observations $y \in R^d$. More precisely we can write the model as:

$$x_t = f_t(x_{t-1}, \psi_t), \tag{1.1}$$
$$y_t = h_t(x_t, \xi_t), \tag{1.2}$$

where (ψ_t) and (ξ_t) are independent noise processes. In geophysical applications, the state x_t is typically a discretized multivariate 3-dimensional field describing the system of interest, for example the atmosphere. The dynamical process in Eq. (1.1) describes the time evolution of the state, typically given by the numerical integration of stochastic or deterministic partial differential equations. The process in Eq. (1.2) describes where and with what measurement errors the system is observed. A graphical representation of such a model is given in Fig. 1.1.

The state space model described in Eqs. (1.1) and (1.2) is completely general, but one is often interested in more specific cases. Instead of arbitrary influence of the noise processes, one often considers additive noise only:

$$x_t = f_t(x_{t-1}) + \psi_t, \tag{1.3}$$
$$y_t = h_t(x_t) + \xi_t, \tag{1.4}$$

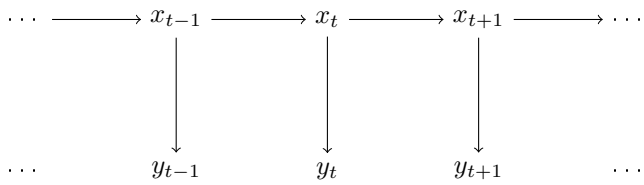

Figure 1.1: Graphical representation of a general state-space model.

where (ψ_t) and (ξ_t) are independent noise processes, as above. Constraining the model even further, one can restrict $f_t(\cdot)$ and $h_t(\cdot)$ to be linear, leading to so-called linear dynamical models:

$$x_t = F_t x_{t-1} + \psi_t, \tag{1.5}$$
$$y_t = H_t x_t + \xi_t, \tag{1.6}$$

where F_t and H_t are matrices, potentially varying in time.

Concerning the distribution of the noise processes various assumptions can be made as well. A typical choice is to assume that both ψ_t and ξ_t are Normal, say $\psi_t \sim \mathcal{N}(0, Q_t)$ and $\xi_t \sim \mathcal{N}(0, R_t)$. Such linear and Gaussian systems are attractive because they are tractable, as we will see below. However, many applications do not fit in this framework. In the geophysical systems that interest us, such as in NWP, we typically consider models of the following type:

$$x_t = f_t(x_{t-1}), \tag{1.7}$$
$$y_t = h_t(x_t) + \xi_t, \quad \xi_t \sim \mathcal{N}(0, R_t). \tag{1.8}$$

The dynamical process (x_t) is non-linear and deterministic, as it corresponds to the solution of partial differential equations embodying physical laws. However, even if the system is deterministic, it might be of interest to add a stochastic component to account for errors in the model. This stochastic component can enter either additively as Gaussian noise, or more indirectly through stochastic parametrization in the numerical integration of the state (see for example Arnold et al. (2013)). In the present work we will focus on the deterministic case and consider the function $f_t(\cdot)$ to be essentially a black box.

The observation operator $h_t(\cdot)$ is often non-linear. For example in NWP applications, it can be the interpolation of gridded quantities such as temperature or wind speed at specific locations, or the expected quantity of

rainfall computed from the whole column of atmosphere. One often makes the assumption that $h_t(\cdot)$ is non-linear but can be linearized in a neighborhood of x_t, thus making some derivations analytically tractable. In most of the applications that interest us, we assume the observation noise ξ_t to be Gaussian for convenience.

1.2 Bayesian filtering

There are essentially two types of questions that one can ask about a state space model: either the system is not perfectly known and one is interested in *learning* it, for example by estimating parameters, as the matrices F_t in Eq. (1.5) and the observation covariance R_t; or the system is assumed to be known and one is interested in recovering the sequence $x_{1:t} = \{x_1, \ldots, x_t\}$ of unobserved state, from a sequence of noisy observations $y_{1:t} = \{y_1, \ldots, y_t\}$. In the present work we are interested in the latter task, which is called *filtering* in the signal processing literature, and *data assimilation* in geophysical applications. Henceforth we will use both terms interchangeably according to convenience. Using $\pi(\cdot)$ to denote a distribution or its corresponding density depending on the context, we are interested in the following distributions:

— $\pi_t(x_{1:t}|y_{1:t})$, the joint distribution of the sequence $x_{1:t}$ given the whole history of observations,

— $\pi^a_t(x_t|y_{1:t})$, the marginal distribution of the above at x_t, also called the *filtering* or *analysis* distribution, and

— $\pi^b_t(x_t|y_{1:t-1})$, the *predictive* or *background* distribution of x_t given the sequence of observations up to the previous time point.

Another related distribution of interest is $\pi^s_{k|t}(x_k|y_{1:t})$, where $k < t$, the *smoothing distribution*, but we will not discuss it here as it is outside the scope of this work. If we could somehow obtain $\pi_t(x_{1:t}|y_{1:t})$ then everything else would be straightforward. However the dimension of this distribution can be very large and keeps growing as time goes on. The smoothing distribution is used for *reanalysis*, but the filtering distribution is sufficient to determine initial conditions for prediction. Therefore, for practical reasons people have been interested in recursive solutions of the filtering distribution which can make use of an incoming stream of observations.

The missing ingredient to establish this recursion is the distribution of the observation y_t given x_t, which is referred to as the likelihood when considered as a function of x_t, and which we write as $\ell_t(x_t|y_t)$. For example in the Gaussian system of Eq. (1.8), this likelihood is the normal density with mean $h_t(x_t)$ and covariance R_t evaluated at y_t, written as $\phi(y_t; h_t(x_t), R_t)$. Using Bayes' formula we can relate the analysis distribution to the background distribution using the likelihood as:

$$\pi_t^a(x_t|y_{1:t}) \propto \pi_t^b(x_t|y_{1:t-1}) \cdot \ell_t(x_t|y_t), \tag{1.9}$$

where we use the fact that y_t given x_t is independent of $y_{1:t-1}$ and where the sign $\propto$ means proportional up to a normalizing constant. In the Bayesian terminology, the background distribution is the *prior*, while the analysis distribution is the *posterior*. From this equation the following recursive algorithm emerges naturally:

1. Start with an initial prior distribution $\pi_0(x_0)$.
2. For $t = 1, \ldots, T$:
   (a) *Forecast*: $\pi_t^b(x_t|y_{1:t-1}) = \int \pi_t^f(x_t|x_{t-1}) \cdot \pi_{t-1}^a(x_{t-1}|y_{1:t-1}) dx_{t-1}$,
   (b) *Analysis*: $\pi_t^a(x_t|y_{1:t}) \propto \pi_t^b(x_t|y_{1:t-1}) \cdot \ell_t(x_t|y_t)$,

where by convention $\pi_0^a(x_0|y_{1:0}) = \pi_0(x_0)$, and $\pi^f(x_t|x_{t-1})$ can be directly obtained from the model specification in Eq. (1.1); for example in the case of the deterministic dynamical system of Eq. (1.7), $\pi^f(x_t|x_{t-1})$ is simply the Dirac function at the values $f_t(x_{t-1})$, which essentially forecasts x_t from x_{t-1}. It should be noted that in the filtering literature the forecast step is typically called the propagation and the analysis step the update.

It is one thing to be able to write the recursive solution like this, but it is another one to actually carry it through. Analytical solutions are typically not available and one has to rely on Monte Carlo approximations, which we discuss later. Before doing so, we quickly review one special case where there is a tractable solution as it is relevant to later developments.

The Kalman filter

In the case of a Gaussian linear system as described in Eqs. (1.5) and (1.6) it is possible to carry out the filtering recursion analytically resulting in the well-known Kalman filter, or Kalman-Bucy filter (Kalman, 1960; Kalman and Bucy, 1961). The filter was originally derived as a sequential least-squares algorithm without making any Gaussian assumptions, but here we show the Bayesian probabilistic derivation.

If one begins with an initial Gaussian distribution π_0, then by properties of the Gaussian distribution the background and analysis distributions remain Gaussian and the filtering recursion consists in updating means and covariance matrices only. Let us assume that we start at time $t-1$ with the following analysis distribution $\pi^a_{t-1}(x_{t-1}|y_{1:t-1}) = \mathcal{N}(\mu^a_{t-1}, P^a_{t-1})$, where the subscript a stands for analysis.

The forecast step consists in propagating this distribution through the system with Eq. (1.5). Because Gaussian distributions remain Gaussian under linear transformations, we can derive the background distribution $\pi^b_t(x_t|y_{1:t-1})$ to be $\mathcal{N}(\mu^b_t, P^b_t)$, with moments defined as

$$\mu^b_t = F_t\, \mu^a_{t-1}, \tag{1.10}$$

$$P^b_t = F_t\, P^b_t\, F'_t + Q_t. \tag{1.11}$$

The analysis step consists in applying Bayes' formula to this prior distribution using the Gaussian likelihood of the observations. Because the Gaussian distribution is conjugate to itself, the posterior π^a_t is again Gaussian $\mathcal{N}(\mu^a_t, P^a_t)$ with new, updated moments

$$\mu^a_t = \mu^b_t + K_t(y_t - H_t\, \mu^b_t), \tag{1.12}$$

$$P^a_t = (I - K_t(P^b_t)H_t)P^b_t, \tag{1.13}$$

where $K_t(P)$ is the Kalman gain computed with covariance matrix P as:

$$K_t(P) = PH'_t(H_tPH'_t + R_t)^{-1}. \tag{1.14}$$

Kalman filtering consists then simply in updating the moments of the filtering distribution by alternating between the forecast step using Eqs. (1.10) and (1.11), and the analysis step using Eqs. (1.12) and (1.13). See for example Wikle and Berliner (2007) or Bishop and Welch (2001) for more detail on the derivation of the filter.

1.3 Ensemble filtering methods

Ensemble, or Monte Carlo, filtering methods represent the background and analysis distributions as finite samples of k particles, $\{x^{b,i}\}$ and $\{x^{a,i}\}$ respectively. The forecast step consists then simply in applying the dynamical equation of the system to each particle individually, e.g. $x^{b,i}_t =$

$f(x_{t-1}^{b,i}, \psi_t^i)$ in the case of Eq. (1.1). The Monte Carlo version of the analysis, on the other hand, is less straightforward to implement and depends on the assumptions that are made about π_t^b and ℓ_t, and on the sampling algorithm used. In the remainder of this section we present three different ensemble filtering methods which vary in the way they implement the analysis. For better legibility we drop the time subscript. Material from this section was published in Section 2 of Robert and Künsch (2017b).

1.3.1 The particle filter

Particle Filters (PFs) directly implement Bayes' formula in the analysis without relying on any additional assumptions about π^b (Gordon et al., 1993; Pitt and Shephard, 1999; Doucet et al., 2001). In its simplest version, the PF represents distributions by weighted samples of *particles*. The analysis consists then in updating these weights by a factor proportional to the likelihood. More precisely, if π^b is represented by the weighted sample of size k $\{x^{b,i}, \alpha^{b,i}\}$, then π^a is represented by the weighted sample $\{x^{a,i}, \alpha^{a,i}\}$, where

$$x^{a,i} = x^{b,i}, \quad \alpha^{a,i} = \frac{\alpha^{b,i} \cdot \ell(x^{b,i}|y)}{\sum_{j=1}^{k} \alpha^{b,j} \cdot \ell(x^{b,j}|y)}. \tag{1.15}$$

The problem with this approach is that, when iterated, the weights tend to be more and more concentrated on a few particles, leading to sample degeneracy. In the worst case the whole distribution could be represented with only one particle, which is clearly not desirable. To avoid this effect, a *resampling* step is introduced: instead of keeping the weighted particles, we produce a non-weighted sample by resampling them with probabilities proportional to their weights. This means that the analysis sample contains $x^{b,i}$ N_i times, where $E(N_i) = k\alpha^{a,i}$ and $\sum N_i = k$. By this process, the particles which fit the data well are replicated, while the others are eliminated. Thus, the PF adaptively explores the filtering distribution by putting more mass in regions of high probability. It should be noted that in the case of a deterministic dynamical system resampling does not help and one has to add some artificial noise to the particles (see for example Liu and West (2001)).

It is useful for future comparison to think of the PF in the following way. The background distribution π^b is taken to be the empirical distribution

of $\{x^{b,i}\}$, which can be written as a mixture of point masses

$$\pi^b_{PF}(x) \propto \sum_{i=1}^{k} \mathbf{1}_{(x=x^{b,i})},$$

where $\mathbf{1}$ is the indicator function. Applying Bayes' theorem to this π^b we find the analysis distribution to be another mixture of point masses

$$\pi^a_{PF}(x) \propto \sum_{i=1}^{k} \alpha^i \cdot \mathbf{1}_{(x=x^{b,i})},$$

where the weights α^i are proportional to $\ell(x^{b,i}|y)$. The PF algorithm described above consists then simply in sampling from this mixture distribution. Using a vector of resampled indices I, such that $P(I(i) = j) \propto \alpha^j$ and $|\{I(i) = j; i = 1, \ldots, k\}| = N_j$ for all j, we can write the algorithm succinctly as

1. Compute the weights $\alpha^j \propto l(x^{b,j}|y)$.
2. Choose the vector of resampled indices I, such that $P(I(i) = j) \propto \alpha^j$.
3. For $i = 1, \ldots, k$, set $x^{a,i} = x^{b,I(i)}$.

How we choose the vector of resampled indices I plays a significant role. Indeed, one could simply choose each $I(i)$ by sampling with replacement from the set of indices with probabilities $\{\alpha^i\}$, but this would lead to unnecessary Monte Carlo error. Instead, we prefer to use balanced sampling schemes, which guarantee that $|N_i - k\alpha^i| < 1$ (see Section 4.2.1 for more detail).

1.3.2 The ensemble Kalman filter

The EnKF is a Monte Carlo approximation of the Kalman filter, in which the background distribution is assumed to be normal and the observations to be linear and Gaussian (Evensen, 1994, 2009). In the stochastic version of the EnKF, the analysis particles are produced as

$$x^{a,i} = x^{b,i} + K(P^b)(y - Hx^{b,i} + \epsilon^i), \tag{1.16}$$

where

$$\epsilon^i \sim \mathcal{N}(0, R),$$

and P^b is an estimate of the background covariance. Essentially, it consists in applying the Kalman update of Eq. (1.12) to each particle individually,

and adding stochastic perturbations ϵ^i to the observations. Why does this heuristic algorithm make sense?

Assuming that the background ensemble is normally distributed – i.e. $x^{b,i} \sim \mathcal{N}(\bar{x}^b, P^b)$, where $\bar{x}^b$ is the background ensemble mean, and P^b an estimate of the background covariance – the analysis distribution is again Gaussian with moments μ^a and P^a derived as in Eqs. (1.12) and (1.13). One can easily check that $\bar{x}^a$, the mean of the analysis ensemble defined in Eq. (1.16), is equal to μ^a. To show that $Cov(x^{a,i})$, the covariance of the analysis ensemble, is also equal to P^a, one needs to notice that Eq. (1.13) can be rewritten as:

$$P^a = (I - K(P^b)H)P^b(I - K(P^b)H)' + K(P^b)\,R\,K(P^b)'.$$

Therefore, the EnKF can be seen as a computationally efficient algorithm to sample from $\mathcal{N}(\mu^a, P^a)$, the analysis distribution when assuming that $\pi^b = \mathcal{N}(\bar{x}^b, P^b)$.

The stochastic perturbations ϵ^i are necessary to ensure that the analysis ensemble has the correct covariance: $Cov(x^{a,i}) = P^a$. One could argue, however, that this adds unnecessary Monte Carlo error, especially when working with small ensemble sizes. Deterministic sampling schemes have thus been developed, leading to so-called square-root filters (Whitaker and Hamill, 2002; Tippett et al., 2003; Ott et al., 2004). In these versions of the EnKF, one defines the analysis mean as $\bar{x}^a = \bar{x}^b + K(P^b)(y - H\bar{x}^b)$. Then the deviations of each particle from the mean, $x^{a,i} - \bar{x}^a$, are chosen jointly such that the empirical covariance of $\{x^{a,i}\}$ is exactly equal to P^a, whereas for the stochastic EnKF this relation holds only on expectation over the ϵ^i. Producing such a sample is not uniquely defined, and there are various implementations of such square-root filters. In the present work we will be interested in one such square-root filter, the Ensemble Transform Kalman Filter (ETKF) of Bishop et al. (2001).

1.3.3 The ensemble Kalman particle filter

The Ensemble Kalman Particle Filter (EnKPF) of Frei and Künsch (2013) is a hybrid algorithm that combines the EnKF and the PF with a single parameter $\gamma \in [0, 1]$ controlling the balance between both. Its core idea is to split the analysis into two stages as $\pi^a(x) \propto \pi^b(x) \cdot \ell(x|y)^\gamma \cdot \ell(x|y)^{1-\gamma}$, following the progressive correction principle of Musso et al. (2001). In a nutshell, the algorithm consists in "pulling" the ensemble members towards

the observations with a partial EnKF analysis using the dampened likelihood $\ell(x|y)^\gamma$, and then in applying a partial PF with the remaining part of the likelihood, $\ell(x|y)^{1-\gamma}$. In this way the algorithm can capture non-Gaussian features of the distribution (by resampling), while maintaining sample diversity.

Assuming linear and Gaussian observations, dampening the likelihood with the exponent γ is equivalent to inflating the error covariance R by the factor γ^{-1}. It is easily seen that this is also equivalent to using the Kalman gain with the original error covariance R and a dampened background covariance γP^b.

A key idea for the EnKPF is that the analysis ensemble of the stochastic EnKF produced by Eq. (1.16) can be seen as a balanced sample of size k from the Gaussian mixture

$$\pi^a_{EnKF} = \sum_{i=1}^{k} \frac{1}{k} \mathcal{N}\left(x^{b,i} + K(P^b)(y - Hx^{b,i}), K(P^b)RK(P^b)'\right), \quad (1.17)$$

where balanced sample means that each component of the mixture is selected exactly once. Applying this representation to the EnKPF, we can see that the partial analysis distribution after the EnKF step, $\pi^\gamma(x) \propto \pi^b(x) \cdot \ell(x|y)^\gamma$, is the following Gaussian mixture:

$$\pi^\gamma = \sum_{i=1}^{k} \frac{1}{k} \mathcal{N}(\nu^{a,i}, Q), \quad (1.18)$$

where

$$\nu^{a,i} = x^{b,i} + K(\gamma P^b)(y - Hx^{b,i}) \quad \text{and} \quad (1.19)$$

$$Q = \frac{1}{\gamma} K(\gamma P^b)RK(\gamma P^b)'. \quad (1.20)$$

A naive implementation of the EnKPF would be to sample from this distribution with a usual stochastic EnKF, i.e. by setting $x^{\gamma,i} = \nu^{a,i} + \epsilon^{\gamma,i}$, where $\epsilon^{\gamma,i} \sim \mathcal{N}(0, Q)$, and then to apply a PF update to this ensemble, using as likelihood $\ell(x^{\gamma,i}|y)^{1-\gamma}$. However, this approach would not be optimal as it could lead to ties, or exactly identical particles, in the analysis ensemble. Instead, one can conduct the second step of the analysis, $\pi^a(x) \propto \pi^\gamma(x)\ell(x|y)^{1-\gamma}$, analytically. Indeed, as shown in Alspach and Sorenson (1972), applying Bayes' formula to a Gaussian mixture with a

linear and Gaussian likelihood results in a new Gaussian mixture with component means $\mu^{a,i}$, component covariance $P^{a,\gamma}$, and component weights $\alpha^{\gamma,i}$:

$$\pi^a_{EnKPF} = \sum_{i=1}^{k} \alpha^{\gamma,i} \mathcal{N}(\mu^{a,i}, P^{a,\gamma}), \tag{1.21}$$

where

$$\mu^{a,i} = \nu^{a,i} + K((1-\gamma)Q)(y - H\nu^{a,i}), \tag{1.22}$$

$$P^{a,\gamma} = \big(I - K((1-\gamma)Q)H\big)Q \quad \text{and} \tag{1.23}$$

$$\alpha^{\gamma,i} \propto \phi\{y; H\nu^{a,i}, HQH' + R/(1-\gamma)\}. \tag{1.24}$$

It is useful to rewrite the equation for the $\mu^{a,i}$ components directly from the background ensemble as:

$$\mu^{a,i} = x^{b,i} + L^{\gamma}(y - Hx^{b,i}), \quad \text{where} \tag{1.25}$$

$$L^{\gamma} = K(\gamma P^b) + K((1-\gamma)Q)\Big(I - HK(\gamma P^b)\Big),$$

where L^{γ} is a sort of composite Kalman gain, resulting from the successive application of the EnKF and PF. However, it should be noted that to the best of our knowledge there is no estimate of P^b such that $L^{\gamma} = K(P^b)$.

The final analysis ensemble is obtained as a sample from the Gaussian mixture of Eq. (1.21). A schematic illustration of the EnKPF can be seen in Fig. 1.2. A short description of the algorithm is given as follows:

1. Compute all the $\mu^{a,j}$.
2. Compute all the weights α^j.
3. Choose the resampled indices I such that $P(I(i) = j) \propto \alpha^j$.
4. For $i = 1, \ldots, k$:
   (a) Generate $\epsilon^{a,i} \sim \mathcal{N}(0, P^{a,\gamma})$.
   (b) Set $x^{a,i} = \mu^{a,I(i)} + \epsilon^{a,i}$.

The step (4a) can be done efficiently, without computing $P^{a,\gamma}$ explicitly. To do so, generate independently $\epsilon^{1,i} \sim \mathcal{N}(0, \frac{1}{\gamma}R)$ and $\epsilon^{2,i} \sim \mathcal{N}(0, \frac{1}{1-\gamma}R)$ and define

$$\epsilon^{a,i} = K((1-\gamma)Q)\Big(I - HK(\gamma P^b)\Big)\,\epsilon^{1,i} + K((1-\gamma)Q)\,\epsilon^{2,i}.$$

In the extreme case of $\gamma = 0$, the EnKPF is equivalent to a pure PF, whereas for $\gamma = 1$ it is equivalent to the stochastic EnKF. γ is therefore

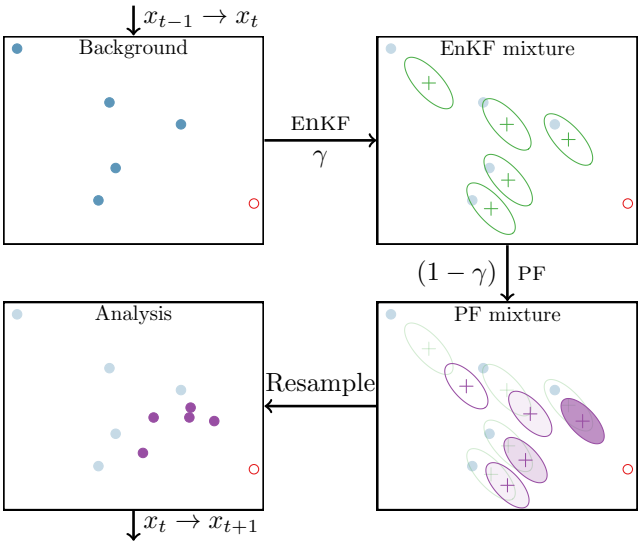

Figure 1.2: Schematic illustration of the EnKPF. Upper left: Background ensemble (full blue dots) and observation (empty red dot). Upper right: Intermediate analysis distribution $\pi^\gamma(x)$ of Eq. (1.18). Each ellipse covers 50% of one component in the mixture. Lower right: Final analysis distribution Eq. (1.21). Ellipses again represent 50% of each component, and the color intensity represents the weights α^i. Lower left: Analysis sample obtained by drawing from Eq. (1.21). The mixture component closest to the observation has been resampled 3 times, while the two components farthest away have been discarded.

a tuning parameter which determines the proportion of EnKF and PF update to use. In Frei and Künsch (2013) it is proposed to choose γ adaptively such that the Equivalent Sample Size (ESS) (Liu, 1996) of the analysis distribution, defined as $1/(k\sum(\alpha^{\gamma,i})^2)$ (where the division by the ensemble size k ensures that the ESS is between 0 and 1), is within a given bound, for example between 50% and 80%. In Section 7.3 we discuss in more detail the optimal choice of γ in practical applications.

1.4 High-dimensional filtering problems

Filtering in geophysical applications such as reservoir modeling and NWP is particularly challenging because of the extreme high-dimensionality of the problem. Typically, the state x comes from the discretization of con-

tinuous fields on a 3-dimensional grid – for example the temperature or the relative humidity in the atmosphere. With a horizontal grid of 1000 by 1000 elements and 30 vertical levels, which is common in large-scale weather forecast, one obtains a 10^7-dimensional state. Multiply this by 12 prognostic variables (in NWP those are typically temperature, pressure, three wind speed components, turbulent kinetic energy, and specific content of water vapor, cloud water, cloud ice, rain, snow and graupel) and the dimension of the state is of the order of 10^8. Furthermore, for large-scale NWP the number of observations assimilated at one time step is of the order of 10^6. The tragedy, however, is that due to the high computational cost of running the forecast model, the ensemble size that one can afford at operational centers is usually less than one hundred. Any sane theoretician would run away from such a problem, but practitioners have tried, and often, surprisingly, succeeded.

1.4.1 Breakdown of the PF in high dimensions

Snyder et al. (2008) explore the limitation of PFs in high-dimensional applications. Their analysis is based on results of Bengtsson et al. (2008); Bickel et al. (2008) that the size of an ensemble should, in some sense, grow exponentially with the dimensionality of the problem. More precisely, they show that, asymptotically, in the case of i.i.d. observations, when the ensemble size k and the number of observations d are large, then the largest weight of the PF (the largest α^i in Eq. (1.15)) behaves as

$$\alpha^{(k)} \approx \frac{1}{1 + \tau^{-1}\sqrt{2 \log k}},$$

where τ^2 is the variance of the observation log-likelihood at the background ensemble, which depends on d (typically $\tau = O(\sqrt{d})$). One can see that $\alpha^{(k)} \to 1$, and thus that the ensemble collapses, unless k is exponentially large compared to τ^2.

Somehow contrary to intuition, the ensemble size requirement does not depend directly on the state space dimension, but rather on the quantity τ^2, which is related in an indirect manner to dimensionality. One can think about τ^2 as an effective dimension, to which it is equivalent in the case of a Gaussian prior with identity covariance matrix, identity observation operator H and identity observation covariance R.

One way to reduce the variance of the weights of the PF is to use a proposal distribution different from $\pi^f(x_t|x_{t-1})$, as in the auxiliary PF (Pitt and

Shephard, 1999), or in geosciences van Leeuwen (2010). Ignoring other issues concerning the applicability of such types of algorithm to NWP, Snyder et al. (2015) show that with a PF using an optimal proposal distribution – meaning that the proposal is the distribution of x_t conditioned on x_{t-1} and y_t – collapse of the weights still occurs if the ensemble size does not grow exponentially. Therefore the only hope to apply PFs to high-dimensional filtering is to reduce the dimensionality of the system, which we will discuss later in Section 1.5.

1.4.2 Spurious correlations with the EnKF

The EnKF does not have a resampling step and thus does not suffer directly from sample depletion or ensemble collapse as the PF does. However, the EnKF analysis depends entirely on the estimated background covariance P^b, and it is well-known that the sample covariance matrix is not a good estimator when the number of observations is smaller than the number of variables, in our case when the ensemble size is smaller than the system dimension (Bickel and Levina, 2008). With a poor estimator of P^b, the Kalman gain is also poorly estimated, and with it the EnKF analysis.

This problem manifests itself in practice with the appearance of spurious correlations: non-zero estimated correlations between components of x that should be intuitively uncorrelated. In particular, one observes long-range correlations between locations, which results in the observations having an influence far away, reminiscent of *spooky action at distance*. While it might be possible that there are such so-called teleconnections in geophysical systems, they are difficult to estimate and with a small ensemble size there will be mostly false positives. This problem has been recognized early on in applications, and in practice regularized estimators of the covariance matrix are used (Hamill et al., 2001), which are an example of localization techniques, as discussed in the next section.

1.5 Localization

The extremely large dimensionality of the state space in geophysical applications is in some sense artificial, because it comes from the discretization of 3-dimensional continuous fields. As one increases the resolution of the grid, the number of dimensions increases, but the dimensionality of the attractor of the dynamical system stays the same. To make an analogy

with statistics, the components of x are strongly correlated and thus the actual number of degrees of freedom is much smaller than the number of dimensions of x. Ideally, one would find a low-dimensional basis for the system and thus overcome the problem of high-dimensional filtering discussed above. However, in geophysical applications such a basis is not available and we have to work with the discretization of the 3-dimensional fields.

A natural assumption to make about a state space with a spatial ordering is that components of x that are far apart are independent. In data assimilation applications such assumptions about the system translate into assumptions about the background distribution π^b. Localization is a set of techniques that try to take advantage of these assumptions to overcome the problem of high-dimensionality in filtering.

1.5.1 Localizing the EnKF

If π^b is assumed to be Gaussian, as in the EnKF, the assumption that far away locations are independent is equivalent to saying that elements of the covariance matrix P^b corresponding to far away locations are zero. One way to impose such sparsity on the covariance matrix is to use tapering (Furrer et al., 2006). A tapered estimate of P^b is of the form

$$P^b_C = C \circ P^b,$$

where C is a valid correlation matrix and $\circ$ denotes the Shur product (element-wise multiplication). Expressing knowledge about independence through the tapering matrix C is an art in itself. Typically one uses a correlation function decreasing with the Euclidean distance between sites and parametrized in a convenient way, as described in Gaspari and Cohn (1999).

Using such regularized estimators of P^b with the EnKF is common in practice (Hamill et al., 2001; Houtekamer and Mitchell, 2005). The advantages of this technique are twofold: first it eliminates spurious correlations at distances, and second it makes the computation more efficient by exploiting the sparsity of the covariance matrix. However, it should be noted that imposing sparsity on the covariance P^b does not imply sparsity of the Kalman gain – because of the inverse in Eq. (1.14) – and observations can have a long-range influence on the analysis.

Another approach to localization is to compute the analysis separately at

each site using only nearby observations (for example in Ott et al. (2004) or Hunt et al. (2007)). The global analysis is then constructed by *gluing* together the local analyses. Such a local update scheme guarantees that the observations cannot have a long-range influence, and means actually that sparsity is imposed on the Kalman gain. The gluing together of local analyses works only if the observations assimilated at nearby locations are similar. A basic step to fulfill this requirement is to tune the localization radius such that the intersection of the sets of observations assimilated in nearby windows is large. To further enforce smooth transitions, one can give decreasing weights to observations as they are further and further away. Such an effect is achieved by tapering the inverse of R, the observation covariance matrix, with a correlation function dependent on the distance of the observation from the site at which the local assimilation is conducted.

Local update schemes are attractive in practice, as they can be implemented in parallel and thus are very efficient. However, they are not well founded from a theoretical point of view and it is difficult to justify their use except for practical reasons. Sakov and Bertino (2011) explore in more detail the differences between local update schemes and covariance tapering, and show that they are generally not equivalent. However, in practice the differences are small and the computational gains of the local update scheme are such that they outweigh its conceptual deficiencies.

1.5.2 Localizing the PF

Given the success of localization for EnKF algorithms, it is tempting to try the same approach with the PF. However, there are some complications due to the resampling in the analysis, which is discrete by nature. First let us consider a simple case where localizing the PF would work. Imagine that x is normally distributed with identity covariance matrix and that observations are of each component of x with a diagonal error covariance matrix. This example is the one discussed in Snyder et al. (2008) to illustrate the collapse of the PF in high dimensions. In this case localizing the PF is trivial: the background distribution factorizes as a product

$$\pi^b(x_1, x_2, \dots) = \pi^b(x_1) \cdot \pi^b(x_2) \dots,$$

where x_i is the ith component of x. The likelihood factorizes as well, because the observations y_i are independent and depend on one location

only, x_i, and thus the analysis distribution factorizes too:

$$\pi^a(x_1, x_2, \dots) \propto \ell(x_1|y_1)\pi^b(x_1) \cdot \ell(x_2|y_2)\pi^b(x_2) \cdots = \pi^a(x_1) \cdot \pi^a(x_2) \dots$$

To produce a sample from this analysis distribution one can thus sample each component x_i from its local analysis distribution $\pi^a(x_i|y_i)$, leading to a fully local PF. Rebeschini and Handel (2015) discuss the theory of such a local PF generalized to the case of blocks of independent locations, and show that it could overcome the curse of dimensionality. However, there are some constraints when trying to apply this idea in practice.

Why can we not we generalize the assumption of a sparse covariance matrix made in the local EnKF to independence relations between far away locations? We can in fact do it, but the problem is that while it makes sense to assume that two remote locations, say x_1 and x_3, are independent, it does not make sense to assume that they are independent given x_2, which lies between them. In the EnKF case, the same phenomenon explains that using a sparse covariance does not result in a sparse Kalman gain. However, with PFs it is not clear how to overcome this problem. Indeed, if one ignores this issue, it leads to the creation of artificial discontinuities in the global analysis. Imagine that to produce the analysis particle 1, $x^{a,1}$, at location 1 particle $x_1^{b,j}$ was selected for resampling, and that at location 2 particle $x_2^{b,i}$ was selected, where $i \neq j$. When gluing together both local analyses to create the global analysis $x^{a,1} = (x_1^{a,1}, x_2^{a,1})' = (x_1^{b,j}, x_2^{b,i})'$, we are combining two different particles. There is no guarantee that this $x^{a,1}$ represents a physically possible particle, as its components were not sampled jointly from the background π^b.

On the one hand, localizing the PF seems to be the only viable solution to avoid the curse of dimensionality (Snyder et al., 2015; Rebeschini and Handel, 2015), but on the other hand, there are some fundamental difficulties due to the resampling nature of the algorithm. There are at least two possible approaches to this conundrum: either one tries to avoid resampling altogether, or one tries to smooth out the discontinuities introduced by the local resampling of particles. For example, the moment-matching filter of Lei and Bickel (2011) depends on the mean and covariance of the background distribution only and can thus avoid resampling. Another possibility is to use deterministic transport maps instead of resampling, as for example in Reich (2013b). In this case, one samples from the posterior by finding a transport map from the background π^b to the analysis π^a such that the particles have to be moved as little as possible. Such transport maps computed locally vary smoothly in space, and thus the resulting local

analyses can be glued together without leading to strong discontinuities. An example of an algorithm which tries to smooth out the discontinuities between locally resampled particles is the LLEnsF of (Bengtsson et al., 2003). Another attempt at localizing the PF is the local PF of Poterjoy (2016), where resampling is applied locally, but then the particles are progressively merged with prior particles followed by a higher-order correction using a deterministic probability mapping.

Localizing hybrid algorithms such as the EnKPF is a bit easier than localizing the pure PF, but the fundamental difficulty due to the local resampling of particles remains to be addressed. In Chapter 2, we present two localized algorithms based on the EnKPF: the naive-LEnKPF, which is a direct implementation of local update to the EnKPF, and the block-LEnKPF, which takes the approach of smoothing out the introduced discontinuities by relying on Gaussian assumptions when unavoidable. In Chapter 3, we investigate in detail the trade-offs involved in localizing PF algorithms. In Section 4.2, we discuss some additional ideas to localize the PF, introducing dependencies between sites during the resampling step.

1.6 Ensemble space algorithms

Depending on the relative value of k, the ensemble size, d, the number of observations, and q, the state dimension, different schemes to compute the analysis are preferable (Tippett et al., 2003). Here we focus on the case where $k \ll min(d, q)$, which is typical for geophysical applications. It should be noted that when localizing the update, though, the number of observations can be greatly reduced, but the computations we describe here are still valid, if not optimal. In the case of $k \ll min(d, q)$, our ensemble is of very low rank, and instead of looking at it as k points in a $d-$ or $q-$dimensional space, one can instead work directly in the basis formed by the ensemble. Applying this idea to the EnKF leads to interesting views on the inner working of the algorithm and to efficient implementations of the computation (see in particular the ETKF in (Bishop et al., 2001; Hunt et al., 2007))

Let us call the *ensemble space* the $(k-1)$-hyperplane spanned by the background ensemble in the q-dimensional state space, where $k \ll q$. Requiring that the analysis stays in this subspace brings great advantages in terms of computation and stability of the assimilation scheme. No new direction of perturbations can be added to the ensemble, which is both a blessing

and a curse: on the one hand, it avoids that completely unphysical artefacts are introduced in the analysis, but, on the other hand, it requires that the main directions of uncertainty are represented by the background ensemble.

Let us write the background and analysis ensembles as $q \times k$ matrices $\boldsymbol{x}^b$ and $\boldsymbol{x}^a$ respectively, with each column being a particle. Additionally, we define the matrix of ensemble perturbations $X^b = \boldsymbol{x}^b - \bar{x}^b \mathbf{1}'$, where $\mathbf{1}$ denotes the vector of length k with all elements equal to 1. Requiring that the analysis lies in ensemble space means that

$$\boldsymbol{x}^a = \bar{x}^b \mathbf{1}' + X^b W, \tag{1.26}$$

where W is a $k \times k$ matrix. Writing an assimilation algorithm in ensemble space means finding the W matrix of analysis weights. In Chapter 5 we will show that the EnKPF can be formulated in ensemble space under the condition that one estimates P^b with the sample covariance. The resulting algorithm can be implemented efficiently and is thus adapted to large-scale applications. Analogous to the ETKF of Bishop et al. (2001), we also derive a square-root version of the EnKPF: the ETKPF, which requires to solve a so-called continuous algebraic Riccati equation (CARE).

1.7 Models where methods were applied

As mentioned at the beginning of this chapter, the target application of the methods discussed here is data assimilation for NWP. However, working with a full-scale weather model is a complex task: first, it involves implementing all the algorithms efficiently and making them interact with the other components of the weather prediction system, mainly written in fortran90; second, any experiment needs a lot of time due to the high computational cost of the dynamical models.

Therefore, the typical approach to develop new data assimilation algorithms is to work with simple low-dimensional dynamical systems. The hope is that performances obtained on these toy models will be representative of the performances on the target application.

In the present work we go through a hierarchy of model complexity. The feasibility of the EnKPF was shown in Frei and Künsch (2013) with the 40-dimensional Lorenz96 model (Lorenz and Emanuel, 1998). In Chapter 2 we test our new localized algorithms on a 900-dimensional toy model

of cumulus convection based on a modified SWEQ (Würsch and Craig, 2014). Finally, in Chapter 6 we apply the new Local Ensemble Transform Kalman Particle Filter (LETKPF) to COSMO, a full-scale weather prediction system.

1.8 Outline of the remaining chapters

The first part of the remainder of this thesis revolves around the localization of the EnKPF. In Chapter 2, which was published as Robert and Künsch (2017b), we present two new local EnKPFs, the naive-local EnKPF (naive-LEnKPF) and the block-local EnKPF (block-LEnKPF). We apply them to the modified shallow water equation (SWEQ) model of Würsch and Craig (2014) and show that they have performance comparable to the Local Ensemble Kalman Filter (LEnKF). In particular, the new methods are well suited to deal with the rain field, which is highly non-Gaussian.

In Chapter 3, published as Robert and Künsch (2017a), we investigate more closely the power of localization to beat the curse of dimensionality. To do so, we extend an example of Snyder et al. (2008) and conduct a simulation study to compare various global and local algorithms. We also show that the conclusions hold for the Lorenz96 model (Lorenz and Emanuel, 1998).

In Chapter 4, we include some unpublished results related to the localization of the EnKPF. First, we explore a simple 3-dimensional example, which provides a good illustration for the local EnKPFs, and which we use to discuss some variations and generalization of the block-LEnKPF. Second, we include an exploration of alternative ideas to localize the resampling step of the PF directly, in particular we present a local balanced sampling scheme.

The second part of the thesis concerns the formulation of the EnKPF in ensemble space and the derivation of the Ensemble Transform Kalman Particle Filter (ETKPF). In Chapter 5, submitted as the first part of Robert et al. (2017), we present the theory associated with the new algorithm. In Chapter 6, published as the second part of Robert et al. (2017), we show the results obtained with the LETKPF applied to COSMO in an experimental setup similar to the one used operationally at MeteoSwiss.

In Chapter 7 we present additional unpublished results related to the ETKPF. First, we present an intuition for the geometric derivation of

the algorithms in ensemble space. Then we use a visualization technique to illustrate the ETKPF and compare it to the EnKF and ETKF. After that, we discuss more thoroughly the problem of selecting γ adaptively, propose some alternative criteria to the ones in Chapter 5 and conduct a simulation study. The details of the implementation in the COSMO assimilation framework are discussed in the last section.

Finally, Chapter 8 concludes with a summary and general perspectives on possible future developments.

Part I

Localization

Chapter 2

Localizing the Ensemble Kalman Particle Filter

The content of this chapter has been published in Robert and Künsch (2017b). Sections 2.1 and 2.2 repeat some things that have been discussed in the first chapter of this thesis, but have been kept for completeness. In Section 2.3 the two local EnKPFs, the block-LEnKPF and the naive-LEnKPF, are introduced before being tested on an artificial model of cumulus convection in Sections 2.4 and 2.5. Details on the derivation of the block-LEnKPF can be found in the Appendix 2.A. Additional unpublished experimental results are discussed in the Appendix 2.B.

The code associated with this chapter is available as an R package on github: `https://github.com/robertsy/assimilr`. The modified SWEQ model of artifical cumulus convection is implemented in fortran90 from the Python code of Michael Würsch (personal communication). Information about the functions and usage is available as built-in documentation.

2.1 Introduction

In many large-scale environmental applications, estimating the evolution of a geophysical system, such as the atmosphere, is of utmost interest. *Data assimilation* solves this problem iteratively by alternating between a forecasting step and an updating step. In the former, information about

the dynamic of the system is incorporated, while in the latter, also called *analysis*, partial and noisy observations are used to correct the current estimate. The optimal combination of the information from these two steps requires an estimate of their associated uncertainty. In statistics, one represents the uncertainty about the state of a system after the forecasting step with a prior distribution, and the uncertainty due to the observations errors with a likelihood. The analysis consists then in deriving the posterior distribution of the current state of the system, combining the prior distribution and the new observations, which can be done with Bayes' rule.

In geophysical applications, such as NWP, the dimension of the system is extremely large and the forecasting step computationally heavy, therefore the focus is on developing efficient methods with reasonable approximations. Even in the simplest case of a linear system with linear Gaussian observations, the optimal method, namely the Kalman filter (Kalman, 1960; Kalman and Bucy, 1961), is difficult to use because of the size of the matrices involved.

Ensemble, or Monte-Carlo, methods, are elegant techniques to deal with non-linear dynamical systems. They use finite samples, or *ensembles* of *particles*, to represent the uncertainty about the state of the system associated with the prior and posterior distributions. The forecasting step consists then simply in integrating each particle according to the law of the system. Ensemble methods were introduced in the geosciences by the EnKF of Evensen (1994, 2009) as a Monte-Carlo approximation of the Kalman filter and have shown great success in practice. However, the analysis step of all EnKF methods implicitly relies on the assumption that the prior uncertainty about the state of the system is Gaussian, which is an acceptable approximation in some cases, but is unlikely to hold with highly non-linear dynamics.

PFs (Gordon et al., 1993; Pitt and Shephard, 1999; Doucet et al., 2001) are a more general class of ensemble methods which differ from the EnKF in the way the analysis step is implemented. They can handle fully non-linear and non-Gaussian systems and are therefore very attractive. Unfortunately, it has been shown that the number of particles needed to avoid sample degeneracy and collapse of the filter increases exponentially with the size of the problem, in a sense made precise in Snyder et al. (2008). Adapting PFs for large-scale environmental applications is an active field of research and there are many propositions of algorithms (see van Leeuwen (2009) for a review, and van Leeuwen (2010); Papadakis et al. (2010); Ades and van Leeuwen (2013); Nakano (2014) for more recent developments).

Here we focus on the EnKPF, introduced in Frei and Künsch (2013), which consists in a combination of the PF with the EnKF. Compared to other similar algorithms, the EnKPF has the distinct advantage of being dependent on a single tuning parameter which defines a continuous interpolation between the EnKF and the PF. Moreover, no approximation of the prior distribution or the transition probability is required.

Recently, there has been a tendency towards using physical models with higher and higher resolution. For example in NWP, regional models are starting to be run with a grid length of the order of 1 kilometer, which allows to resolve explicitly highly non-linear phenomena such as cumulus convection. In general, with non-linear dynamical systems, the uncertainty after the forecasting step can become highly non-Gaussian. Therefore there is a growing need for data assimilation methods which can handle non-linear and non-Gaussian systems while being computationally efficient to be applied to large-scale problems (Bauer et al., 2015). The EnKF implicitly assumes Gaussian uncertainty while the PF requires an exponentially large number of particles. The EnKPF is a compromise between both, but it still requires too many particles for practical applications. The main goal of this article is to contribute towards a full solution to the non-linear and non-Gaussian large-scale data assimilation problem.

The methods proposed in this article expand on the EnKPF by introducing localization. The idea of localizing the analysis was first proposed by Houtekamer and Mitchell (1998) as a device to reduce dimensions and thus to allow for smaller ensemble sizes. While localization has been widely used within the EnKF family of algorithms, e.g. the Local Ensemble Transform Kalman Filter (LETKF) of Hunt et al. (2007), applications to the PF are much rarer. The reason for this is that PF methods introduce a discrete component in the analysis by resampling particles, which breaks the necessary smoothness of the fields to be estimated.

In the literature, the main approaches to this problem have been either to avoid resampling altogether, or to correct the introduced discontinuities. The moment-matching filter of Lei and Bickel (2011) avoids resampling and can be localized straightforwardly as it depends on the first two moments of the distribution only. Attempts to replace resampling by deterministic transport maps from prior to posterior distributions are a promising way to reformulate PFs such that they can be easily localized (Reich, 2013a). An early example of algorithm which keeps resampling while localizing the analysis is the local-local EnsF (LLEnsF) of Bengtsson et al. (2003), with which our new algorithm share many similarities. In the recent local PF

of Poterjoy (2016) resampling is applied locally by progressively merging resampled particles with prior particles, followed by a higher-order correction using a deterministic probability mapping.

In this article, we propose two new localized algorithms based on the EnKPF: the naive-LEnKPF and the block-LEnKPF. In the naive-LEnKPF, assimilation is done independently at each location, ignoring potential problems associated with discontinuities; in the block-LEnKPF, data are assimilated by blocks, whose influence is limited to a neighborhood, and discontinuities are smoothed out in a transition area by conditional resampling. The first method is easier to implement as it mirrors the behavior of the LETKF, but the second one deals better with the specific problems associated with localized PFs. The localization of the EnKPF or any PF method is highly non-trivial, but it can potentially bring remarkable improvement in terms of their applicability to large-scale applications (Snyder et al., 2015).

The original EnKPF has been shown in Frei and Künsch (2013) to perform well on the Lorenz 96 model (Lorenz and Emanuel, 1998) and other rather simple setups. The extensions that we propose in the present paper should allow the algorithm to work on more complex and larger models. Here, we test the feasibility of our methods with some numerical experiments on an artificial model of cumulus convection based on a modified SWEQ (Würsch and Craig, 2014) and show that we obtain similar or better results than the EnKF.

In Section 2.2 we briefly review ensemble data assimilation and the EnKF, the PF and the EnKPF. Then we discuss localization and explain the two new localized EnKPFs algorithms in Section 2.3. The numerical experiments and the results are discussed in Section 2.4. Conclusions are presented in Section 2.6

2.2 Ensemble data assimilation

Consider the problem of estimating the state of a system at time t, x_t, given a sequence of partial and noisy observations $y_{1:t} = (y_1, \dots, y_t)$. The underlying process (x_t) is unobserved and represents the evolution of the system, described typically by partial differential equations. The observations are assumed to be conditionally independent given the states and are characterized by the likelihood $l(x_t|y_t)$. This problem fits in the frame-

work of general state space models and is generally known as *filtering* in the statistics and engineering community, and as *data assimilation* in the geosciences.

Mathematically, the goal is to compute the conditional distributions of x_t given $y_{1:t}$, called the filtering or *analysis distribution* π_t^a. There exists a recursive algorithm which alternates between computing the analysis distributions and the conditional distributions of x_t given $y_{1:t-1}$, called the *predictive* or *background* distribution π_t^b. In the forecast step, the background distribution at time t is derived from the analysis distribution at time $t-1$, using the dynamical laws of the system. In the *analysis* or *update* step, the analysis distribution at time t is derived from the background distribution at the same time t, using Bayes' theorem: $\pi_t^a(x) \propto \pi_t^b(x) \cdot l(x|y_t)$. However, there is no analytically tractable solution to this recursion except in the case of a discrete state space or a linear and Gaussian system. In the latter case, the solution is known as the Kalman filter.

One of the problems that arise when trying to apply this theoretical framework to large-scale systems such as in NWP is that the forecast step is not given by an explicit equation but comes from the numerical integration of the state vector according to the dynamical laws of the system. *Ensemble* or *Monte Carlo* methods address this problem by representing the distributions π_t^b and π_t^a by finite samples or *ensembles* of *particles*: $x_t^{b,i} \sim \pi_t^b$ and $x_t^{a,i} \sim \pi_t^a$ for $i = 1, \ldots, k$, where k is the size of the ensemble. The forecast step produces the background ensemble members $x_t^{b,i}$ by propagating the analysis ensemble members $x_{t-1}^{a,i}$ according to the dynamics of the system. The analysis step, that is the transformation of the background ensemble $(x_t^{b,i})$ into the analysis ensemble, is however more challenging for ensemble methods. There are various solutions to this problem, depending on the assumptions about the distribution π^b and the observation process and the heuristic approximations that are used.

Henceforth we drop the time index t and consider the analysis step only. We also assume that the observations are Gaussian and linear with mean Hx and covariance R, where H is the observation operator applied on a state vector x and R is a valid covariance matrix. We now review the EnKF and the PF in this context and describe the EnKPF, before discussing in more detail the problem of localization and introducing new algorithms.

2.2.1 The EnKF

If one assumes that the background distribution π^b is Gaussian and that the observations are linear and Gaussian, then the analysis distribution π^a is again Gaussian with a new mean and covariance given by simple formulae. All EnKF methods are based on this result and apply it by ignoring non-Gaussian features of π^b. They use the background ensemble to estimate the mean and covariance of π^b and draw the analysis sample to match the mean and covariance of π^a under Gaussian assumptions. Square-root filters such as the LETKF transform the background ensemble so that the first and second moments match exactly those of the estimated analysis distribution, whereas the stochastic EnKF applies a Kalman filter update with some stochastically perturbed observations to each ensemble member. More precisely, in the stochastic EnKF an ensemble member from the analysis distribution is produced as follows:

$$x^{a,i} = x^{b,i} + K(P^b)(y - Hx^{b,i} + \epsilon^i), \tag{2.1}$$

where P^b is an estimate of the background covariance matrix and $\epsilon^i \sim \mathcal{N}(0, R)$ is a vector of observation perturbations. $K(P)$ denotes the Kalman gain computed using the covariance matrix P and is equal to $PH'(HPH' + R)^{-1}$. Conditional on the background ensemble $(x^{b,i})$, $x^{a,i}$ is thus normal with mean $x^{b,i} + K(P^b)(y - Hx^{b,i})$ and covariance $K(P^b)RK(P^b)'$. A key idea for the EnKPF is that, conditional on the background ensemble, the analysis ensemble is a balanced sample of size k from the Gaussian mixture

$$\sum_{i=1}^{k} \frac{1}{k} \mathcal{N}\big(x^{b,i} + K(P^b)(y - Hx^{b,i}), K(P^b)RK(P^b)'\big), \tag{2.2}$$

where balanced sample means that each component of the mixture is selected exactly once.

2.2.2 The PF

The PF does not make any assumption about π^b, but applies Bayes' formula to the empirical distribution provided by the background ensemble. In its simplest version it represents the background and analysis distributions by weighted samples of *particles*, whose weights are updated at each time step by a factor proportional to the likelihood of the observations.

That is, if π^b is represented by the weighted sample $(x^{b,i}, \alpha^{b,i})$, then π^a is represented by the weighted sample $(x^{a,i}, \alpha^{a,i})$, where

$$x^{a,i} = x^{b,i}, \quad \alpha^{a,i} = \frac{\alpha^{b,i} \cdot l(x^{b,i}|y)}{\sum_{j=1}^{k} \alpha^{b,j} \cdot l(x^{b,j}|y)}.$$

In the forecast step, weights remain unchanged. However, when iterated this leads to sample degeneracy, that is the weights are effectively concentrated on fewer and fewer ensemble members. To avoid this, a resampling step is introduced, where the particles are resampled with probability proportional to their weight. This means that the analysis sample contains $x^{b,i}$ N_i times where $E(N_i) = k\alpha^{a,i}$ and $\sum N_i = k$. In this way particles which fit the data well are replicated and the others eliminated, thus allowing to explore adaptively the filtering distribution by putting more mass in regions of high probability.

There are balanced sampling schemes, where $|N_i - k\alpha^{a,i}| < 1$, which reduce the error due to resampling as much as possible (discussion of balanced sampling can be found in Carpenter et al. (1999); Crisan (2001) or Künsch (2005)). But resampling also has problems with sample depletion if the likelihood values $l(x^{b,i}|y)$ are very unbalanced or the dynamical system is deterministic. In that case one has to add some kind of perturbations to the analysis particles, but it is not clear how to choose the covariance of this noise.

Using a vector of resampled indices I, such that $P(I(i) = j) \propto \alpha^j$ and $\#\{I(i) = j, i = 1, \ldots, k\} = N_j$ for all j, we can write the PF algorithm succinctly as follows:

1. Compute the weights $\alpha^j \propto l(x^{b,j}|y)$.
2. Choose the vector of resampled indices I, such that $P(I(i) = j) \propto \alpha^j$ and $|N_j - k\alpha^j| < 1$.
3. For $i = 1, \ldots, k$, set $x^{a,i} = x^{b,I(i)}$.

2.2.3 The EnKPF

The EnKPF (Frei and Künsch, 2013) is a hybrid algorithm that combines the EnKF and the PF with a single parameter $\gamma \in [0, 1]$ controlling the balance between both. Its core idea is to split the analysis in two stages, following the progressive correction principle of Musso et al. (2001). In

a nutshell, the algorithm consists in "pulling" the ensemble members towards the observations with a partial EnKF analysis using the dampened likelihood $l(x|y)^{\gamma}$, and then applying a partial PF with the remaining part of the likelihood, $l(x|y)^{1-\gamma}$. In this way the algorithm can capture some non-Gaussian features of the distribution (by resampling), while maintaining sample diversity. For any fixed $\gamma > 0$, it does not converge to the true posterior distribution as the number of particles tends to infinity, unless the background distribution is Gaussian. The justification of the EnKPF is rather that, for non-Gaussian background distributions, it reduces the variance of the PF at the expense of a small bias.

We now review the derivation of the algorithm briefly but refer to Frei and Künsch (2013) for more detail. Assuming linear and Gaussian observations, dampening the likelihood with the exponent γ is equivalent to inflating the error covariance R by the factor γ^{-1}, and it is easily seen that this is also equivalent to using the Kalman gain with the original error covariance R and a dampened background covariance γP^b. From the Gaussian mixture representation of the EnKF analysis described in Eq. (2.2), we can see that the first step of the algorithm produces the partial analysis distribution

$$\pi^{\gamma} = \sum_{i=1}^{k} \frac{1}{k} \mathcal{N}(\nu^{a,i}, Q), \tag{2.3}$$

where

$$\nu^{a,i} = x^{b,i} + K(\gamma P^b)(y - Hx^{b,i}), \tag{2.4}$$

$$Q = \frac{1}{\gamma} K(\gamma P^b) R K(\gamma P^b)'. \tag{2.5}$$

For the second step, we have to apply Bayes' formula using π^{γ} as the prior and $l(x|y)^{1-\gamma}$ as the likelihood. This has a closed form solution (Alspach and Sorenson, 1972), namely a Gaussian mixture with new centroids $\mu^{a,i}$, covariance $P^{a,\gamma}$, and unequal weights α^i:

$$\pi^a_{EnKPF} = \sum_{i=1}^{k} \alpha^i \mathcal{N}(\mu^{a,i}, P^{a,\gamma}), \tag{2.6}$$

where

$$\mu^{a,i} = \nu^{a,i} + K((1-\gamma)Q)(y - H\nu^{a,i}), \tag{2.7}$$

$$P^{a,\gamma} = \big(I - K((1-\gamma)Q)H\big)Q, \tag{2.8}$$

$$\alpha^i \propto \phi\{y; H\nu^{a,i}, HQH' + R/(1-\gamma)\}, \tag{2.9}$$

and $\phi\{y; \mu, \Sigma\}$ denotes the density of a Gaussian with mean μ and covariance matrix Σ evaluated at y. One can rewrite the equation for the $\mu^{a,i}$ components directly from the background ensemble as:

$$\mu^{a,i} = x^{b,i} + L^\gamma(y - Hx^{b,i}), \quad \text{where} \tag{2.10}$$

$$L^\gamma = K(\gamma P^b) + K((1-\gamma)Q)\Big(I - HK(\gamma P^b)\Big).$$

The final analysis sample is obtained as a sample from the Gaussian mixture (2.6), which can be done at a computational cost comparable to the EnKF. A short description of the algorithm is given as follows:

1. Compute all the $\mu^{a,j}$ as in Eq. (2.7).
2. Compute all the weights α^j.
3. Choose the vector of resampled indices I, such that $P(I(i) = j) \propto \alpha^j$ and $|N_j - k\alpha^j| < 1$.
4. For $i = 1, \ldots, k$:
   (a) Generate $\epsilon^{a,i} \sim \mathcal{N}(0, P^{a,\gamma})$.
   (b) Set $x^{a,i} = \mu^{a,I(i)} + \epsilon^{a,i}$.

The step Item 4a can be done efficiently, without computing $P^{a,\gamma}$ explicitly, as described in Frei and Künsch (2013). A schematic illustration of the algorithm can be seen in Fig. 2.1.

In the extreme case of $\gamma = 0$, the EnKPF is equivalent to a pure PF, whereas for $\gamma = 1$ it is equivalent to the stochastic EnKF. γ is therefore a tuning parameter which determines the proportion of EnKF and PF update to use. In practice it is chosen adaptively such that the ensemble is as close as possible to the PF solution while conserving enough diversity. Diversity of the mixture weights α^j can be quantified by the ESS (Liu, 1996).

The EnKPF has been shown to work well with the Lorenz 96 models and with other simple examples (Frei and Künsch, 2013). However, because

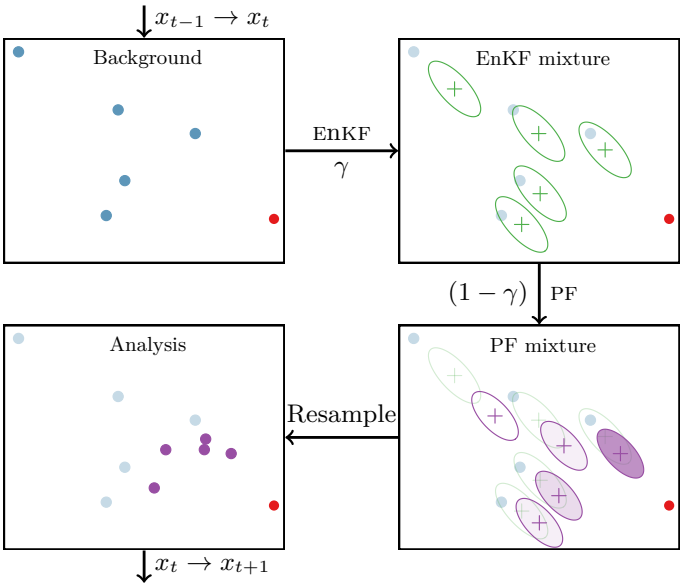

Figure 2.1: Schematic illustration of the EnKPF. Upper left: Background ensemble (blue dots) and observation (red dot). Upper right: Intermediate analysis distribution π^γ (2.3). Each ellipse covers 50% of one component in the mixture. Lower right: Final analysis distribution Eq. (2.6). Ellipses again represent 50% of each component, and the color intensity represents the weights α^i. Lower left: Analysis sample obtained by drawing from Eq. (2.6). The mixture component closest to the observation has been resampled 3 times, while the two components farthest away have been discarded.

it has a PF component, it cannot be directly applied to large-scale systems without suffering from sample degeneracy. In the following section we discuss the technique of localization and introduce two new localized algorithms based on the EnKPF.

2.3 Local algorithms

One of the key element for the success of the EnKF in practice is localization, either by background covariance tapering as in Hamill et al. (2001), or by doing the analysis independently at each grid point, using only nearby observations, as in Ott et al. (2004). Localization suppresses spurious correlations at long distances and generally increases the statistical accuracy

of the estimates by reducing the size of the problem to solve. Its drawback, however, is that it can easily introduce non-physical features in the global analysis fields. For the LEnKF such problems are reduced by ensuring with some means that the analysis varies smoothly in space. It should be noted that physical properties of the global fields cannot be guaranteed in a strong sense without incorporating some explicit constraints.

Without Gaussian assumptions localization becomes even more crucial but also more difficult, as the analysis does not anymore depend on the background mean and covariance only. The collapse of the PF with small ensemble sizes could be avoided by using a very strong localization. However, a pure local PF would probably not be practical as it would introduce arbitrarily large discontinuities in the analysis since different particles can be resampled at neighboring grid points and need to be *glued* together.

Localizing the EnKPF is easier than for a pure PF but it still requires some care due to the resampling step. We propose two different localized algorithms based on the EnKPF: the first one is based on the same principle as the LEnKF of Ott et al. (2004), while the second one is closer to the idea of covariance tapering of Hamill et al. (2001) and serial assimilation of Houtekamer and Mitchell (2001), but adapted to the PF context.

2.3.1 The naive-LEnKPF

In the naive-LEnKPF, we apply the exact same approach as in the LEnKF of Ott et al. (2004) and do an independent analysis at each grid point. We call the resulting algorithm *naive* because it ignores dependencies between grid points. More precisely, the analysis at a given site is produced by sampling from a local analysis distribution which has the same form as Eq. (2.6), but which is computed only using observations close to this site. As in the LEnKF, some level of smoothness is ensured by using the same perturbed observations at every grid point and by choosing a local window large enough such that the observations assimilated do not change too abruptly between neighboring grid points.

In order to mitigate the problem of discontinuities further, we introduce some basic dependency by using a balanced sampling scheme with the same random component for every grid point and by reordering the resampling indices such that the occurrence of such breaks is minimized. This does not remove all discontinuities, but essentially limits them to regions where the resampling weights of the particles change quickly.

In conclusion, the naive-LEnKPF has the advantage to be straightforward to implement, following closely the model of the LEnKF, but it is not completely satisfactory as it introduces potential discontinuities in the global analysis fields. We now consider a second algorithm which is a bit more complicated but avoids this problem.

2.3.2 The block-LEnKPF

In the naive-LEnKPF, localization consists in doing a separate analysis at each grid point, using the observations at nearby locations. We now consider another approach to localization, in which the influence of each observation is limited to state values at nearby locations. This seemingly innocuous change of perspective leads to the development of a new algorithm, the block-LEnKPF. Assuming that R is diagonal or block-diagonal, the observations y can be partitioned into disjoint blocks $y_1, \ldots, y_B$ and then assimilated sequentially, as for example in the EnKF of Houtekamer and Mitchell (2001). The way that localization is implemented for the block-LEnKPF is similar in spirit to the global-to-local adjustment of the LLEnsF of Bengtsson et al. (2003), but the derivation and the resulting algorithms are not identical.

In the case of the EnKF, the influence of one block of observations can be limited to a local area by using a tapered background covariance matrix (Hamill et al., 2001). However, only in the Gaussian case, setting correlations to zero implies independence, but for general π^b this is not true. The PF and EnKPF maintain higher-order dependencies by resampling particles globally, but with a local algorithm some dependencies will necessarily be broken. The block-LEnKPF maintains these dependencies when it is possible, but falls back on a conditional EnKF and implicitly relies on Gaussian assumptions to bridge discontinuities when they are unavoidable. We now describe in more detail how to derive the algorithm for one block of observations and then discuss the general method and parallel assimilation.

Assimilation of one block of observations:

Let us say we partitioned the observations into B blocks and want to assimilate y_1. If we assume that the observation operator is local, then only a few elements of the state vector influence the block y_1 directly (i.e.

have non-zero entry in H for a linear operator). We denote their indices by u with corresponding state vector x_u. Hereafter we use subscripts to denote subvectors or submatrices.

Let us assume also that we use a valid tapering matrix C, for example the one induced by the correlation function given in Gaspari and Cohn (1999). We denote by x_v the subvector of elements that do not influence y_1, but are correlated with some elements of x_u (i.e. correspond to non-zero entries in the tapering matrix C). Additionally, we define as x_w the subvector of all remaining elements.

The principle of the algorithm is to first update x_u with the EnKPF while keeping x_w unchanged. In a second step, x_v is updated conditionally on x_u and x_w, such that potential discontinuities are smoothed out. If x_u and x_w are not only uncorrelated, but also independent, the background distribution can be factored as:

$$\pi^b(x_u, x_v, x_w) = \pi^b(x_u)\pi^b(x_v|x_u, x_w)\pi^b(x_w). \tag{2.11}$$

By construction, only x_u influences y_1 so that one can write $l(y_1|x_u, x_v, x_w) = l(y_1|x_u)$. Applying Bayes' rule, the analysis distribution is

$$\pi^a(x_u, x_v, x_w|y_1) \propto \pi^a(x_u|y_1)\pi^b(x_v|x_u, x_w)\pi^b(x_w).$$

A natural way to sample from this distribution goes as follows: *(i)* sample $x_u^{a,i}$ from the analysis distribution $\pi^a(x_u|y_1)$, *(ii)* keep $x_w^{a,i} = x_w^{b,i}$ unchanged and *(iii)* sample $x_v^{a,i}$ from $\pi^b(x_v|x_u, x_w)$, conditionally on $x_u^{a,i}$ and $x_w^{a,i}$. Steps *(i)* and *(iii)* are clear, but *(ii)* requires more discussion.

One could assume normality and sample $x_v^{a,i}$ as a random draw from a normal distribution with the conditional mean and covariance computed from the background sample moments. However this would add unnecessary randomness and it is more judicious to sample $x_v^{a,i}$ as a correction to the background ensemble member $x_v^{b,i}$, as is done in the EnKF. Using this sampling scheme, we can show that the analysis of $x_v^{b,i}$ conditioned on $x_u^{a,i}$ and $x_w^{a,i}$ is given by the following simple expression:

$$x_v^{a,i} = x_v^{b,i} + P_{vu}^b (P_{uu}^b)^{-1}(x_u^{a,i} - x_u^{b,i}), \tag{2.12}$$

where the matrix inverse is well defined if a tapered estimate of P^b is used, and should be understood as a generalized inverse otherwise.

At first sight it is puzzling that $x_w^{a,i}$ does not appear in the formula, but the correlation between x_v and x_w is present in the background sample $x_v^{b,i}$

and thus does not need to be explicitly taken into account in the analysis. Note that $x_u^{a,i}$ depends on $x_u^{b,I(i)}$.

In cases where $I(i) = i$, the entire particle $x^{a,i}$ is therefore obtained as a correction of the entire particle $x^{b,i}$, according to the original EnKPF algorithm. In cases where $I(i) \neq i$, $x_v^{a,i}$ will depend on two background particles $x_u^{b,I(i)}$ and $x_u^{b,i}$ and the analysis relies on additional Gaussian assumptions of the background sample. Formula (2.12) then makes sure that the correlation between $x_u^{a,i}$ and $x_v^{a,i}$ is nevertheless correct. In order to stay as close as possible to the EnKPF, we permute the resampling indices $I(i)$ such that the number of cases with $I(i) = i$ is maximal. More details about the derivation of the algorithm are provided in Section 2.A.

Putting everything together, the assimilation of one block of observations in the block-LEnKPF algorithm can be summarized as follows:

1. Compute all the $\mu_u^{a,j}$.
2. Compute all the weights α^j.
3. Choose the vector of resampled indices I, such that $P(I(i) = j) \propto \alpha^j$ and $|N_j - k\alpha^j| < 1$.
4. Permute I such that $\#\{j, I(j) = j\}$ is maximal.
5. For $i = 1, \ldots, k$:
   (a) Generate $\epsilon^{a,i} \sim \mathcal{N}(0, P_{uu}^{a,\gamma})$.
   (b) Set $x_u^{a,i} = \mu_u^{a,I(i)} + \epsilon^{a,i}$.
   (c) Set $x_v^{a,i} = x_v^{b,i} + P_{vu}^b (P_{uu}^b)^{-1} (x_u^{a,i} - x_u^{b,i})$.
   (d) Set $x_w^{a,i} = x_w^{b,i}$.

The algorithm is illustrated in Fig. 2.2.

Parallel assimilation of observations:

In the previous paragraph we described how one block of observations is assimilated in the block-LEnKPF. Now let us consider the case of two blocks of observations to be assimilated, say y_1 and y_2. Defining the corresponding state vector indices as above and using an additional subscript for the block, we can see that if $(u_1, v_1) \cap (u_2, v_2) = \emptyset$, then in principle parallel instead of serial assimilation of y_1 and y_2 is possible. In the case where v_1 and v_2 are contiguous, one might worry about discontinuities at

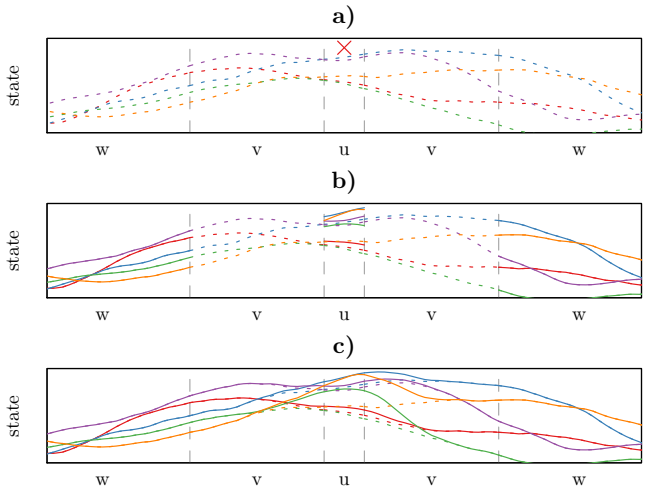

Figure 2.2: Illustration of the assimilation of one observation (red cross in panel a)) with the block-LEnKPF. Each particle is shown in a different color, the dotted lines being the background and the solid lines the analysis. In panel b) x_u is updated while x_w is unchanged. In panel c) we see how the update in x_v makes a transition between x_u and x_w. For the orange and green particles, which are not resampled in x_u, the analysis has to bridge between two different particles by relying on Gaussian assumptions as described in Section 2.A.

the boundary between v_1 and v_2. However, the tapering matrix C ensures that the correlations between sites near this boundary and sites in u_1 and u_2 is small and thus the parallel assimilation of y_1 and y_2 through Eq. (2.12) makes only small changes near this boundary. The procedure could however introduce some discontinuities in higher-order dependence between $x_{v_1}^{a,i}$ and $x_{v_2}^{a,i}$. To avoid this, one could require an additional buffer area between v_1 and v_2, but it would slow down the algorithm and most likely not bring much improvement.

We can therefore assimilate all blocks of observations where the corresponding sets u are well separated in parallel. However, blocks where the corresponding sets u are close have to be assimilated serially. In theory, each assimilation of one block increases the correlation length because the analysis covariance becomes the new background covariance, but we neglect this increase and continue using the same taper matrix C until all observations have been assimilated. This additional approximation is necessary to keep the filter local and it is also used in the serial LEnKF of Houtekamer and Mitchell (2001). The resulting algorithm can be described more precisely as follows:

1. Partition the observations in B blocks $y_1, \ldots, y_B$ and determine the sets u_j and v_j $(j = 1, \ldots, B)$.
2. Choose a block i which has not been assimilated so far or exit if none is left.
3. Assimilate in parallel y_i and all y_j such that $(u_j, v_j) \cap (u_i, v_i) = \emptyset$.
4. Go to step 2

The number of times that the algorithm has to loop between successive updates depends on the specific geometry of the problem and on the partitioning of observations in $y_1, \ldots, y_B$. In general one should try to partition the observations such that as many blocks as possible can be assimilated in parallel, but it is not necessary to find the global optimum to this combinatorial problem.

To recapitulate, the block-LEnKPF consists in assimilating data by blocks and in limiting their influence to a local area. The analysis at sites that do not directly influence the observations in the current block but are correlated with x_u is done by drawing from the conditional background distribution. For cases where resampling does not occur, doing so is equivalent to applying EnKPF in the local window, whereas for cases where it does

occur, the algorithm avoids to introduce harmful discontinuities and produces a smoothed analysis. The block-LEnKPF satisfies all our desiderata for a successful localized algorithm based on the EnKPF. Its disadvantage, however, is that it requires more overhead for the partitioning of observations and its implementation in an operational setting is more complicated.

Now that we introduced two new localized algorithms based on the EnKPF, we will proceed to numerical experiments in order to better understand their properties and test their validity by comparing their performance to the LEnKF.

2.4 Numerical experiments

The algorithms introduced in the present paper can be applied to any task of data assimilation for large-scale systems. However we expect that relaxation of Gaussian assumptions will be most beneficial when the dynamical system is strongly non-linear. Such an application is data assimilation for NWP at convective scale. Würsch and Craig (2014) introduced a simple model of cloud convection which allows one to quickly test and develop new algorithms for data assimilation at convective scale (as for example in Haslehner et al. (2016)). We first briefly introduce the model and the mechanism to generate artificial observations. Then we present results of two cycled data assimilation experiments.

2.4.1 The modified shallow water equation model

The model is based on a modified SWEQ on a one-dimensional domain to generate patterns that are similar to the creation of convective precipitations in the hot months of summer. The convective cells are triggered by plumes of ascending hot air generated at random times and locations. The SWEQ is modified in such a way that if h, the height of the fluid, in the present case humid air, reaches a given threshold (h_c) the convection is reinforced and leads to the creation of a cloud. The convection mechanism is maintained until the fluid reaches a new threshold (h_r), above which the cloud starts to produce rain at a given rate and then slowly disappears. The state of the system can thus be described by three variables: the fluid height h, the rain content r and the horizontal wind speed u. We do not use any units as the scales are arbitrary.

The parameters (fluid height thresholds, precipitation rate, etc.) are the same as for the model run described in (Würsch (2014), Chap. 5) except for the cloud formation threshold set to $H_c = 90.02$ as in (Würsch and Craig, 2014). They have been tuned so that the system exhibits characteristics similar to real convection (fraction of clouds, life-time of precipitation, etc). The random perturbations are introduced at a rate of $8{\cdot}10^{-5}$ $\mathrm{m}^{-1}\,\mathrm{min}^{-1}$. We use a domain size of 150 km with periodic boundary conditions and a resolution of 500 m.

From this system we generate artificial observations that imitate radar measurements. In order to make the experiment realistic, we use a non-linear and non-Gaussian mechanism for generating the observations, but consider them as linear and Gaussian during the assimilation. The rain field is observed at every grid point, but set to zero if below a threshold (r_c), and with some skewed error whose scatter increases with the average amount of rain otherwise. Our observation mechanism is different from the one of (Würsch and Craig, 2014), where simple truncated Gaussian errors were used, and is intended to be more realistic and challenging than the latter. In more detail, the rain observations y_r are generated as follows:

$$y_r = \begin{cases} 0 & \text{if } r \leq r_c \text{ or } \frac{1}{2}\epsilon \leq -\sqrt{r - r_c} \\ (\sqrt{r - r_c} + \frac{1}{2}\epsilon)^2 & \text{otherwise} \end{cases} ,$$

where $\epsilon \sim \mathcal{N}(0, \sigma_r^2)$, independently at every grid point. Such a skewed error distribution for rain observations is a common choice (see for example Sigrist et al. (2012); Stidd (1973)). It consists in applying a Box-Cox transform (with parameter $\lambda = 0.5$), adding some white noise and then transforming back to the original scale. Besides rain, wind speed is also observed with some additive Gaussian noise (with variance σ_u^2), but only at grid points where the observed rain is positive ($y_r \geq r_c$). For the present experiment $\sigma_r^2 = 0.1^2$ and $\sigma_u^2 = 0.0025^2$.

Such artificial observations make data assimilation realistic and challenging due to the non-linearity and sparsity of the observation operator and the non-Gaussian errors. One could consider transforming the observations to obtain a more normal distribution, but we want to test if our algorithms can handle such difficult situations.

A typical example of a field produced with the model and some artificial observations is displayed in Fig. 2.3. The *bumps* in fluid height represent clouds which start to appear if the first threshold is reached (lower dashed line) and are associated with precipitation if they reach the second threshold (upper dashed line), after which they start to decay. Rain can remain

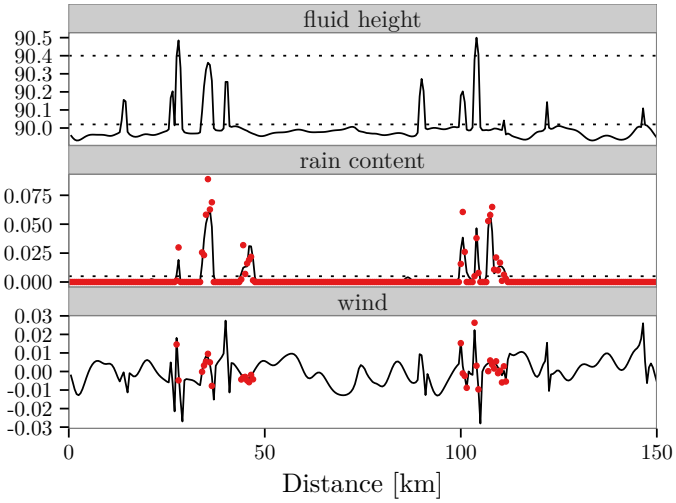

Figure 2.3: Typical example of the modified SWEQ model with artificial radar observations (red dots) and critical values (h_c, h_r and r_c) as dashed lines.

for some time after a cloud has reached its peak. The sharp perturbations in the wind field are the random triggering plumes.

2.4.2 Assimilation setup

An initial ensemble of 50 members is generated by letting the model evolve without assimilating any observations and taking each member at 200 days interval from each other such that they are not correlated. We consider only perfect model experiments and do not take into account model error; in particular we do not use any form of covariance inflation. All the observations are assumed to be Gaussian, with a diagonal covariance matrix R with non-zero elements R_r or R_u, depending on which type of observations it corresponds to. For wind observations the error is the same as for the true generating process, that is we set $R_u = \sigma_u^2$. Rain and no-rain observations are both assimilated and assumed to have the same error. The true rain distribution is non-Gaussian, so R_r is not straightforward to choose, especially because it depends on the rain level. We set $R_r = 0.025^2$ during the assimilation, which is equivalent to the error observed for a rain level of 0.06125 (relatively big, but in the range of observed values). In general more could be done to treat rain observations properly, but it is beyond

the scope of the present study.

We use one localization parameter l set to 5 km. Every method uses a taper of the covariance matrix as defined in Gaspari and Cohn (1999) with half correlation length l. For LEnKF and naive-LEnKPF, the size of the local window is set to l in each direction, for a total of approximately 10 km or 21 grid points. Similarly, the observation blocks for the block-LEnKPF are defined from segments of 10 km in the domain (one block contains all the rain observations falling in a specific 10 km segment and the associated wind observations, if any).

The EnKPF has one free parameter, γ, which controls the balance between EnKF and PF. We choose it adaptively such that the ESS at the resampling step is between 50 and 80% of the ensemble size. A different γ can be selected for each site in the case of the naive-LEnKPF, or for each block of observations for the block-LEnKPF, which allows the method to be closer to the PF in regions where non-Gaussian features are present and to fall back closer to the EnKF when it is necessary. In general the criterion for adaptive γ could be refined and tuned more closely, but it is beyond the scope of the present paper.

The two new local algorithms are compared against the LEnKF of Ott et al. (2004) and not the LETKF of Hunt et al. (2007), because both the EnKPF and the LEnKF are based on the stochastic EnKF and thus are more comparable. Furthermore our results cannot be directly compared to the ones in Haslehner et al. (2016) as our experimental setup is substantially different from theirs.

2.5 Results

In order to highlight some key properties of the new proposed algorithms, we start with an example where high-frequency observations are assimilated and study the resulting analysis ensembles visually. In a second step, repeating this experiment many times, we can evaluate the performance of the algorithms and their differences. In a third step we discuss longer assimilation periods with lower frequency observations. We show the results as figures only as we believe that they are only indicative of some possible advantages but should not be taken too literally as the system under study is very artificial and the results can vary with different choice of parameters. The quality of assimilation is assessed with the Continuous Ranked

Probability Score (CRPS) (Gneiting and Raftery, 2007) commonly defined as:

$$\mathrm{CRPS}(F, x) = \int (F(x') - 1_{(x' \geq x)} dx'),$$

where $F(\cdot)$ is the predictive cumulative probability function, in our case given by the empirical distribution of the ensemble. Because we have a perfect model scenario we can directly evaluate the CRPS of the one-step ahead forecast ensemble compared to the underlying true state of the system. The CRPS is a strictly proper scoring rule, which implies that using such a score allows one to control calibration and sharpness at the same time, contrary to the more commonly used RMSE (see Gneiting and Katzfuss (2014) for a general discussion of probabilistic forecasting).

2.5.1 High-frequency observations

In this first scenario we are interested in seeing if it is possible to use high-frequency radar data, especially for short term prediction. To do so we run a cycled experiment where data are assimilated every 5 min for a total of 1 hour. Starting from an initial ensemble which has no information about the current meteorological situation, the goal of the filter is to quickly capture areas of rain from the observations.

The analysis ensembles of the different algorithms for the rain field show that the local EnKPFs are better able to identify dry areas. In Fig. 2.4 we can see the analysis ensembles after one hour of assimilation in the same typical situation as in Fig. 2.3. All methods recover the zones of heavy precipitation relatively well, with some minor differences in terms of maximum intensity. One should not conclude too much from an isolated case, but there is one interesting trait which is not peculiar to this example and illustrates how the local EnKPF algorithms are able to model non-Gaussian features: the LEnKF maintains some medium level of rain at almost all sites, while both naive-LEnKPF and block-LEnKPF are better at using the *no-rain* observations to suppress spurious precipitation.

Going beyond this particular example, we now consider a simulation study where we repeated the above experiment 1000 times and computed the average CRPS of each algorithm for every assimilation cycle. To make the results more understandable and to remove some natural variability, we always compute the performance relative to a free forecast run. The latter is based on the same initial ensemble used by all algorithms, but does

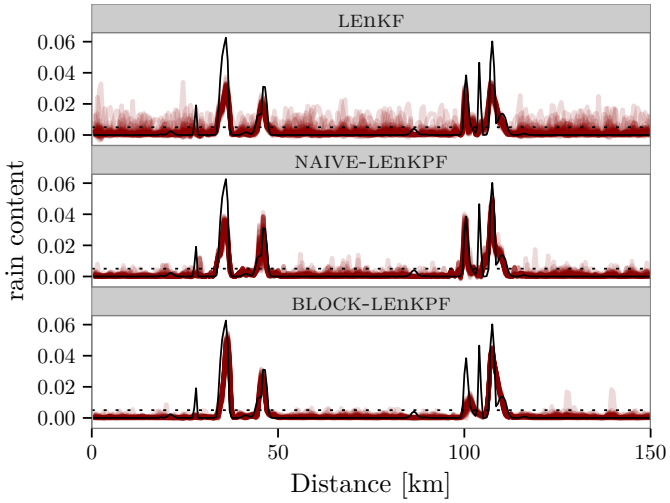

Figure 2.4: Typical example of analysis ensembles for the rain field after 1 hour of high-frequency observations assimilation. Each of the red line is one ensemble member. On top is the LEnKF, followed by the naive-LEnKPF and the block-LEnKPF.

not assimilate any observation, and is thus equivalent to a climatological forecast.

Both new algorithms achieve good performances compared to the LEnKF for the first hour of assimilation, as can be seen in Fig. 2.5. The gains are in terms of rain content, probably because it is the field with the most non-Gaussian features. The block-LEnKPF seems to have a slight advantage over the naive-LEnKPF for the fluid height and the other way round for the rain field, but otherwise their performance is very similar.

Interestingly, if one looks at the evolution of the CRPS for an assimilation period of six hours instead of only one hour in Fig. 2.6, some issues start to become apparent. After an initial drop, the CRPS for the fluid height field increases again and gets worse than the free forecast reference, which means that observations actually hamper the algorithms. The effect is greatest for the fluid height but also slightly visible for the rain for the naive-LEnKPF. Physically, the problem comes from the fact that there is a delay between the formation of a cloud and the appearance of rain. Having assimilated many no-rain observations the algorithm can become overconfident that an area is dry and cannot adapt when new rain starts to appear. The LEnKF is a bit less susceptible to this problem, for the simple

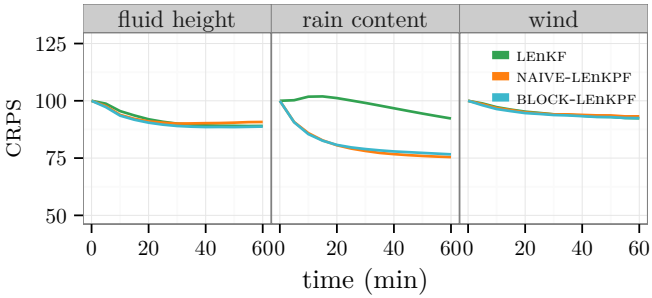

Figure 2.5: Evolution of the CRPS in the first hour with high-frequency observations. The value is given as a percentage relative to a free forecast run. Notice the truncated y-axis.

reason that it is less good at identifying dry areas and thus maintains more spread in the ensemble while the local EnKPFs are too sure that no rain is present. Such an effect can be understood as a form of sample degeneracy coming from the fact that a large number of observations are assimilated, which will be confirmed in the next experimental setup.

To assess the calibration of the algorithms we also look at the rank histograms of all fields in the first hour of assimilation for the block-LEnKPF in Fig. 2.7. The one-step ahead forecast is more or less calibrated, except for a non-negligible fraction of cases where the truth lies outside the range of the ensemble. These problems can be attributed to the inherent difficulty coming from the fundamentally random nature of the system. Indeed, attempts at improving the calibration with covariance inflation and tuning of the R matrix have not been successful. The histograms for the fluid level and the rain content reveal that some newly appeared clouds are missed, while some spurious clouds are sometimes created. The histogram for the absolute wind speed is generally uniform except for a slight underestimation, which comes from the random perturbations of the wind field. Conclusions are similar for other algorithms and no clear differences can be identified. Therefore the improvements in CRPS can be interpreted mainly as better sharpness while keeping calibration the same.

2.5.2 Low-frequency observations

In a second scenario we consider the assimilation of lower frequency observations (every 30 min) but for a longer period (three days). It would be in

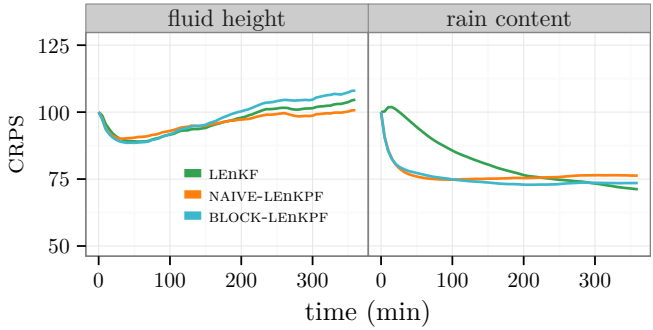

Figure 2.6: Evolution of the CRPS (relative to a free forecast run) for the fluid height and rain fields in the first 6 hours with high-frequency observations. It becomes obvious that after the first initial improvement, all algorithms deteriorate in term of their ability to capture the underlying fluid height. Notice the truncated y-axis.

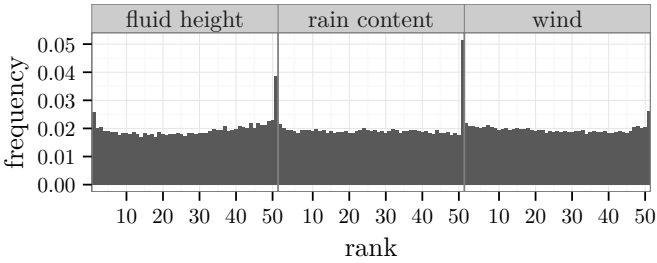

Figure 2.7: Rank histograms computed at one-step ahead forecast for the block-LEnKPF in the high-frequency observations experiment. Only every 10 grid points and every 30 minutes are used to increase independence between observations.

principle possible to run one long cycled experiment and to compute the average performance of the algorithms, but we decided to run 100 repetitions of a three days assimilation period instead, because it can be done in parallel and it is more fault tolerant.

In this scenario the U-shape pattern highlighted in Fig. 2.6 is not present anymore, as more diversity is introduced between each assimilation cycle and the ensemble does not become overconfident. In term of calibration, the results are similar to the high-frequency scenario, but with less tendency to create spurious clouds and rain but a slight bias towards too small clouds. The boxplots of Fig. 2.8 show that block-LEnKPF outperforms the other methods for the rain field, while the naive-LEnKPF shows most difficulties, especially for the fluid height where it gets sometimes worse than the free forecast. One can notice that the fluid height and the wind fields are the most difficult to capture, which comes as no surprise as they are not observed. Furthermore, it should be noted that there is a lot of variability from experiment to experiment, and the CRPS gets regularly worse than the free forecast.

The relatively less good performance of the naive-LEnKPF compared to the block-LEnKPF in this scenario might come from the added discontinuities in the analysis. In the high-frequency scenario the problem was not apparent as the system only evolved for a short time before new observations were assimilated. With low-frequency observations, however, the discontinuities introduced by the naive-LEnKPF have more time to produce a detrimental effect on the dynamical evolution of the system, which results in a poorer performance.

2.6 Summary and discussion

We introduced two new localized algorithms based on the EnKPF in order to address the problem of non-linear and non-Gaussian data assimilation, which is becoming increasingly relevant in large-scale applications with higher resolution. The algorithms that we propose combine the EnKF with the PF in a way that avoids sample degeneracy. We took particular care to localize the analysis without introducing harmful discontinuities, which is an inherent problem of local PFs.

The results of the numerical experiments with a modified SWEQ model confirm that the proposed algorithms are promising candidates for appli-

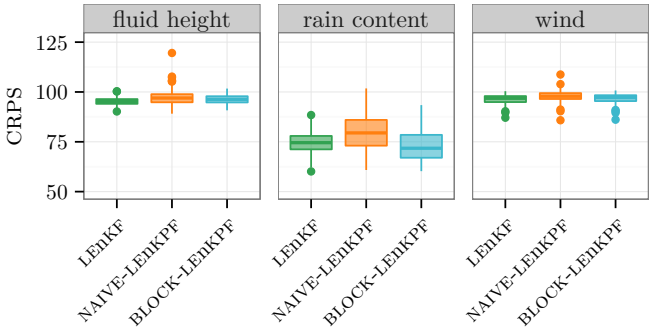

Figure 2.8: Low-frequency observations assimilated for a period of 3 days. Boxplot of the CRPS for the different algorithms and fields considered. The values are given relative to a free forecast. Notice the truncated y-axis.

cation to convective scale data assimilation problems and have some distinct advantages compared to the LEnKF. The two local EnKPFs provide better estimates of the rain field, which has non-Gaussian characteristics and thus benefits greatly from the PF component of the algorithms. This advantage is the strongest either in the high-frequency scenario for short assimilation periods, or in the low-frequency scenario on longer time scales for the block-LEnKPF. Calibration is not perfect as can be seen from the rank histograms, but the improvements in CRPS indicate that the new algorithms are better in terms of sharpness while keeping calibration the same. In general the naive-LEnKPF performs a little worse than the block-LEnKPF, but it is more straightforward to implement and might thus be suitable for large scale applications.

Assimilating high-frequency observations over long periods of time seems to be problematic for all algorithms, as they grow overconfident and are not able to adapt when new clouds appear in the field. Indeed, if one does not assimilate other types of observations the filter is not able to correctly capture the unobserved fluid height field before it produces rain and performances deteriorate quickly after an initial improvement. It is not certain if such a behavior is particular to the present SWEQ model or if it is an inherent characteristic of convective scale assimilation, but it indicates some potential limits of such scenarios. One possible path to tackle this issue would be to properly account for model error. Indeed, the present case study is not a perfect model experiment, as we assume that the observations are linear and Gaussians where in fact they are not. The

typical approach to account for model error is to inflate the covariance, but there is a lot of research to do to understand how to apply such ideas in the context of PF like algorithms.

In many applications, the use of a square-root filter such as the LETKF has been shown to be of great benefit. Therefore, we are currently investigating possibilities to reformulate the EnKPF in this framework. In order to study the impact of localization on the quality of the analysis, we applied the two new localized EnKPFs to some simpler setups than in the present paper, where it is possible to analyze more closely the problem of discontinuities (Robert and Künsch, 2017a). Given the promising results of the localized EnKPFs, we are collaborating with Meteoswiss and Deutscher Wetterdienst on adapting our algorithms to COSMO, a convective scale, non-hydrostatic NWP model.

Appendix 2.A Derivation of the block-LEnKPF

Using the factorization (2.11) and $l(y_1|x_u, x_v, x_w) = l(y_1|x_u)$, Bayes' rule gives the following factorization of the analysis distribution

$$\pi^a(x_u, x_v, x_w|y_1) = \frac{l(y_1|x_u)\pi^b(x_u)\pi^b(x_w)\pi^b(x_v|x_u, x_w)}{\pi^b(y_1)} \tag{2.13}$$

where

$$\pi^b(y_1) = \int l(y_1|x)\pi^b(x)dx = \int l(y_1|x_u)\pi^b(x_u)dx_u.$$

By integrating out x_v, we obtain

$$\begin{aligned}\pi^a(x_u, x_w|y_1) &= \int \pi^a(x_u, x_v, x_w|y_1)dx_v \\ &= \frac{l(y_1|x_u)\pi^b(x_u)}{\pi^b(y_1)}\pi^b(x_w) \cdot 1 \\ &= \pi^a(x_u|y_1)\pi^b(x_w). \end{aligned} \tag{2.14}$$

The last step can be easily checked by integrating out x_w and x_u respectively. The posterior of x_w is nothing else than the prior, which comes as no surprise because x_w is independent from x_u and thus is not influenced by y_1. Also, independence between x_u and x_w continues to hold.

Finally, the conditional posterior of x_v can be derived from the definition of conditional probability

$$\begin{aligned}\pi^a(x_v|x_u, x_w, y_1) &= \frac{\pi^a(x_u, x_v, x_w|y_1)}{\pi^a(x_u, x_w|y_1)} \\ &= \frac{l(y_1|x_u)\pi^b(x_u)\pi^b(x_w)\pi^b(x_v|x_u, x_w)}{l(y_1|x_u)\pi^b(x_u)\pi^b(x_w)} \\ &= \pi^b(x_v|x_u, x_w), \end{aligned} \tag{2.15}$$

where we see that all the terms involving y_1 cancel out. Therefore the conditional posterior of x_v is nothing else than the conditional prior distribution.

Assuming additionally that π^b is normal, the conditional distribution of

x_v is again normal with the following mean and covariance:

$$\mu^b_{v|uw} = \mu^b_v + M_{vu}(x_u - \mu^b_u) + M_{vw}(x_w - \mu_w), \tag{2.16}$$

$$P^b_{v|uw} = P^b_{vv} - M_{vu}P^b_{uv} - M_{vw}P^b_{wv}, \tag{2.17}$$

where

$$M_{vu} = P^b_{vu}(P^b_{uu})^{-1} \text{ and } M_{vw} = P^b_{vw}(P^b_{ww})^{-1}. \tag{2.18}$$

However, instead of making a new random draw from this conditional normal distribution, one can instead devise a method which reuses the background samples. Again under the Gaussian assumption, the residual

$$r^{b,i}_v = x^{b,i}_v - \left(\mu^b_v + M_{vu}(x^{b,i}_u - \mu^b_u) + M_{vw}(x^{b,i}_w - \mu^b_w)\right) \tag{2.19}$$

is independent of $x^{b,i}_u$ and $x^{b,i}_w$ and normally distributed with mean 0 and covariance $P^b_{v|uw}$. Hence we can use this residual for sampling from the conditional distribution $\pi^a(x_v|x^{a,i}_u, x^{a,i}_w, y_1) = \pi^b(x_v|x^{a,i}_u, x^{a,i}_w)$:

$$\begin{aligned} x^{a,i}_v &= \mu^b_v + M_{vu}(x^{a,i}_u - \mu^b_u) + M_{vw}(x^{a,i}_w - \mu^b_w) + r^{b,i}_v \\ &= x^{b,i}_v + M_{vu}(x^{a,i}_u - x^{b,i}_u) \end{aligned} \tag{2.20}$$

because $x^{a,i}_w = x^{b,i}_w$. Plugging in the definition of M_{vu} we obtain the analysis of Eq. (2.12).

In order to understand better how the block-LEnKPF works, let us compare it to the EnKPF analysis of the entire state $x = (x_u, x_v, x_w)$ using the block y_1 and not assuming the factorization Eq. (2.11). Applying the definitions Eq. (2.10) and Eq. (2.8) in the case where P^b and H have the block structure

$$P^b = \begin{pmatrix} P^b_{uu} & P^b_{uv} & 0 \\ P^b_{vu} & P^b_{vv} & P^b_{vw} \\ 0 & P^b_{wv} & P^b_{ww} \end{pmatrix} \tag{2.21}$$

and

$$H = (H_u \;\; 0 \;\; 0) \tag{2.22}$$

it can be easily verified that

$$L^\gamma = (L^{\gamma\prime}_u \;\; (M_{vu}L^\gamma_u)' \;\; 0)' \tag{2.23}$$

and

$$P^{a,\gamma} = \begin{pmatrix} P^{a,\gamma}_{uu} & P^{a,\gamma}_{uu}M'_{vu} & 0 \\ M_{vu}P^{a,\gamma}_{uu} & M_{vu}P^{a\gamma}_{uu}M'_{vu} & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{2.24}$$

Therefore

$$\mu_v^{a,i} = x_v^{b,i} + M_{vu} L_u^{\gamma}(y - H_u x_u^{b,i}) \tag{2.25}$$

$$= x_v^{b,i} + M_{vu}(\mu_u^{a,i} - x_u^{b,i}) \tag{2.26}$$

and

$$\epsilon_v^i = M_{vu}\epsilon_u^i. \tag{2.27}$$

Moreover $\mu_w^{a,i} = x_w^{b,i}$ and $\epsilon_w^i = 0$. Combining these results with Eq. (2.6), we obtain the following for the analysis of $x_v^{a,i}$ with a full EnKPF:

$$x_v^{a,i} = x_v^{b,I(i)} + M_{vu}(x_u^{a,i} - x_u^{b,I(i)}), \quad x_w^{a,i} = x_w^{b,I(i)}. \tag{2.28}$$

Therefore the block-LEnKPF update given by Eq. (2.12) is the same as the EnKPF update given by Eq. (2.28) for those indices where $I(i) = i$. However, for indices i with $I(i) \neq i$, the EnKPF analysis applies a correction to $x_v^{b,I(i)}$ and not to $x_v^{b,i}$ and the size of the correction depends on $x_u^{a,i} - x_u^{b,I(i)}$ and not on $x_u^{a,i} - x_u^{b,i}$. Moreover $x_w^{b,i}$ is replaced by $x_w^{b,I(i)}$ which is in conflict with the requirement of a local analysis. Applying a correction to $x_v^{b,I(i)}$ while setting $x_w^{a,i} = x_w^{b,i}$ would introduce a discontinuity between $x_v^{a,i}$ and $x_w^{a,i}$. Therefore if $I(i) \neq i$, we do not apply an EnKPF analysis to $x_v^{b,i}$, but use instead Eq. (2.12), ensuring a smooth transition between $x_u^{a,i}$ and $x_w^{b,i}$. If $x_u^{b,i}$ and $x_w^{b,i}$ are only uncorrelated, but not independent, we ignore some higher order dependence between these values by pairing $x_u^{a,i}$ with $x_w^{b,i}$ in cases where $I(i) \neq i$, but this seems unavoidable.

Appendix 2.B Additional experimental results

2.B.1 Alternative parameters

As mentioned in Section 2.5, the results obtained with the modified SWEQ model should be taken with a grain of salt as they are sensitive to the choice of parameters. In this section we explore an alternative set of parameters, referred to in the right column of Table 5.1 in Würsch (2014). Our earlier experiments were based on the parameters in the left column of this same table, except for H_c which was as in Würsch and Craig (2014). We did not consider different parameters for the model run and assimilation as we focused on the perfect model scenario.

In Fig. 2.9 we can see a typical realization of the model with this alternative set of parameters. Contrasting this figure with Fig. 2.3, which is the same

except for the parameters, one can see that in the alternative scenario the patterns of precipitations are sensibly different: there is more rain in the domain and more contiguous small rain events. The wind field, however, is qualitatively similar. The fluid height field also displays more clouds, which explains the presence of more abundant rain.

The qualitative conclusions concerning the ability of the local EnKPFs to better capture dry and wet areas in the domain, as illustrated in Fig. 2.4, are not changed with the alternative set of parameters. However, in Fig. 2.10 the evolution of the CRPS during 6 hours of high-frequency observations assimilation show different results from the one in Figs. 2.5 and 2.6. In particular, the LEnKF is now clearly better at estimating the rain field, but on the other hand it suffers more acutely in terms of fluid height, where the U-pattern discussed earlier is very visible. The same phenomenon happens for the the low-frequency setup: the boxplots in Fig. 2.11 are sensibly different from the ones in Fig. 2.8.

The fact that the conclusions drawn from the experiments with the modified SWEQ model are sensitive to the choice of parameters indicates some inherent problems with this model as a tool to test data assimilation algorithms. Indeed, there is no *right* choice of parameters which to agree on, and one can adapt them according to its need (Würsch and Craig, 2014; Würsch, 2014). The experience gained with this model is still valuable, but in the future I would not advise someone to use it because of this reason.

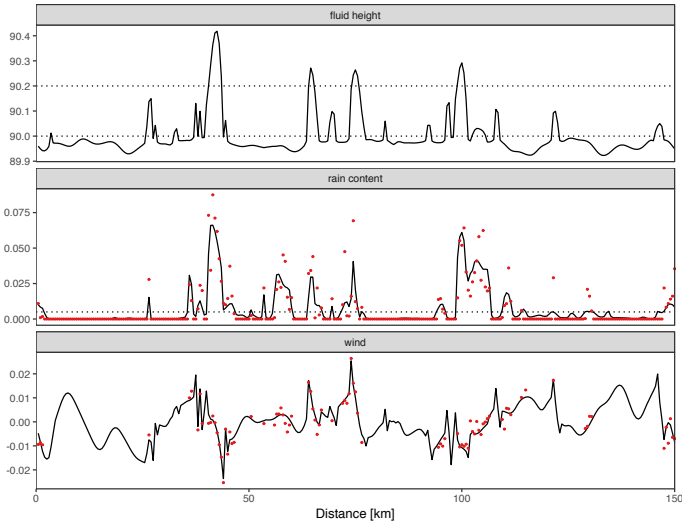

Figure 2.9: Example of the modified SWEQ model as in Fig. 2.3, except with an alternative set of parameters. Artificial radar observations are the red dots, and critical values (h_c, h_r and r_c) are shown as dashed lines.

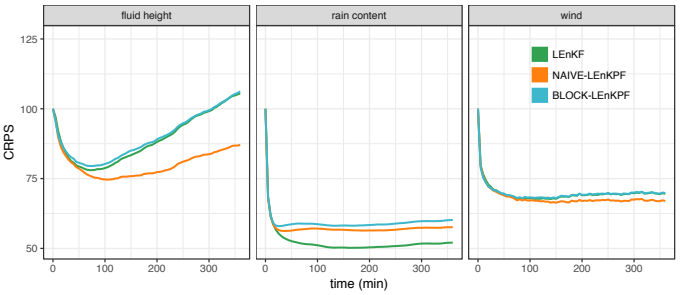

Figure 2.10: Evolution of the CRPS in the first 6 hours with high-frequency observations and the alternative parameters. The value is given as a percentage relative to a free forecast run. Notice the truncated y-axis.

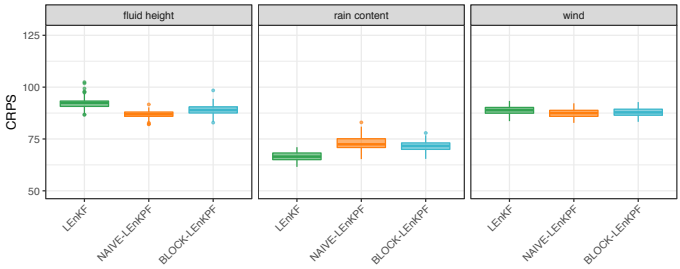

Figure 2.11: Low-frequency observations assimilated for a period of 3 days with the alternative parameters. Boxplot of the CRPS for the different algorithms and fields considered. The values are given relative to a free forecast. Notice the truncated y-axis.

2.B.2 Filter divergence with high-frequency observations

One feature of the numerical experiments that we found particularly intriguing was the influence of the assimilation frequency on the results, and in particular the appearance of the U-pattern in the case of high-frequency observations. This phenomenon is interesting as it highlights some issue with the EnKPF and EnKF that would need to be fixed. Even though in practice we are still far from assimilating data at frequencies higher than hourly, it might become feasible in the future with radar data. Also, the U-pattern could appear for lower-frequency observations too, if the observation error is sufficiently small.

To study the phenomenon in more detail we conducted the same experiment as in Section 2.4, but with a richer set of frequencies. In Fig. 2.12 we show the evolution of the CRPS as a function of lead time for increasing frequency of observations. A frequency of 5 min corresponds to the high-frequency scenario of Section 2.5.1, while a frequency of 30 min corresponds to the low-frequency scenario of Section 2.5.2.

The U-pattern for the fluid height field is clearly visible and becomes more and more acute as the frequency is increased. It is interesting to see that it also appears for the wind field at frequency faster than 5 min, which we had not noticed before. For the rain field it seems that only the local EnKPFs are subject to this problem, where a U-pattern appears clearly for high-frequency, but was not very visible at a frequency of 5 min. It is not clear why the LEnKF does not seem to suffer from the same issue.

To test if this phenomenon is particular to the modified SWEQ model, we ran similar experiments with the Lorenz96 model (Lorenz and Emanuel, 1998). The setup used is the same as in Frei and Künsch (2013), except that we vary the frequency of observations and use a smaller assimilation window. The results are averaged over a 100 simulations. As for the modified SWEQ, we show all the results relative to the performance of a free-forecast run for easier comparison. Here we look at the Root Mean Squared Error (RMSE) but similar conclusions apply to the CRPS. We also included the global EnKF and EnKPF without tapering as reference.

In Fig. 2.13 we can see the evolution of the relative RMSE as a function of lead time for various frequencies of observations. In the case of a frequency of 0.4, which is the same used in Frei and Künsch (2013), all local algorithms perform relatively well, with the LEnKF and naive-LEnKPF

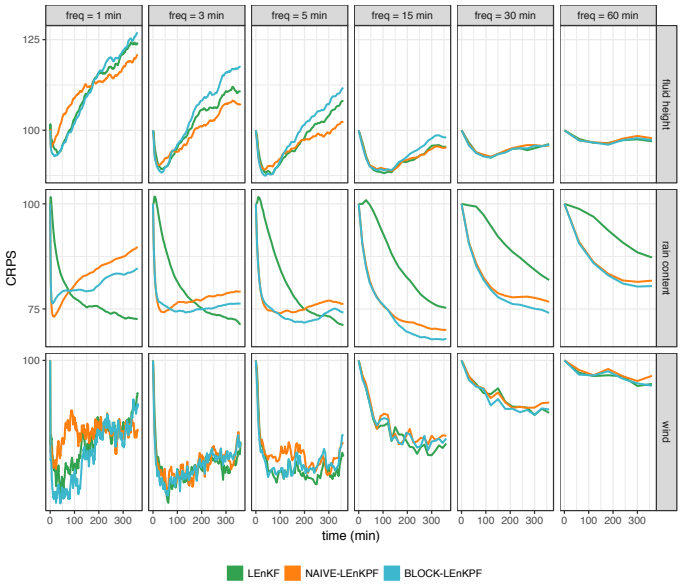

Figure 2.12: Evolution of CRPS as a function of lead time for the different fields and different frequency of observations with the modified SWEQ model. Referring to Section 2.5, the case with a frequency of 5 min corresponds to the high-frequency experiment, while the case with frequency of 30 min corresponds to the low-frequency experiment.

having an advantage over the block-LEnKPF. The global algorithms, however, perform rather poorly, which is to be expected without the use of any localization and a relatively small ensemble size. What is interesting is that the U-pattern also appears with this model. The block-LEnKPF suffers from it the earliest, already at 0.2 frequency. Then at a frequency of 0.1 the LEnKF shows some issue, while the naive-LEnKPF is the only method still stable. At a frequency of 0.05 all methods display a strong U-pattern.

The reasons for the appearance of the U-pattern are still poorly understood, but the two examples discussed here show that it is not a phenomenon particular to our modified SWEQ model setup. A possible intuition to explain what is going on is that as the frequency is increased, the effective number of observations assimilated increases. On the other hand, some variability is introduced in the ensemble between two assimilation times, but less and less as the frequency gets higher. In the limit of infinitely fast assimilation cycles, the situation is equivalent to assimilating all the observations together (assuming that serial and batch assimilation of observations are equivalent, which is not strictly the case for ensemble methods). In terms of Bayesian update of the distribution, it means that we have a very peaked likelihood, which leads to a poorly defined posterior. For the EnKPFs it is clear that with a very peaked likelihood the variance of the weights $\alpha^{\gamma,i}$ is large and ultimately might lead to sample degeneracy. In other words, the more observations are assimilated, the larger the ensemble size should be, or the smaller the localization radius should be used. This comes as no surprise from the discussion on the behavior of PFs in high-dimension as discussed in Section 1.4. What is maybe more surprising is that the EnKF is also subject to this effect, but it makes sense in the light of the intuition given here of a very peaked likelihood.

The phenomenon described here is not without connection to the catastrophic filter divergence discussed in Gottwald and Majda (2013). They work with a 5-dimensional Lorenz96 model and show that catastrophic filter divergence (where the error goes to machine infinity) occurs for sparse observation networks with sufficiently small observation error variance and small ensemble size. We could not compute the same statistics that they used as our setup is higher dimensional and it would take too much time. However, it seems that the U-pattern that we observe is influenced by similar factors, in particular we also identify an increase in divergence for higher assimilation frequency. Furthermore, the problem becomes more acute when the observation variance is reduced (not shown here), indicating that it is a case of what is called catastrophic divergence in Gottwald

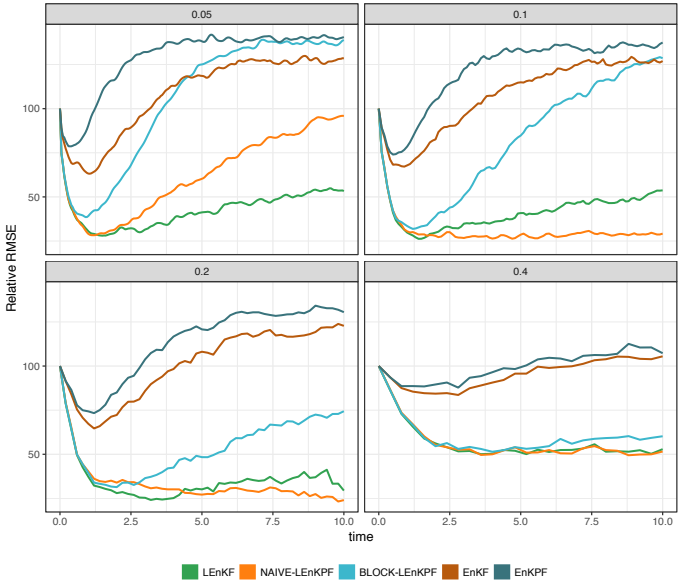

Figure 2.13: Influence of frequency on the evolution of RMSE as a function of lead time for the Lorenz96 model. A frequency of 0.4 is the same as used in (Frei and Künsch, 2013).

and Majda (2013). What we have not observed is a reduction of error for very high frequency, but we have probably not pushed it far enough. In Gottwald and Majda (2013) they write that localization and covariance inflation can efficiently control the problem of divergence. From Fig. 2.13 we can see that indeed the LEnKF always has an error smaller than the free forecast and thus does not diverge, even if it displays the U-pattern. However it seems that for the EnKPF localization is not enough to control the problem. Further research should be conducted to better understand the factors at play and to find potential remedies.

Chapter 3

Beating the curse of dimensionality: a case study

Most of the content of this chapter has been published in Robert and Künsch (2017a). A paragraph with a figure in Section 3.4.1 was not included in the original publication. Sections 3.1 to 3.3 are essentially repetition, while Section 3.4 presents new numerical experiments with a conjugate normal setup and with the Lorenz96 model.

3.1 Introduction

Monte Carlo methods are becoming increasingly popular for filtering in large-scale geophysical applications, such as reservoir modeling and numerical weather prediction, where they are often called ensemble methods for data assimilation. The challenging (and interesting) peculiarity of this type of applications is that the state space is extremely high dimensional (the number of dimensions of the state x is typically of the order of 10^8 and the dimension of the observation y of the order of 10^6), while the computational cost of the time integration step limits the sample size to less than a hundred. Because of those particularly severe constraints, the emphasis is on developing approximate but highly efficient methods, typically relying

on strong assumptions and exploiting parallel architectures.

The PF provides a fully general Bayesian solution to filtering (Gordon et al., 1993; Pitt and Shephard, 1999; Doucet et al., 2001), but it is well-known that it suffers from sample degeneracy and cannot be applied to high-dimensional settings (Snyder et al., 2008). The most popular alternative to the PF in large-scale applications is the EnKF (Evensen, 1994, 2003), a successful but heuristic method, which implicitly assumes that the predictive distribution is Gaussian.

Three main routes for adapting the PF to high-dimensional settings can be identified. The first one is to use an adaptive PF with a carefully chosen proposal distribution (Pitt and Shephard, 1999; van Leeuwen, 2010). A second approach is to build hybrid methods between the EnKF and the PF, as for example the EnKPF (Frei and Künsch, 2013). A third route is localization, as it is a key element of the success of the EnKF in practice and could avoid the curse of dimensionality (Snyder et al., 2008; Rebeschini and Handel, 2015).

The first approach requires an explicit model for the transition probabilities, which is typically not available in practical applications. Furthermore Snyder et al. (2015) showed that even with the optimal proposal distribution the PF suffers from the curse of dimensionality. Therefore in the present paper we focus on the second and third approaches and explore some possible localized algorithms based on the PF and the EnKPF. In a simulation study, we extend an example of Snyder et al. (2008) to illustrate how localization seemingly overcomes the curse of dimensionality, but at the same time introduces some harmful discontinuities in the estimated state. In a second experiment we show how local algorithms can be applied effectively to a filtering problem with the Lorenz96 model (Lorenz and Emanuel, 1998). The results from these numerical experiments highlight key differences between the algorithms and demonstrate that local EnKPFs are promising candidates for large-scale filtering applications.

3.2 Ensemble filtering algorithms

Consider a state space model with state process (x_t) and observations (y_t), where the state process evolves according to some deterministic or stochastic dynamics and the observations are assumed to be independent given the state process, with likelihood $p(x_t|y_t)$. The goal is to estimate the

conditional distribution of x_t given $y_{1:t} = (y_1, \ldots, y_t)$, called the *filtering* distribution and which we denote by π_t^f. In general it is possible to solve this problem recursively by alternating between a *prediction* step where the filtering distribution at time $(t-1)$ is propagated into the predictive distribution π_t^p at time t, and an *update* step, also called assimilation, where the predictive distribution is updated with the current observation to compute π_t^f. The update step is done by applying Bayes' rule as $\pi_t^f(x) \propto \pi_t^p(x) \cdot p(x|y_t)$, where the predictive distribution is the prior and the filtering distribution the posterior to be estimated.

Sequential Monte Carlo methods approximate the predictive and filtering distributions by finite samples, or ensembles of *particles*, denoted by $(x_t^{p,i})$ and $(x_t^{f,i})$ respectively, for $i = 1, \ldots, k$ (Doucet et al., 2001). The update step consists in transforming the predictive ensemble $(x_t^{p,i})$ into an approximate sample from the filtering distribution π_t^f. We briefly present the PF and EnKF in this context and give an overview of the EnKPF. Henceforth we consider the update step only and drop the time index t. Additionally for the EnKF and EnKPF we assume that the observations are linear and Gaussian, i.e. $p(x|y) = \phi(y;\ Hx, R)$, the Gaussian density with mean Hx and covariance R evaluated at y.

The Particle Filter approximates the filtering distribution as a mixture of point masses at the predictive particles, reweighed by their likelihood. More precisely:

$$\hat{\pi}_{PF}^f(x) = \sum_{i=1}^{k} w_i \, \delta_{x^{p,i}}(x), \qquad w_i \propto \phi(y;\ Hx^{p,i}, R). \tag{3.1}$$

A non-weighted sample from this distribution can be obtained by resampling, for example with a balanced sampling scheme (Künsch, 2005). The PF is asymptotically correct (also for non-Gaussian likelihoods), but to avoid sample degeneracy it needs a sample size which increases exponentially with the size of the problem (for more detail see Snyder et al. (2008)).

The Ensemble Kalman Filter is a heuristic method which applies a Kalman filter update to each particle with stochastically perturbed observations. More precisely it constructs $(x^{f,i})$ as a balanced sample from the

following Gaussian mixture:

$$\hat{\pi}^f_{EnKF}(x) = \sum_{i=1}^{k} \frac{1}{k}\, \phi(x;\ x^{p,i} + \hat{K}(y - Hx^{p,i}), \hat{K}R\hat{K}'), \tag{3.2}$$

where $\hat{K}$ is the Kalman gain estimated with $\hat{\Sigma}^p$, the sample covariance of $(x^{p,i})$. The stochastic perturbations of the observations are added to ensure that the filter ensemble has the correct posterior covariance on expectation.

The Ensemble Kalman Particle Filter combines the EnKF and the PF by decomposing the update step into two stages as $\pi^f(x) \propto \pi^p(x) \cdot p(x|y)^\gamma \cdot p(x|y)^{1-\gamma}$, following the progressive correction idea of Musso et al. (2001). The first stage, going from $\pi^p(x)$ to $\pi^\gamma(x) \propto \pi^p(x) \cdot p(x|y)^\gamma$ is done with an EnKF. The second stage is done with a PF and goes from $\pi^\gamma(x)$ to $\pi^f(x) \propto \pi^\gamma(x) \cdot p(x|y)^{1-\gamma}$. The resulting posterior distribution can be derived analytically as the following weighted Gaussian mixture:

$$\hat{\pi}^f_{EnKPF}(x) = \sum_{i=1}^{k} \alpha^{\gamma,i}\, \phi(x;\ \mu^{\gamma,i}, \Sigma^\gamma), \tag{3.3}$$

where the expressions for the parameters of this distribution and more details on the algorithm can be found in Frei and Künsch (2013). To produce the filtering ensemble $(x^{f,i})$, one first samples the mixture components with probability proportional to the weights $\alpha^{\gamma,i}$, using for example a balanced sampling scheme, and then adds an individual noise term with covariance Σ^γ to each particle. The parameter γ defines a continuous interpolation between the PF ($\gamma = 0$) and the EnKF ($\gamma = 1$). In the present study the value of γ is either fixed, for the sake of comparison, or chosen adaptively. In the later case γ is chosen such that the equivalent sample size of the filtering ensemble is within some acceptable range. Alternative schemes for choosing γ such as minimizing an objective cost function are currently being investigated but are beyond the scope of this work.

3.3 Local algorithms

Localization consists essentially in updating the state vector by ignoring long range dependencies. This is a sensible thing to do in geophysical applications where the state represents discretized spatially correlated fields

of physical quantities. By localizing the update step and using local observations only, one introduces a bias, but achieves a considerable gain in terms of variance reduction for finite sample sizes. For local algorithms the error is asymptotically bigger than for a global algorithm, but it is not dependent on the system dimension anymore and therefore avoids the curse of dimensionality. Furthermore, local algorithms can be efficiently implemented in parallel and thus take advantage of modern computing architectures.

The LEnKF consists in applying a separate EnKF at each site, but limiting the influence of the observations to sites that are spatially close (there are different ways to accomplish this in practice, see for example Houtekamer and Mitchell (2001); Ott et al. (2004); Hunt et al. (2007)). Analogously, we define the Local Particle Filter (LPF) as a localized version of the PF, where the update is done at each location independently, considering only observations in a ball of radius ℓ. In order to avoid arbitrary "scrambling" of the particles indices, we use a balanced sampling scheme (Künsch, 2005), and some basic ad-hoc methods to reduce the number of discontinuities, but we do not solve this problem optimally as it would greatly hinder the efficiency of the algorithm.

For the EnKPF we define two different local algorithms: the naive-LEnKPF, in which localization is done exactly as for the LEnKF, and the block-LEnKPF, in which the observations are assimilated sequentially but their influence is restricted to a local area. The naive-LEnKPF does not take particular care of the introduced discontinuities beyond what is done for the PF, but it is straightforward to implement. The block-LEnKPF, on the other hand, uses conditional resampling in a transition area surrounding the local assimilation window, which ensures that there are no sharp discontinuities, but it involves more overhead computation. For more detail about the local EnKPF algorithms see Robert and Künsch (2017b).

3.4 Simulation studies

We conducted two simulation studies: first a one-step conjugate normal setup where the effect of localization can be closely studied, and second a cycled experiment with the Lorenz96 model, a non-linear dynamical system displaying interesting non-Gaussian features.

3.4.1 Conjugate normal setup

We consider a simple setup similar to the one in Snyder et al. (2008), with a predictive distribution π^p assumed to be a N-dimensional normal with mean zero and covariance Σ^p. To imitate the kind of smooth fields that we encounter in geophysical applications, we construct the covariance matrix as $\Sigma^p_{ii} = 1$ and $\Sigma^p_{ij} = K_{GC}(d(i,j)/r)$, where K_{GC} is the Gaspari-Cohn kernel (Gaspari and Cohn, 1999), $d(i,j)$ the distance between sites i and j on a one-dimensional domain with periodic boundary conditions, and the radius r in the denominator is chosen such that the covariance has a finite support of 20 grid points. From this process we generate observations of every component of x and standard Gaussian noise:

$$x \sim \mathcal{N}(0, \Sigma^p), \qquad y|x \sim \mathcal{N}(x, I). \tag{3.4}$$

In order to study the finite sample properties of the different algorithms, we compute the Mean Squared Error (MSE) of the ensemble mean in estimating the value x at each location, which we denote by $\text{MSE}(x)$. Because the prior is conjugate to the likelihood, we can compute the $\text{MSE}(x)$ of the posterior mean analytically for known Σ^p as the trace of the posterior covariance matrix and use this as a reference. For the simulation we use a sample size of $k = 100$ and average the results over 1000 runs. It should be noted that because the predictive distribution is normal, this setup is favorable to the EnKF and LEnKF, but the EnKPFs should still perform adequately. For the local algorithms the localization radius ℓ was set to 5, resulting in a local window of 11 grid points, which is smaller than the correlation length used to generate the data. Later on we study the effect of ℓ on the performance of the algorithms. For the EnKPF algorithms the parameter γ was fixed to 0.25, which means a quarter of EnKF and three-quarter of PF. In practice one would rather choose the value of γ adaptively, but the exact value does not influence the qualitative conclusions drawn from the experiments and fixing it in this way makes the comparison easier.

An example of a sample from the filtering distribution produced by different local algorithms is shown in Fig. 3.1, with each particle represented as a light blue line, the true state in dark and the observations in red. For more clarity the ensemble size is set to 10 and the system dimension to 40. While all algorithms manage to recover more or less the underlying state, it is clear that they vary in terms of quality. The LPF in particular suffers from sample depletion, even when applied locally, and displays strong

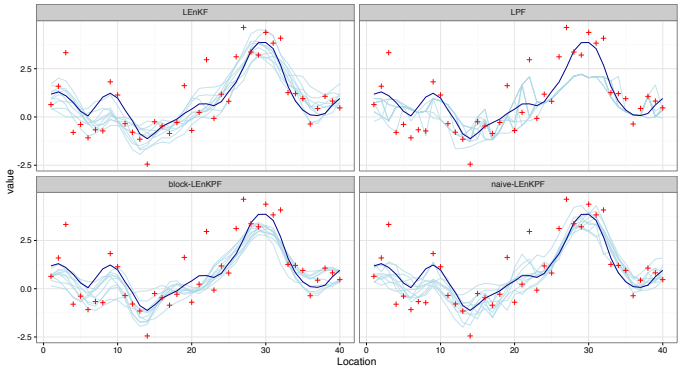

Figure 3.1: Example of analysis ensemble with different local algorithms. Each particle is a light blue line, the true state in dark and the observations in red. The ensemble size is restricted to 10 and the domain size to 40 for better legibility.

discontinuities. If one looks closely at the naive-LEnKPF ensemble, discontinuities can also be identified. The block-LEnKPF and the LEnKF, on the other hand, produce smoother posterior particles. This example is useful to highlight the behavior of the different local algorithms qualitatively, but we now proceed to a more quantitative assessment with a repeated simulations experiment.

Beating the curse of dimensionality

In the first row of Fig. 3.2, the $\text{MSE}(x)$ is plotted as a function of the system dimension N, for the global algorithms on the left and the local algorithms on the right. The values are normalized by the optimal $\text{MSE}(x)$ to make them more interpretable. The PF degenerates rapidly, with an $\text{MSE}(x)$ worse than using the prior mean (upper dashed line). The EnKF and the EnKPF suffer as well from the curse of dimensionality, although to a lesser extent. The local algorithms, on the other hand, are immune to the increase of dimensions N and their $\text{MSE}(x)$ is constant and very close the optimum, which confirms that localization is working as expected. The LEnKF, naive-LEnKPF and block-LEnKPF make an error of less than 5% while the LPF is 20% worse than the optimum.

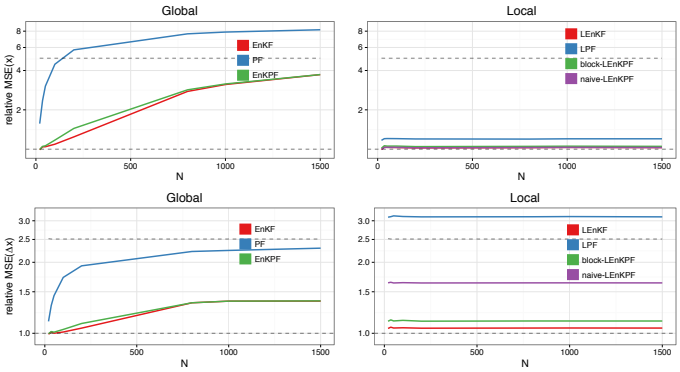

Figure 3.2: Illustration of the relationship between the system dimension N and different quantities. In the first row is the $\text{MSE}(x)$ for the global algorithms on the left and the local algorithms on the right. In the second row the same but for the $\text{MSE}(\Delta x)$. All the values are given relative to the optimal one obtained with the true posterior distribution (dashed line at 1). The relative MSE of using the prior without using any observation is given by the second dashed line. Notice the log-scale on the y-axis.

The cost of localization

As the old statistical adage goes, there is no free lunch: localization comes at a cost, particularly for PF algorithms. When doing the update locally with the EnKF, the filtering samples are relatively smooth fields, because the update applies spatially smoothly varying corrections to the predictive ensemble. However, for the LPF, when different particles are resampled at neighboring sites, arbitrarily large discontinuities can be created. While this might be discarded as harmless, it is not the case when the fields of interest are spatial fields of physical quantities used in numerical solvers of partial differential equations. One way to measure the impact of discontinuities is to look at the MSE in estimating the lag one increments Δx, which we denote as $\text{MSE}(\Delta x)$. While the $\text{MSE}(x)$ is computed for the posterior mean, the $\text{MSE}(\Delta x)$ is computed for each particle separately and then averaged. We again compute the expected $\text{MSE}(\Delta x)$ under the conjugate posterior distribution and use it as reference.

The plots in the second row of Fig. 3.2 show this quantity for the different algorithms averaged over 1000 simulation runs. The $\text{MSE}(\Delta x)$ of the local algorithms is still constant as a function of N, as expected, but in the cases of the naive-LEnKPF and the LPF its value is worse than for the respective global algorithms. On the other hand, the LEnKF and the block-LEnKPF

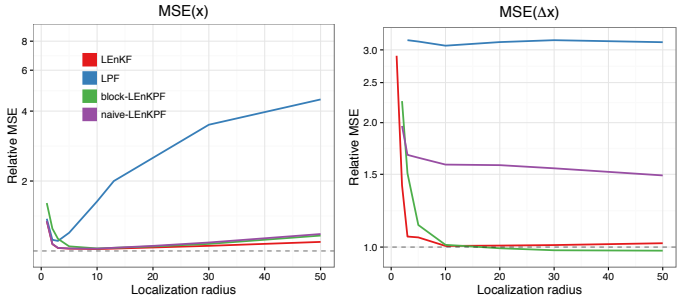

Figure 3.3: Trade-off of localization: influence of ℓ on $\text{MSE}(x)$ and $\text{MSE}(\Delta x)$ for the local algorithms. The ensemble size k was fixed to 100 and the system dimension N to 200. Notice the log-scale on the y-axis.

improve on their global counterparts and have an error relatively close to the optimum.

Localization trade-off

In the previous experiment we fixed ℓ, the localization radius, to 5, and looked at what happens in terms of prediction accuracy with the $\text{MSE}(x)$, and in terms of discontinuities with the $\text{MSE}(\Delta x)$. In Fig. 3.3 we now look at MSE as a function of ℓ, fixing N to 200 and k to 100. For large values of ℓ the $\text{MSE}(\Delta x)$ is smallest as discontinuities are avoided, but the $\text{MSE}(x)$ is not optimal, particularly for the LPF. As ℓ is reduced the $\text{MSE}(x)$ decreases for all methods, while $\text{MSE}(\Delta x)$ is kept constant for a wide range of ℓ values. At some point, different for each algorithm, the localization is too strong and becomes detrimental, with both $\text{MSE}(x)$ and $\text{MSE}(\Delta x)$ sharply increasing. This behavior illustrates the trade-off at hand when choosing the localization radius: picking a too small value introduces a bias by neglecting useful information and creates too much discontinuities, while choosing a too large value does not improve $\text{MSE}(\Delta x)$ but leads to poorer performance in terms of $\text{MSE}(x)$.

In Fig. 3.4 this trade-off is illustrated most clearly. Both $\text{MSE}(x)$ and $\text{MSE}(\Delta x)$ are displayed in a same plot, with each curve showing their evolution as a function of ℓ, encoded by color. One can see that initially increasing ℓ leads to both a sharp decrease in $\text{MSE}(x)$ and $\text{MSE}(\Delta x)$. However after some point increasing ℓ continues to reduce the $\text{MSE}(\Delta x)$ but leads to an increase in $\text{MSE}(x)$. Each method is displayed in a different

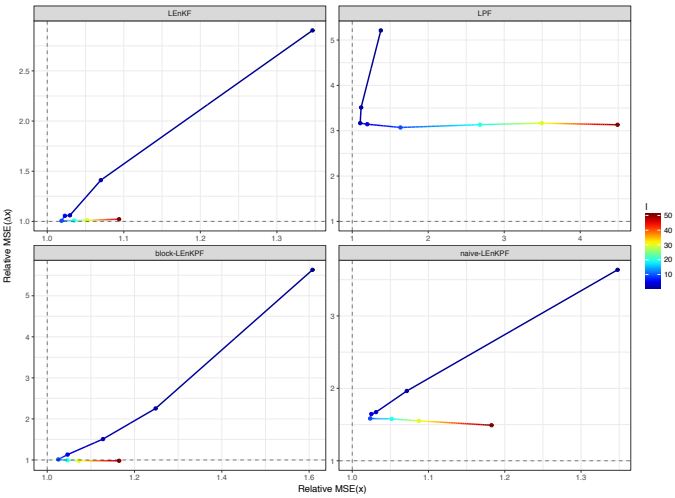

Figure 3.4: Trade-off of localization: interplay between $\text{MSE}(x)$ and $\text{MSE}(\Delta x)$ as a function of ℓ (notice the different scale). Each algorithm is represented as a parametric curve with $\text{MSE}(x)$ on the x-axis and $\text{MSE}(\Delta x)$ on the y-axis and with the color representing ℓ. The ensemble size k was fixed to 100 and the domain size N to 200.

panel with a different scale, but the general functional shape of the relationship is the same for all of them and highlights the trade-off involved when choosing the strength of localization.

3.4.2 Filtering with the Lorenz96 model

The Lorenz96 model (Lorenz and Emanuel, 1998) is a 40-dimensional nonlinear dynamical system which displays a rich behavior and is often used as a benchmark for filtering algorithms. In Frei and Künsch (2013) it was shown that the EnKPF outperforms the LEnKF in some setups of the Lorenz96 model, but the sample size required was of 400. In the present experiment we use the same setup as in Frei and Künsch (2013) but with much smaller and realistic ensemble sizes. The data are assimilated at time intervals of 0.4, which leads to strong non-linearities and thus highlights better the relative advantages of the EnKPF. Each experiment is run for 1000 cycles and repeated 20 times, which provides us with stable estimates of the average performance of each algorithm. As in Frei and Künsch (2013), the parameter γ of the EnKPFs is chosen adaptively such

that the equivalent sample size is between 25 and 50% of the ensemble size. It should be noted that for local algorithms, a different γ is chosen at each location, which provides added flexibility and allows to adapt to locally non-Gaussian features of the distribution. We consider $\text{MSE}(x)$ only and denote it simply by MSE. It also takes errors in the estimation of increments into account through integration in time during the propagation steps.

In the left panel of Fig. 3.5 the MSE of the global algorithms is plotted against ensemble size. The PF is not represented as it diverges for such small values. The MSE is computed relative to the performance of the prior, which is simply taken as the MSE of an ensemble of the same size evolving according to the dynamical system equations but not assimilating any observations. With ensemble sizes smaller than 50, the filtering algorithms are not able to do better than the prior, which means that trying to use the observations actually makes them worse than not using them at all. Only for ensemble sizes of 100 and more do the global algorithms start to become effective. In practice we are interested in situations where the ensemble size is smaller than the system dimension (here 40), and thus the global methods are clearly not applicable.

On the right panel of Fig. 3.5 we show the same plot but for the local algorithms. For sample sizes as small as 20 or 30 the performances are already quite good. The LPF, however, does not work at all, probably because it still suffers from sample depletion and because the discontinuities it introduces have a detrimental impact during the prediction step of the algorithm. The block-LEnKPF clearly outperforms the other algorithms, particularly for smaller sample sizes. This indicates that it can localize efficiently the update without harming the prediction step by introducing discontinuities in the fields.

In order to better highlight the trade-off of localization, we plot similar curves but as a function of the localization radius ℓ in Fig. 3.6. One can see that for small k (left panel), the error is increasing with ℓ, which shows that localization is absolutely necessary for the algorithm to work. For $k = 40$ (right panel), the MSE first decreases and then increases, with an optimal ℓ. Experiments with larger values display curves that get flatter and flatter as k increases, showing that as the ensemble size is larger, the localization strength needed is smaller, as expected.

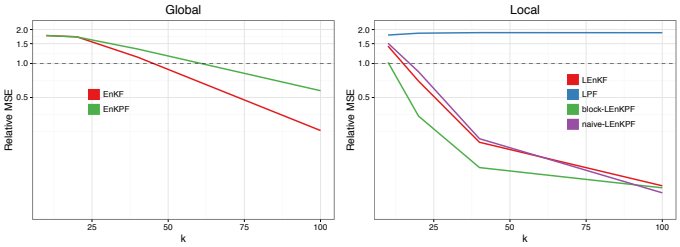

Figure 3.5: Global and local algorithms results with the Lorenz96 model. On the y-axis the MSE and on the x-axis increasing ensemble sizes k. There is no global PF as the algorithm does not work for such small ensemble sizes. The MSE is computed relative to the MSE of an ensemble of the same size but which does not assimilate any observation. Notice the log-scale on the y-axis.

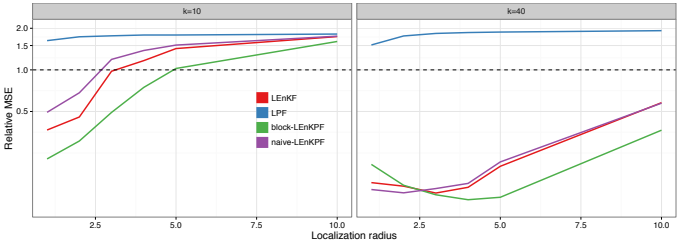

Figure 3.6: Interplay of ensemble size k and localization radius ℓ for the the Lorenz96 model. The relative MSE is plotted as a function of ℓ for different value of k in the different panels. Notice the log-scale on the y-axis.

3.5 Conclusion

Localization is an effective tool to address some of the difficulties associated with high-dimensional filtering in large-scale geophysical applications. Methods such as the EnKF can be localized easily and successfully as they vary smoothly in space. At first sight, the LPF does seem to overcome the curse of dimensionality; however, looking more carefully, one notices that it introduces harmful discontinuities in the updated fields. The two localized EnKPFs both overcome the curse of dimensionality and handle better the problem of discontinuities.

The simple conjugate example studied in this paper highlighted the potential improvements coming from localization, as well as the pitfalls when applied blindly to the PF. The trade-off between the bias coming from localization and the gain coming from the reduced variance was illustrated by exploring the behavior of the algorithms as a function of the localization radius ℓ. Experiments with the Lorenz96 model showed that local algorithms can be successfully applied with ensemble sizes as small as 20 or 30, and highlighted the localization trade-off. In particular, the block-LEnKPF fared remarkably well, outperforming both the naive-LEnKPF and the LEnKF in this challenging setup. This confirms other results that we obtained with more complex dynamical models mimicking cumulus convection (Robert and Künsch, 2017b) and encourages us to pursue further research with localized EnKPFs in a large-scale application in collaboration with Meteoswiss.

Chapter 4

Other topics related to localization

In this chapter we present unpublished material related to the localization of the EnKPF. In Section 4.1 we illustrate the local EnKPFs through a 3-dimensional example and discuss possible variations and extensions of the block-LEnKPF. In Section 4.2 we present an algorithm to resample particles locally with a balanced scheme while ensuring that the number of discontinuities between neighboring sites is kept to a minimum.

4.1 Local EnKPFs: a 3D example

To explore in more detail the behavior of the localized EnKPF algorithms we study a simple 3-dimensional example with a multimodal prior. The effect of localization is particularly well illustrated and the difference between the block-LEnKPF and the naive-LEnKPF put in a new light. The example is somehow extreme and is meant to test the limitations of the algorithms when the assumptions are not fulfilled. Furthermore we use the example to discuss variations of the block-LEnKPF, in particular we compare the results of using a direct sample instead of using residuals in Section 4.1.4, and we consider a non-linear generalization of the block-LEnKPF in Section 4.1.5.

4.1.1 Model for the prior and the observation

As prior we define the following mixture of three Gaussians:

$$\pi^b(x) = \frac{1}{4}N(\mu^1, \Sigma) + \frac{1}{4}N(\mu^2, \Sigma) + \frac{1}{2}N(\mu^3, \Sigma) \tag{4.1}$$

where

$$\mu^1 = \begin{pmatrix} 1 \\ -0.5 \\ -1 \end{pmatrix},\ \mu^2 = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix},\ \mu^3 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \text{ and} \tag{4.2}$$

$$\Sigma = \sigma^2 \cdot \begin{pmatrix} 1 & \rho & 0 \\ \rho & 1 & \rho \\ 0 & \rho & 1 \end{pmatrix} \tag{4.3}$$

where $\rho = 0.5$, and $\sigma = 1/5$. Now let us assume that we observe the first dimension only:

$$y = 1, \quad H = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \quad R = \sigma_y^2, \tag{4.4}$$

where the observation standard deviation is $\sigma_y = 0.5$.

The purpose of this prior is to have no correlation between the first and the third variable, but a strong non-linear dependency nonetheless. To verify that we can compute the covariance matrix of x, P^b, as:

$$P^b = \Sigma + \sum_{j=1}^{3} \alpha^j (\mu^j - \bar{\mu})(\mu^j - \bar{\mu})' = \begin{pmatrix} 0.540 & 0.145000 & 0 \\ 0.145 & 0.836875 & 0.895 \\ 0 & 0.895 & 1.040 \end{pmatrix},$$

where we denoted as α^j the weights of each component in Eq. (4.1) and $\bar{\mu}$ is the weighted mean of the μ^j. Following the notation for the block-LEnKPF in Section 2.3.2, the first dimension corresponds to x_u, the second to x_v and the third to x_w. Indeed x_u is correlated with x_v, x_v with x_w, but x_u is orthogonal to x_w. However, the factorization of the prior in Eq. (2.11) does not hold because x_u and x_w are clearly not independent. We will investigate further to see how much of an effect it has on the results.

4.1.2 Global assimilation

We generate a background ensemble of size 40 from the above prior distribution, with a balanced scheme for the mixture indicators. Using this

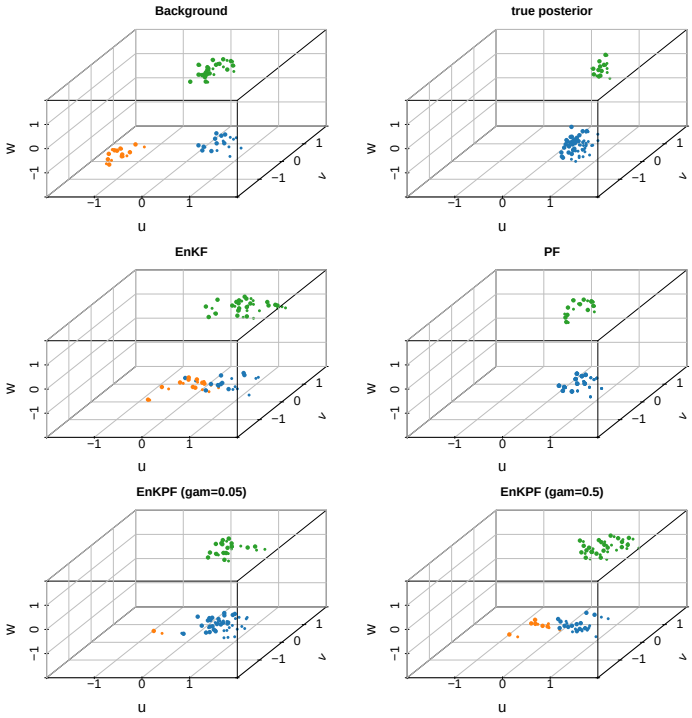

Figure 4.1: Background and analysis ensembles with various non-local assimilation algorithms for the 3D example. The color indicates the mixture components, blue is the first one, orange the second, and green the third. In the analysis the color indicates the component from which the resampled particles was taken.

ensemble we assimilate the observation y with the EnKF, the PF, and the EnKPF with parameter γ set to 0.05. We chose such a small γ to ensure that the analysis is far from the EnKF to make the illustration clearer. The true posterior can be computed analytically as a new Gaussian mixture, where the mean of the first component is unchanged, while the other two are moved towards the observations and the mixture weights are updated (a sample from the posterior is shown in the upper-right panel of Fig. 4.1).

The background and analysis ensembles are shown in Fig. 4.1, with the color indicating the mixture components. The observation y corresponds to a vertical plane going through $u = 1$, and is compatible with both the first and third mixture components (blue and green), but not with the second one (orange). The EnKF moves each particle towards the obser-

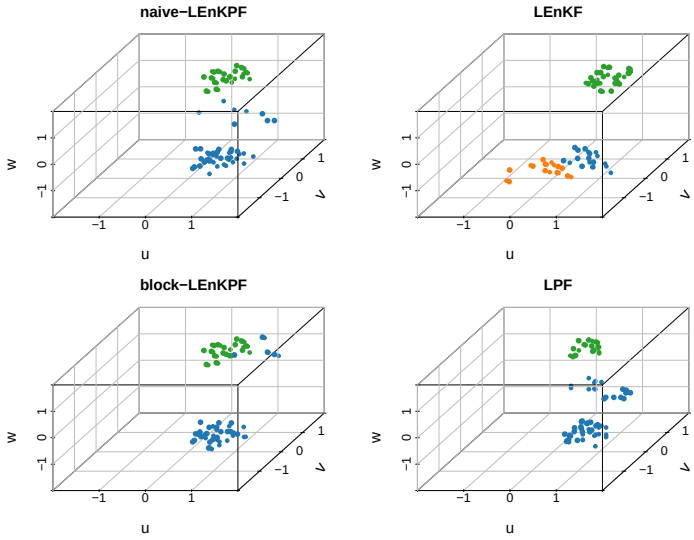

Figure 4.2: Analysis ensembles with various local assimilation algorithms for the 3D example. The color indicates the mixture components from which the resampled particles was taken: blue is the first one, orange the second, and green the third.

vations, resulting in elongated shapes quite far from the high-mass region of the prior. The PF works quite well in this low-dimensional example, in the sense that it is close to the sample from the true posterior and does not suffer too much from sample depletion. The EnKPF analysis is close to the PF but the influence of the EnKF is visible from the same elongated shapes of the mixture components.

4.1.3 Local assimilation

The same observation is now assimilated by four different local algorithms: the LEnKF, the naive-LEnKPF, the block-LEnKPF, and the LPF. To ensure that the covariance between x_u and x_w is really null we use a tapering matrix with ones on the diagonal, 0.5 on the sub- and super-diagonals, and 0 otherwise. For the LPF we simply apply a PF to x_u and x_v, and match with the particles x_w. For all algorithms x_w is kept unchanged.

The resulting analysis ensembles are shown in Fig. 4.2. The LEnKF is

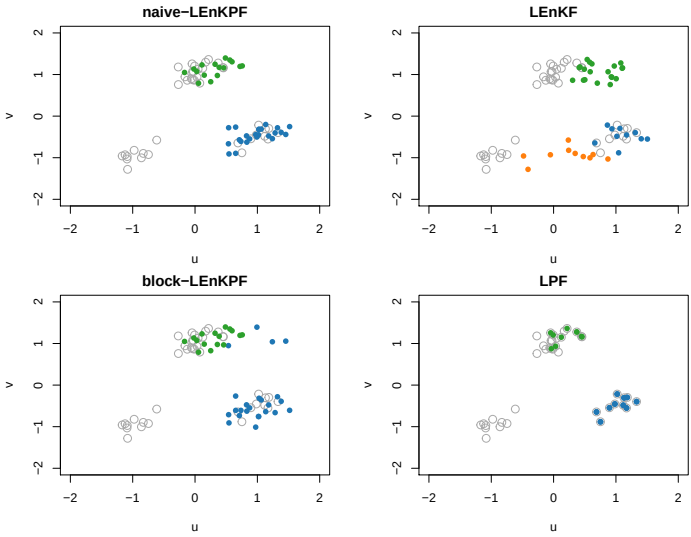

Figure 4.3: u and v marginals of analysis ensembles with various local assimilation algorithms for the 3D example. The color indicates the mixture components from which the resampled particles was taken: blue is the first one, orange the second, and green the third. The empty circles show the background ensemble members.

relatively close to the EnKF, except that it exerts a stronger pull on the third components (in orange) towards the observations, resulting in the two modes overlapping more than for the EnKF where they follow each other in the dimension u. The block-LEnKPF is more or less similar to the EnKPF, except for a group of particles coming from the first component (in blue) being put close to the third component (in green). This same group of blue particles is put in another region of space by both the naive-LEnKPF and the LPF. This artefact comes essentially from the patching of locally resampled particles, which we will now investigate in more detail.

For better illustration, in Fig. 4.3 we show the same as in Fig. 4.2 but projected onto the u and v axes. The two modes produced by the LEnKF in the bottom right component are clearly visible and come from the shifting of the particles from the second component (in orange) towards the observation on the right. By looking carefully it is possible to notice that the update in x_u is the same for the naive-LEnKPF and block-LEnKPF, as is expected. The particles coming from the first component (in blue) but put close to the third component (in green) by the block-LEnKPF

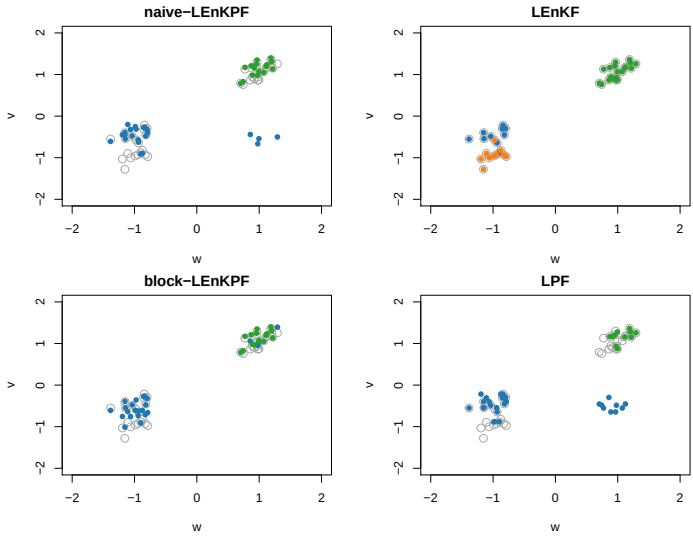

Figure 4.4: v and w marginals of analysis ensembles with various local assimilation algorithms for the 3D example. The color indicates the mixture components from which the resampled particles was taken: blue is the first one, orange the second, and green the third. The empty circles show the background ensemble members.

are well visible: they were resampled in x_u and then their analysis x_v was computed using Eq. (2.12). For the naive-LEnKPF however the analysis of x_v is the usual EnKPF analysis and there is no shift of the resampled particles toward the other component.

In Fig. 4.4 we can see the v and w marginals of Fig. 4.2. The creation of an artificial mode (in the bottom right) in the distribution by the LPF, and to a lesser extent by the naive-LEnKPF, is now evident. The x_u and x_v of these particles were sampled locally from the first component (in blue) but then they were matched with particles from the third component (in green) for x_w. This example illustrates well the nature of local resampling, which can be seen both as an advantage and a disadvantage: on the one hand it allows the ensemble to explore new regions of the state space which were not populated by the background ensemble, maybe too small to sample them; on the other hand it might create artefacts and put particles in region of state space that are physically not possible.

4.1.4 Sampling scheme for the block-LEnKPF

The block-LEnKPF did not create the artifical mode in the analysis ensemble that the naive-LEnKPF and LPF have in Fig. 4.4. The reason for this is that it does not apply an EnKPF analysis to x_v but rather a correction which implicitly takes into account the correlation between x_v and x_w through resampling of residuals. In the top-left panel of Fig. 4.5 we show the same marginals as in Fig. 4.4 but add the regression line corresponding to the conditional mean $\mu^b_{v|uw}$ defined in Eq. (2.16). The vertical bars are the residuals $r^{b,i}_v$ defined in Eq. (2.19) which were taken from the background sample and added to $\mu^b_{v|uw}$ evaluated at the analyses $x^{a,i}_u$.

On the top-right panel of Fig. 4.5 is the analysis ensemble of what we call the kriging-local EnKPF (kriging-LEnKPF): it is the same as the block-LEnKPF except that instead of using the residuals to sample from $\pi^b(x_v|x_u, x_w)$ it relies more strongly on Gaussian assumptions and for the cases where a particle in x_u has to be matched with a different particle in x_w it generates x_v as:

$$x^{a,i}_v = \mu^b_{v|uw}(x^{a,i}_v, x^{b,i}_v) + \xi^{b,i}_v, \quad \text{and} \quad \xi^{b,i}_v \sim \mathcal{N}(0, P^b_{v|uw}), \tag{4.5}$$

where we used $\mu^b_{v|uw}$ as a function of x_u and x_w, and where $P^b_{v|uw}$ was defined in Eq. (2.17). In fact because x_u and x_w are uncorrelated the $\mu^b_{v|uw}$ is independent of w and corresponds simply to the linear regression of x_v onto x_u, which is shown on the first row of Fig. 4.5 together with the 2.5% and 97.5% quantiles.

The difference between the block-LEnKPF and the kriging-LEnKPF is illustrated in Fig. 4.5. The first row shows that the $\xi^{b,i}_v$ sampled as in Eq. (4.5) (gray bars in right column) have a different structure than the residuals used by the block-LEnKPF (gray bars in left column): they do not have two modes but rather span the whole interval between the two modes. In the second row we show these residuals $r^{b,i}_v$ and the $\xi^{b,i}_v$ respectively versus the variable x_w. From this we can clearly see that the $r^{b,i}_v$ are strongly correlated with x_w: positive residuals are associated with $x_w = 1$ and negative residuals with $x_w = -1$. This dependency comes from the fact that some residuals were sampled from the second component (leftmost background particles in the upper panels), while others from the third component (topmost in the upper panels). In the last row we can see how this dependency ensures that the correlation between x_w and x_v

is maintained for the block-LEnKPF, while it is destroyed by the kriging-LEnKPF, which smears the separation between the modes. Relating this to Fig. 4.4 it becomes clear why the block-LEnKPF does not create the artefact at $x_w = 1$ and $x_v = -1$ present in the naive-LEnKPF and LPF.

4.1.5 Generalization of the block-LEnKPF

A natural idea that comes to mind to generalize the block-LEnKPF is to model the conditional distribution $\pi^b(x_v|x_u, x_w)$ in a more flexible manner. One feature that we would like to keep is that the conditional mean should depend on x_u only, or potentially on x_u and a small neighborhood of x_w, otherwise the algorithm is not local anymore. Another feature that we can easily keep is the idea of using the residuals instead of sampling, which spares us the trouble of estimating their distribution. One possible proposition of algorithm is to model the conditional mean $\mu^b_{v|uw}$ with a Generalized Additive Model (GAM) (Hastie and Tibshirani, 1990), but any other flexible regression could be used. The resulting algorithm which we call GAM-local EnKPF (GAM-LEnKPF) is identical to the block-LEnKPF except that it sample x_v as follows for the cases where a particle in x_u has to be matched with a different particle in x_w:

$$x_v^{a,i} = \beta_0 + \sum_{j=1}^{N_u} s_j(x_u^{a,i}(j)) + r_v^{b,i}, \tag{4.6}$$

where β_0 is an intercept, $s_j(\cdot)$ are univariate smoothers, typically splines, and the residuals are with respect to the predicted value with the above GAM: $r_v^{b,i} = x_v^{b,i} - \beta_0 - \sum s_j(x_u^{b,i}(j))$; N_u is the number of dimension of x_u while $x_u(j)$ select the jth component of x_u.

In our 3D example the GAM prediction is simply a univariate smoothing spline which predicts x_v as a function of x_u. The resulting analysis is shown in Fig. 4.6, where we see clearly how the conditional mean captures the non-linearity in the background distribution. However the second and third panel, constructed as the second and third row of Fig. 4.5, show that the GAM creates similar artefacts than the naive-LEnKPF and LPF, as its residuals are not correlated with x_w as they were for the block-LEnKPF. As discussed earlier, however, it is not clear if such a behavior is an advantage or a disadvantage in general, and one should not draw conclusions from this very particular example.

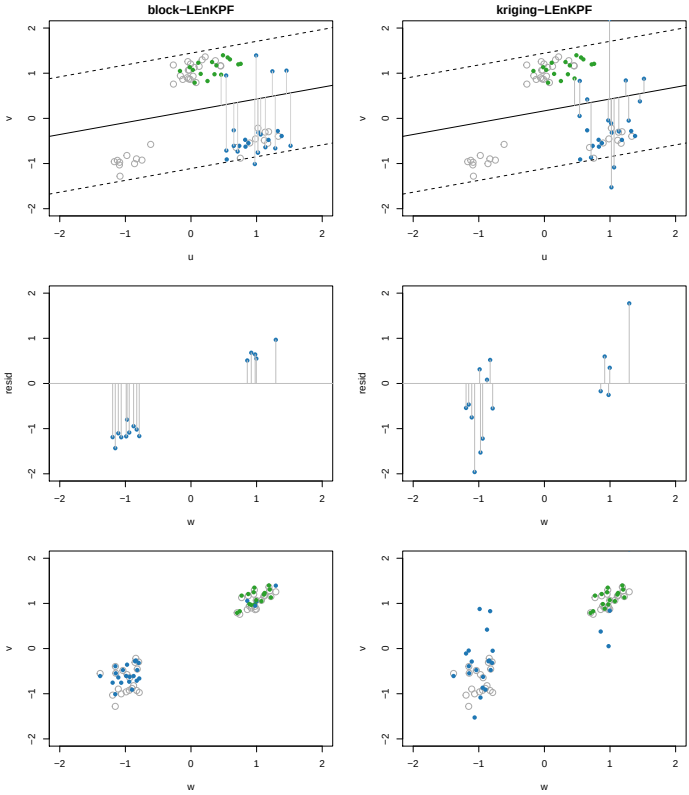

Figure 4.5: Illustration of the difference between the block-LEnKPF and the kriging-LEnKPF. In the top row we see the u and v marginals with the mean and 2.5% and 97.5% quantiles of the conditional normal distribution $\pi^b(x_u|x_v, x_w)$. In the second row the residuals of the first rows are plotted against the variable x_w. The third row shows x_v versus x_w.

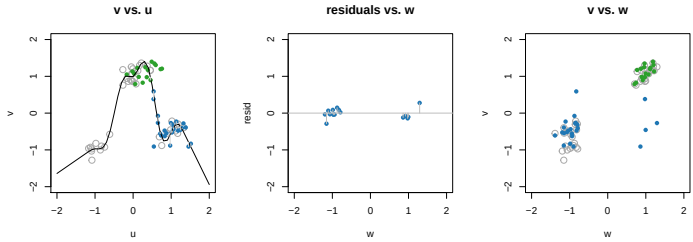

Figure 4.6: Illustration of the GAM-LEnKPF. In the left panel we see the u and v marginals with the estimated GAM regression curve $s(x_u)$. In the second panel are the residuals plotted against the variable x_w. The third row shows x_v versus x_w.

4.2 Local balanced sampling

In this section we discuss ways to produce locally balanced samples to use with localized PF type of algorithms. First we review univariate balanced sampling as is discussed for example in Carpenter et al. (1999); Crisan (2001) or Künsch (2005). Then we develop our intuition with the case of bivariate balanced sampling and discuss the trick of using a unique uniform at all sites. After that we formalize the problem and present a heuristic local balanced sampling algorithm.

4.2.1 Balanced sampling

In the context of PF one is interested in (re)sampling N analysis particles from an ensemble of N background particles, where each of them has a different probability of being chosen. The first idea that comes to mind is to sample the particles independently with replacement, like from a multinomial distribution. However, such a sampling scheme leads to additional Monte Carlo error due to variability between samples. Especially in applications where the number of particles is limited, making sure that the resampled particles represent well the target distribution is of great importance.

A balanced sampling scheme produces a random sample from the target distribution, while ensuring that the discrepancy between the expected and actual number of particles from each component is at most 1. Let us say we have N particles with probabilities of being resampled $(\alpha_1, \alpha_2, \ldots, \alpha_N)$. We call N_i the number of time that particle i is resampled: its multiplicity. Then balanced sampling is a sampling scheme such that:

1. $\sum_{i=1}^{N} N_i = N$
2. $E(N_i) = N\alpha_i, \quad i = 1, \ldots, N$
3. $|N_i - N\alpha_i| < 1, \quad i = 1, \ldots, N$

There are various ways to produce such a balanced sample but we describe here one simple solution. The first step is to set for sure each N_i to the integer part of $N\alpha_i$, which we denote by $[N\alpha_i]$. Then we add 1 to it with the remaining probability, which is the remaining fractional part $r_i = N\alpha_i - [N\alpha_i]$. In this way N_i can be written as:

$$N_i = [N\alpha_i] + Z_i, \tag{4.7}$$

where Z_i is a Bernoulli random variable with $P(Z_i) = r_i$.

However this has to be done jointly for all particles in order to ensure condition 1. Given a uniform random number U, such a sample can be produced by choosing multiplicities as:

$$N_j = \left| [N \sum_{i=1}^{j-1} \alpha_i + U, N \sum_{i=1}^{j} \alpha_i + U) \cap \{i, 2, \ldots, N\} \right| \tag{4.8}$$

Even though this expression can seem a bit confusing at first, it is actually not so complicated. A simple way to understand how it works is to think of the target distribution as a mixture of dirac measures at the particles with state $x_i = i$, whose cumulative distribution is

$$F(x) = P(X \leq x) = \sum_{i=1}^{N} \alpha_i \mathbf{1}_{(x \leq i)}.$$

A typical way to generate a random variable x from this distribution is to use a uniform U and set $x = F^{-1}(U)$. Applying this sampling N times independently produces an independent sample with replacement. Instead of using N uniforms U_i, one could instead spread them evenly between 0 and 1. The scheme in Eq. (4.8) does exactly this, but adds a random shift U to the evenly spread uniforms to guarantee condition 2.

An illustration of the method is shown in the left panel of Fig. 4.7. Like in the wheel of fortune, the sampling is done by spinning the outer wheel. The length of the segments on the outer wheel are proportional to $N\alpha_i$. When the wheel stops the sample is determined by the ticks on the inner wheel: the segment toward which the ticks point indicates which particles are resampled. On the figure for example, particles 1, 3, 4 and 6 are sampled twice, while particles 2, 5, 7 and 8 are skipped. A different realization of the sampling can be produced by spinning the wheel again. With the help of this illustration it is easy to see that particle i will be sampled at least $[N\alpha_i]$ times. Sometimes one more tick will fall on the segment, sometimes not, depending on the size of the remaining fractional part r_i.

4.2.2 Localization

In large-scale environmental applications such as data assimilation for weather forecast, one possibility to reduce the size of the problem is to restrict the dependencies between dimensions to some local neighborhood. If one tries to apply a particle filter like method in a localized way, this

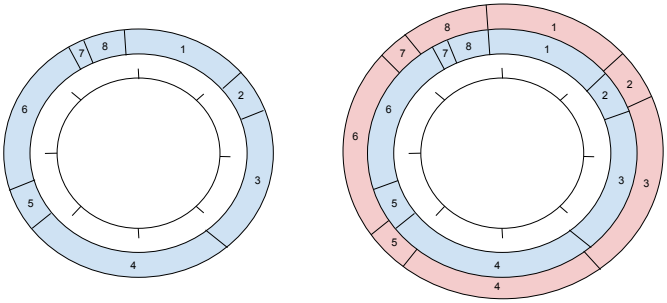

Figure 4.7: Illustration of balanced sampling at one site on the left, and at two sites on the right. In the outer wheels the length of each segment is equal to $N\alpha_j$. The ticks on the inner wheel determine which particles are resampled. The use of a uniform shift U is equivalent to spinning the wheel. On the right, using the same uniform for both sites means that their first weights are aligned.

means resampling independently particles in different locations, or sites. When different particles are sampled at neighboring sites, this create sharp discontinuities, which are very detrimental to the quality of the analysis. Therefore it is highly desirable to introduce some dependency between neighboring sites to limit the number of such discontinuities to a minimum.

4.2.3 Unique uniform trick

Balanced sampling at one site requires the use of one uniform U. An initial attempt to reduce discontinuities was made by simply using the same uniform at all sites. This solution reduces some discontinuities, but it does not optimally solve the problem. Indeed if one pictures the weights α_i at two sites stacked together side by side, one can see that if the weights sequence is more or less similar (as it should be expected for sites which are neighbors in model space) then the use of the same uniform will ensure that some of the resampled particles are the same, but differences will accumulate and at the end errors will be made.

This problem is best illustrated by the same wheel of fortune example. We take two neighboring sites and put them on the same *wheel*, the first one in blue like before, and the second one in red. The situation is depicted on the right panel of Fig. 4.7. The weights are aligned for the first particle, but then discrepancies between the weight sequences accumulate and lead to

bigger and bigger differences. Using the same uniform at both sites, which is equivalent to using only one central wheel and aligning the first weights, makes the beginning of the sample similar, but not toward the end. In this particular case the sample at site 1 (blue) would be $(1, 1, 3, 3, 4, 4, 6, 6)$, whereas at site 2 (red) it would be $(1, 1, 3, 3, 4, 5, 6, 7)$. Here we have only 8 particles, but it is easy to see that the problem would be magnified with a bigger sample size.

Using the same uniform everywhere is therefore rather a trick than a solution to the problem. Although it is not optimal, it has the non-negligible advantage to be extremely simple and can be implemented at virtually no cost in a real-world application.

4.2.4 Formalization of the problem

We now formalize the problem and characterize the desired solution. The setup is as above, but everything is indexed by site s. In particular the probability of resampling particle i at site s is denoted as α_i^s, and its chosen multiplicity as N_i^s. In addition, we define a neighborhood relation $\sim$ on the domain S such that $s \sim s'$ means that s and s' are neighboring sites. Therefore, the goal of minimizing the number of discontinuities can be formalized as minimizing the following for all pairs of neighboring sites in the domain S and all particles $i = 1, \ldots, N$:

$$E\Big(|N_i^s - N_i^{s'}|\Big). \tag{4.9}$$

We want to minimize the number of jumps while keeping the marginal distribution at each site correct (respecting the three conditions for balanced sampling above). Our goal is then to find a joint structure which minimizes the number of discontinuities while maintaining the correct marginals.

We define as $s \overset{i}{\sim} s'$ two sites which are neighbors and satisfy additionally $[N\alpha_i^s] = [N\alpha_i^{s'}]$. Minimizing Eq. (4.9) while satisfying conditions 2 and 3 above is then equivalent to the following additional condition:

4. $\alpha_i^s \leq \alpha_i^{s'} \Rightarrow N_i^s \leq N_i^{s'}, \quad \forall s, s'$ such that $s \overset{i}{\sim} s'$.

Explanation

By symmetry we can consider the case where $\alpha_i^s \geq \alpha_i^{s'}$. Then using the decomposition in Eq. (4.7), the absolute difference between the multiplicities

for particle i at both sites can be splitted in two cases as follows:

$$|N_i^s - N_i^{s'}| = \begin{cases} |Z_i^s - Z_i^{s'}| & \text{if } [N\alpha_i^s] = [N\alpha_i^{s'}] \\ [N\alpha_i^s] - [N\alpha_i^{s'}] + Z_i^s - Z_i^{s'} & \text{otherwise} \end{cases}$$

In the latter case, the first difference is at least one by assumption, and the second part coming from the random resampling of the Z can only be -1, 0 or +1, that is $N_i^s \leq N_i^{s'}$ always holds. Let us compute the expected value of this expression:

$$E\Big(|N_i^s - N_i^{s'}|\Big) = \begin{cases} P(Z_i^s \neq Z_i^{s'}) & \text{if } [N\alpha_i^s] = [N\alpha_i^{s'}] \\ [N\alpha_i^s] - [N\alpha_i^{s'}] + P(Z_i^s) - P(Z_i^{s'}) & \text{otherwise} \end{cases}$$

In the first case the expected value depends on the joint probability of Z_i^s and $Z_i^{s'}$. If they are the same, a discontinuity is avoided, whereas if they are different, a jump will be created between the two sites: therefore, the expected number of discontinuities can be minimized by minimizing $P(Z_i^s \neq Z_i^{s'})$. In the second case, however, the expected value depends on the marginal distribution of $P(Z_i^s, Z_i^{s'})$ only. In other words, without modifying the marginal distributions, which would contradict condition 2), it is possible to minimize the number of discontinuities in the first case, but not in the second.

Therefore, minimizing the expected number of discontinuities happens when for all pairs of sites and particle indices such that $s \overset{i}{\sim} s'$, the multiplicity of the site with the bigger α_i must be bigger or equal than the one with the smaller α_i; or vice and versa, as in condition 4. To see that more clearly, we consider the problem of coupling two Bernoulli random variables in the next paragraph.

Coupling of two Bernoulli random variables

We describe quickly how to maximize the coupling of two Bernoulli random variables, in our case Z_i^s and $Z_i^{s'}$, while respecting their marginal distributions, α_i^s and $\alpha_i^{s'}$. We assume that $\alpha_i^s \leq \alpha_i^{s'}$. One can find that the probability that $Z_i^s = 1$ and $Z_i^{s'} = 1$ should equal $min(\alpha_s^i, \alpha_{s'}^i) = \alpha_i^s$, and similarly that the probability that $Z_i^s = 0$ and $Z_i^{s'} = 0$ should equal $1 - \alpha_i^{s'}$. The probabilities $P(Z_i^s = 1, Z_i^{s'} = 0)$ and $P(Z_i^s = 0, Z_i^{s'} = 1)$ can then be found easily from the conditions on the marginals.

		Z_i^s		
		0	1	Sum
$Z_i^{s'}$	0	$1-\alpha_i^{s'}$	0	$1-\alpha_i^{s'}$
	1	$\alpha_i^{s'}-\alpha_i^s$	α_i^s	$\alpha_i^{s'}$
	Sum	$1-\alpha_i^s$	α_i^s	1

Table 4.1: Joint probability distribution of Z_i^s and $Z_i^{s'}$ with maximal coupling where $\alpha_i^s \leq \alpha_i^{s'}$ and $s \overset{i}{\sim} s'$.

We summarize this joint distribution with maximal coupling in Table 4.1. From this table one can see that the probability to get $Z_i^s = 1$ and $Z_i^{s'} = 0$ is zero. This ensures the fulfillment of condition 4, that the multiplicity of the site with the smaller probability (here site s by assumption) cannot be bigger than the multiplicity at the other site. A simple way to sample from this distribution is to generate one uniform U, and set Z_i^s to 1 if $U <= \alpha_i^s$, and $Z_i^{s'}$ to 1 if $U <= \alpha_i^{s'}$. In other word, for two Bernoulli random variables the unique uniform trick gives the optimal coupling.

4.2.5 Local balanced sampling algorithm

So far we described the conditions that an optimal local balanced sampling schemes should fulfill, and gave some explanations and justifications for them. However, until now we did not consider how one would obtain such a sample. The goal is to sample N particles, at all sites in the domain S, such that the samples are balanced at each site (conditions 1 to 3), and that the number of discrepancies between the multiplicities of neighboring sites is minimized (condition 4). Finding the global optimum to this problem is not feasible, so we consider an approximate sequential scheme.

First let us define a i-chain as a subset of S that contains at least two points and forms a connected component w.r.t. $s \overset{i}{\sim} s'$. We can then sample all the Z_i^s of the i-chain at once by using the same uniform everywhere, which guarantees that condition 4 is fulfilled.

In the global balanced sampling scheme, all the particles are sampled at once using a unique uniform, which guarantees conditions 1 to 3. In the local scheme, however, we sample *one* particle in many sites at the same time, such that condition 4 is satisfied. If we would do that independently for each particle, there would be no guarantee that the other conditions

for balanced sampling are also satisfied. Therefore, we need to update the probabilities at each step accordingly. We will discuss how to do so next, but we first summarize the sequential update scheme:

1. Find all i-chains.
2. Select the largest i-chain in the domain S.
3. Generate $U \sim Unif$.
4. Set $N_i^s = [N\alpha_i^s] + 1_{[U<r_i^s]}$ at all sites of the i-chain.
5. Update the remaining probabilities at all sites.
6. Find all i-chains using the new updated probabilities.
7. Go to 2

If finding all i-chains is too expensive to perform, a simplified version of the algorithm could instead pick the first i-chain it finds, for example by starting at a random location and particle, and search adaptively among neighbors for the connected component. The solution would probably be suboptimal, but the computational gain might well compensate for it.

On the other hand, if computational cost is not an issue, we could replace step 2 by the following: instead of selecting the largest i-chain, select the i-chain that maximizes the gain of coupling over independent sampling. The expected number of discrepancies in a i-chain is given by $\sum P(Z_i^s \neq Z_i^{s'})$ for all sites in the chain. In the case of independent sampling this is equal to $\sum r_i^s(1-r_i^{s'}) + (1-r_i^s)r_i^{s'}$, while in the case of coupling with the same uniform this is equal to $\sum |r_i^s - r_i^{s'}|$. Therefore one can select the i-chain that maximize the difference between this two terms.

We now discuss step 5, how to update the probabilities r_i after sampling an i-chain.

4.2.6 Update of probabilities

In the previous section we described how one could produce a balanced sample which minimizes discontinuities between neighboring sites. However this requires to produce the sample sequentially and thus to update the probabilities at each step when sampling occurs at a site. To describe more clearly how to do so, we first restate what we have at our disposal, and introduce and few more elements. We consider the update at one site at a time and thus we can skip the superscript s for ease of notation.

Remember from Eq. (4.7) that we have the remainders r_i which determine the probabilities to add 1 to the multiplicity of particle i, or in other word $P(Z_i = 1) = r_i$. Depending on N and the probabilities α_i, the total number of particles which need to be chosen in this way might vary. We denote this number as $d = \sum_{i=1}^{N} r_i = \sum_{i=1}^{N} Z_i$.

Let us assume that the multiplicity for the first particle has been chosen, and thus that Z_1 is set to 0 or 1. We can then define the desired updated probabilities as:

$$r_{i+} = P(Z_i = 1 | Z_1 = 1), \tag{4.10}$$

$$r_{i-} = P(Z_i = 1 | Z_1 = 0). \tag{4.11}$$

If Z_i and Z_1 could be considered independent, then $r_{i+} = r_{i-} = r_i$. However, because of condition 1, the Z_i have to be dependent, and the conditionals above have to be different from the marginals. In particular, we see that to ensure that the total number of particles N stay correct, these probabilities must satisfy:

$$\sum_{i=2}^{N} r_{i+} = E\Big(\sum_{i=2}^{N} Z_i \Big| Z_1 = 1 \Big) = d - 1, \tag{4.12}$$

$$\sum_{i=2}^{N} r_{i-} = E\Big(\sum_{i=2}^{N} Z_i \Big| Z_1 = 0 \Big) = d. \tag{4.13}$$

We thus have some freedom in the way we set r_{i+} and r_{i-}, as long as we satisfy the above conditions, and of course that we have valid probabilities such that $0 \leq r_{i+}, r_{i-} \leq 1$. To fulfill the conditions above one need to *renormalize* the probabilities such that they sum to the correct amount. From $\sum_{i=2}^{N} r_i = d - r_1$, we can find the following simple solution:

$$r_{i+} = r_i \cdot \frac{d-1}{d-r_1} \quad \text{and} \quad r_{i-} = r_i \cdot \frac{d}{d-r_1}. \tag{4.14}$$

Part II

Ensemble space

Chapter 5

The local ensemble transform Kalman particle filter

The content of this chapter has been submitted as the first part of Robert et al. (2017). Sections 5.1 and 5.2 repeat some things that have been discussed in the first chapter of this thesis, but have been reproduced here for completeness. In Section 5.3 the new LETKPF is presented, with some more detail about the derivations in the Appendixes 5.A and 5.B. In Chapter 6 we will discuss numerical experiments with the COSMO model.

5.1 Introduction

Probabilistic weather forecasts are superior to deterministic ones for a wide range of applications, such as evaluating weather-related risks or managing renewable energy production. A key element of probabilistic weather forecasting is the use of ensembles methods: instead of running one highly accurate prediction, we can produce an ensemble of typically 5 to 100 forecasts, which provides information not only on the most probable evolution of the atmosphere, but also on the associated uncertainties. Producing such probabilistic forecasts from ensembles is a complex endeavor involving quantification of initial conditions' uncertainties, representation of model

errors, post-processing of ensembles, bias corrections, etc (Gneiting and Raftery, 2005). In the present paper, we focus on the production of initial conditions with ensemble data assimilation methods, which combine information coming from the previous weather forecast with the stream of incoming observations.

Benefiting from increasing computational resources, regional weather forecast models nowadays run with a very high spatial resolution (1 to 5 kilometers), which allows them to resolve small-scales dynamical effects, for example convection (Harnisch and Keil, 2015). On the one hand this is an advantage, as it provides forecasts of high-impact weather events such as heavy storms, but on the other hand it makes the task of data assimilation much harder due to the intrinsic non-linearity of the phenomena resolved at those scales (Bauer et al., 2015). Indeed, strong non-linearities lead to non-Gaussian uncertainties in the initial conditions, which current methods are poorly equipped to deal with. Therefore, there is a growing need for computationally efficient ensemble data assimilation algorithms able to handle non-linearities and non-Gaussian distributions.

Ensemble data assimilation methods are sequential algorithms which alternate between two steps. First, during the forecast step, they propagate the ensemble of particles from the previous iteration through the dynamical system, which produces a so-called *background* ensemble. Then, during the update step, or *analysis*, they use the newly available observations to modify the ensemble of particles and produce a so-called *analysis* ensemble. The various assimilation methods typically differ in the way they implement the analysis. The current state-of-the-art ensemble methods are based on the ensemble Kalman filter (EnKF) (Evensen, 1994, 2009), which conducts the analysis by moving the particles towards the observations in a way that relies implicitly on Gaussian assumptions. Particle filters (PFs), on the other hand, directly implement Bayes' formula for the analysis without relying on any Gaussian assumptions (Gordon et al., 1993; Pitt and Shephard, 1999; Doucet et al., 2001). However, this flexibility comes at a cost, and while EnKFs are highly efficient and used in practice, PFs are prohibitively expensive to implement, as they need a very large number of particles to work in high-dimensional systems such as in numerical weather prediction (see Snyder et al. (2008) for more details on the limits of PFs in high-dimensions).

Adapting the PF to high-dimensional applications is an active field of research and there have been many propositions of new algorithms, which can be broadly categorized in three different approaches. The first one is

to use variants of the PF with different proposal distributions (Pitt and Shephard, 1999; van Leeuwen, 2010; Ades and van Leeuwen, 2013). The second is to to create hybrid methods which somehow combine the PF with the EnKF (Frei and Künsch, 2013; Reich, 2013b). The last approach is to localize the PF, which is difficult but might be the only viable solution for very high-dimensional systems (Poterjoy, 2016; Robert and Künsch, 2017a; Snyder et al., 2015; Rebeschini and Handel, 2015). In the present paper we focus on methods which combine the hybrid algorithm approach with localization (see for example Robert and Künsch (2017b) or Chustagulprom et al. (2016)).

Promising results with PFs have been reported on various small- to medium -scale toy models, but so far the only application to full-scale weather prediction system that we are aware of is Poterjoy and Anderson (2016). Here we describe a newly developed localized hybrid algorithm based on the ensemble Kalman particle filter (EnKPF) of Frei and Künsch (2013). We implemented it in the assimilation framework of the COSMO (Consortium for Small-scale Modeling) model (Baldauf et al., 2011), and we ran successful experiments within the operational data assimilation system of MeteoSwiss. The implementation of our algorithm was made possible thanks to a collaboration with the Deutscher Wetter Dienst, which is also working on PFs for data assimilation.

A key development of the new algorithm, called the local ensemble transform Kalman particle filter (LETKPF), consisted in formulating the EnKPF in ensemble space, from which we could derive a computationally efficient implementation and a deterministic, or *transform*, analysis scheme. While other localization methods might be theoretically better (Robert and Künsch, 2017b), we used the scheme of the local ensemble transform Kalman filter (LETKF) (Hunt et al., 2007) for ease of implementation, as it is the assimilation algorithm used by COSMO. A critical aspect of hybrid methods is to choose the balance between the EnKF and the PF, which is represented by the parameter γ in the EnKPF. We proposed and explored a new objective criterion to choose this parameter γ adaptively in space and time and compared it to the standard approach.

We conducted numerical experiments with a convective-scale regional model for a period of 12 days in June 2015, with a setup similar to the one used operationally at MeteoSwiss. The new algorithm is shown to perform at a similar level to the LETKF, with some noticeable improvements for non-Gaussian variables such as wind and hourly precipitation. These results are very promising for the future of localized hybrid algorithms in chal-

lenging real-world applications and we hope that they will spark further interest in our algorithm.

In Section 5.2 we review ensemble data assimilation EnKPF. In Section 5.3 we derive the new LETKPF algorithm in ensemble space, describe how to compute it efficiently, and discuss how to localize the analysis and choose the parameter γ adaptively. In Chapter 6 we present experimental results with the COSMO model.

5.2 Background

The uncertainty about the q-dimensional state x_t of a dynamical system based on a stream of observations is best described by probability distributions: The background or forecast distribution $\pi_t^b(x_t)$ is based on observations before time t whereas the analysis distribution $\pi_t^a(x_t)$ includes in addition the current observation y_t according to Bayes' formula: $\pi_t^a(x_t) \propto \pi_t^b(x_t) \cdot \ell_t(x_t|y_t)$ where $\ell_t(x_t|y_t)$ is the likelihood of x_t if y_t has been observed.

Ensemble methods represent these distributions with finite samples of k particles, $\{x_t^{b,i}\}$ and $\{x_t^{a,i}\}$. These particles are propagated and updated sequentially: Propagating $\{x_{t-1}^{a,i}\}$ according to the dynamics of the system produces $\{x_t^{b,i}\}$, updating $\{x_t^{b,i}\}$ by a sampling version of Bayes' theorem produces $\{x_t^{a,i}\}$. Different analysis algorithms vary in the assumptions they make about $\pi_t^b(x_t)$ and $\ell_t(x_t|y_t)$, and in the sampling version of Bayes' theorem.

In the present paper we focus on a single analysis step and thus omit the time index t. We also assume that the observations are linear and Gaussian with mean Hx and covariance R. We next review the EnKPF algorithm in this context and present the EnKF and the PF as special cases.

The EnKPF introduced in Frei and Künsch (2013) decomposes the analysis into two stages as $\pi^a(x) \propto \pi^b(x) \cdot \ell(x|y)^\gamma \cdot \ell(x|y)^{1-\gamma}$, where $0 \leq \gamma \leq 1$. The core idea of the algorithm is to conduct the first part of the analysis with an EnKF using the dampened likelihood $\ell(x|y)^\gamma$, and then to apply a pure PF to the remaining likelihood $\ell(x|y)^{1-\gamma}$.

The first part of the analysis implicitly relies on Gaussianity of the background distribution, but the second part does not make any assumption. The EnKPF can thus adapt to some non-Gaussian features of the back-

ground distribution without suffering from sample degeneracy like the pure PF. The parameter γ allows one to choose how much of the analysis should be done with the EnKF and how much with the PF, depending on the particular situation at hand.

Using the Gaussian mixture representation of the analysis distribution after the EnKF step, it is possible to derive the final analysis distribution as the following Gaussian mixture:

$$\pi^a_{EnKPF}(x) = \sum_{i=1}^{k} \alpha^{\gamma,i} \mathcal{N}(\mu^{\gamma,i}, P^{a,\gamma}), \tag{5.1}$$

whose component means $\mu^{\gamma,i}$, mixing weights $\alpha^{\gamma,i}$ and component covariance $P^{a,\gamma}$ are defined as:

$$\begin{aligned}
\mu^{\gamma,i} &= \nu^{\gamma,i} + K((1-\gamma)Q)(y - H\nu^{\gamma,i}), \\
P^{a,\gamma} &= \big(I - K((1-\gamma)Q)H\big)Q, \\
\alpha^{\gamma,i} &\propto \phi\{y;\ H\nu^{\gamma,i}, HQH' + R/(1-\gamma)\},
\end{aligned}$$

where $\nu^{\gamma,i}$ and Q are intermediary quantities from the EnKF step derived from the background particles $x^{b,i}$ and background covariance matrix P^b as

$$\begin{aligned}
\nu^{\gamma,i} &= x^{b,i} + K(\gamma P^b)(y - Hx^{b,i}) \quad \text{and} \\
Q &= \frac{1}{\gamma} K(\gamma P^b) R K(\gamma P^b)'.
\end{aligned}$$

$K(P)$ denotes the Kalman gain computed using the covariance matrix P and is equal to $PH'(HPH'+R)^{-1}$, while $\phi\{y; \mu, P\}$ denotes the density of a Gaussian distribution with mean μ and covariance matrix P evaluated at y. More details about the derivation of the EnKPF algorithm can be found in Frei and Künsch (2013) and Robert and Künsch (2017b).

It is convenient for later derivations to rewrite the expression for the $\mu^{\gamma,i}$ components directly from the background ensemble as:

$$\mu^{\gamma,i} = x^{b,i} + L^\gamma(y - Hx^{b,i}), \quad \text{where} \tag{5.2}$$

$$L^\gamma = K(\gamma P^b) + K((1-\gamma)Q)\Big(I - HK(\gamma P^b)\Big). \tag{5.3}$$

L^γ is the composite Kalman gain resulting from the successive application of the EnKF and PF. It plays a similar role to the Kalman gain, but it

should be noted that there is no estimate of the background covariance P^b such that a pure EnKF would have this gain.

Sampling from Eq. (5.1) can be done by first sampling the indicators $I(i)$ of the mixture components according to $P(I(i) = j) = \alpha^{\gamma,j}$ and then adding an artificial noise $\epsilon^i \sim \mathcal{N}(0, P^{a,\gamma})$ to $\mu^{\gamma,I(i)}$:

$$x^{a,i} = \mu^{\gamma,I(i)} + \epsilon^i. \tag{5.4}$$

Instead of sampling with replacement from the set of indices, one can generate the indicators $I(i)$ by a balanced sampling scheme which guarantees that N^j, the multiplicity or number of times a particle j is selected, is less than one unit away from its expected value, i.e. $|N^j - k\alpha^{\gamma,j}| < 1$ (for more details on balanced sampling see for example Carpenter et al. (1999), Crisan (2001) or Künsch (2005)).

The EnKF and the PF can be seen as special cases of the EnKPF. Setting γ to 1 we find

$$\begin{aligned}
\alpha^{1,i} &\propto 1, \\
\mu^{1,i} &= x^{b,i} + K(P^b)(y - Hx^{b,i}), \\
P^{a,1} &= K(P^b)RK(P^b)'.
\end{aligned}$$

A balanced sampling scheme, therefore, selects each index exactly once, and thus we recover the stochastic version of the EnKF. At the other end of the spectrum, setting γ to 0 we find

$$\begin{aligned}
\alpha^{0,i} &\propto \phi(y;\ Hx^{b,i}, R), \\
\mu^{0,i} &= x^{b,i}, \\
P^{a,0} &= 0.
\end{aligned}$$

The analysis ensemble is thus a resample of the background ensemble with weights proportional to the likelihood, and we recover the PF. For $\gamma > 0$, the artificial noise ϵ^i is not zero and thus no two analysis particles are exactly the same, which is one of the drawbacks of the PF.

5.3 Derivation of the LETKPF

When the number of particles k is much smaller than the dimension q of the system, it is desirable that the analysis ensemble belongs to the ensemble space, i.e. the $(k-1)$-dimensional hyperplane in $\mathbb{R}^q$ spanned by the

background ensemble. This has advantages both for efficient implementation and for stability of the assimilation scheme, since the ensemble space usually contains the main directions of instability.

In the following we represent the background and analysis ensembles as $q \times k$ matrices $\boldsymbol{x}^b$ and $\boldsymbol{x}^a$ such that each column is one ensemble member. The analysis ensemble belongs to the ensemble space if

$$x^{a,i} = \sum_{j=1}^{k} x^{b,j} W_{ji}, \qquad \text{with } \sum_j W_{ji} = 1.$$

Equivalently, if and only if the analysis belongs to the ensemble space, it can be expressed as:

$$\boldsymbol{x}^a = \bar{x}^b \mathbf{1}' + X^b W, \tag{5.5}$$

where $\mathbf{1}$ denotes the vector of length k with all elements equal to 1, $X^b = \boldsymbol{x}^b - \bar{x}^b \mathbf{1}'$ the $q \times k$ matrix of deviations from the background mean, and W is a $k \times k$ weight matrix. Because X^b does not have full rank, we do not need to impose the condition $\sum_j W_{ji} = 1$.

In order to implement the EnKPF we have to estimate the background covariance P^b. Using the sample covariance matrix

$$P^b = \frac{1}{k-1} X^b (X^b)',$$

the resulting analysis is in ensemble space and can be expressed in the form of Eq. (5.5). To prove this and to derive the corresponding W matrix, we first pull out a factor X^b from the matrix L^γ defined in Eq. (5.3)

$$L^\gamma = X^b \tilde{L}^\gamma.$$

From Eq. (5.2) it then follows that the ensemble of component means $\mu^{\gamma,i}$ of Eq. (5.1) is automatically in ensemble space:

$$\boldsymbol{\mu}^\gamma = \bar{x}^b \mathbf{1}' + X^b W^\mu, \quad W^\mu = I + \tilde{L}^\gamma (y\mathbf{1}' - H\boldsymbol{x}^b).$$

Resampling of the component means can be described by multiplying $\boldsymbol{\mu}^\gamma$ from the right with the matrix W^α, which has exactly one 1 in each column, indicating which particle is resampled, or more precisely

$$W^\alpha_{ij} = \begin{cases} 1 & \text{if } I(j) = i, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, the analysis ensemble from Eq. (5.4) lies in ensemble space if the matrix of perturbations $\boldsymbol{\epsilon}$ from Eq. (5.4) can be expressed as $X^b W^\epsilon$:

$$\boldsymbol{x}^a = \bar{x}^b \mathbf{1}' + X^b (W^\mu W^\alpha + W^\epsilon). \tag{5.6}$$

If we estimate the background covariance by the sample covariance, we can pull out a factor X^b on both sides of $P^{a,\gamma}$:

$$P^{a,\gamma} = X^b \tilde{P}^{a,\gamma} (X^b)'.$$

Hence in a stochastic version of the filter, we could generate W^ϵ as follows

$$W^\epsilon = \left(\tilde{P}^{a,\gamma}\right)^{1/2} E, \tag{5.7}$$

where E is a $k \times k$ matrix of centered *i.i.d.* samples from a standard normal and $(\cdot)^{1/2}$ is any matrix square-root. Then $\boldsymbol{\epsilon} = X^b W^\epsilon$ has exactly mean zero and covariance $P^{a,\gamma}$.

Instead of using a random draw for the added perturbations we would like to use a deterministic scheme for producing $\boldsymbol{\epsilon}$. The first idea that comes to mind is to redefine W^ϵ in Eq. (5.7) as the symmetric matrix square-root of $(k-1)\ \tilde{P}^{a,\gamma}$, because then $\boldsymbol{\epsilon}$ has exactly covariance $P^{a,\gamma}$. However, using such a scheme results in an analysis ensemble with the wrong covariance, because the W^ϵ generated in this way is strongly correlated with the W^μ matrix and their effects tend to cancel each other. For the stochastic version of the filter this problem is not present because the samples E in Eq. (5.7) are independent of the background ensemble. However, for a deterministic filter we need to take these correlations explicitly into account and match the first and second moments of the analysis ensemble with their expected values.

The analysis mean, $\bar{x}^a$, should be equal to the mean of the resampled component means

$$\bar{\mu}^\gamma = \sum_{i=1}^{k} \frac{N^i}{k} \boldsymbol{\mu}^{\gamma,i}. \tag{5.8}$$

Noticing that $\bar{x}^b + 1/k \cdot X^b W^\mu W^\alpha \mathbf{1} = \bar{\mu}^\gamma$, it is clear that for $\bar{x}^a$ to equal $\bar{\mu}^\gamma$, $W^\epsilon \mathbf{1}$ must equal $\mathbf{0}$. In other words, the added perturbations should have mean zero. For the stochastic W^ϵ defined in Eq. (5.7) this holds because the matrix E is centered such that $E\mathbf{1} = \mathbf{0}$.

The covariance of $\boldsymbol{x}^a$ should be equal to the covariance of the resampled

component means plus the component covariance:

$$P^\gamma = \sum_{i=1}^{k} \frac{N^i}{k-1} (\boldsymbol{\mu}^{\gamma,i} - \bar{\mu}^\gamma)(\boldsymbol{\mu}^{\gamma,i} - \bar{\mu}^\gamma)' + P^{a,\gamma}. \tag{5.9}$$

Computing everything in ensemble space, we can find that for the covariance of $\boldsymbol{x}^a$ to equal P^γ, the matrix W^ϵ must satisfy the equation

$$A(W^\epsilon)' + W^\epsilon A' + W^\epsilon (W^\epsilon)' = (k-1)\tilde{P}^{a,\gamma}, \tag{5.10}$$

where A is the centered matrix

$$A = W^\mu W^\alpha - \frac{1}{k} W^\mu W^\alpha \mathbf{1}\mathbf{1}'.$$

This is a special form of a continuous algebraic Riccati equation, or CARE. In general, it has infinitely many solutions. In our experience, requiring W^ϵ to be symmetric and positive definite leads to good properties of the analysis. Such a solution exists and it can be found efficiently using Newton's method. Moreover, because of special properties of the matrices involved, it can be shown that this solution of Eq. (5.10) guarantees a correct first moment with $W^\epsilon \mathbf{1} = \mathbf{0}$. Details about the algorithm to solve W^ϵ and the latter property are given in the Section 5.A. A related algorithm which solves a CARE to obtain an analysis ensemble with correct covariance is described in de Wiljes et al. (2016).

We have thus found a deterministic version of the EnKPF in ensemble space, which we call the ETKPF by analogy with the ETKF, which it is equivalent to when $\gamma = 1$. It should be noted that when γ is not equal to 1 the solution found by the ETKPF is not the same as simply taking the symmetric square-root of the Gaussian mixture covariance. Indeed, the square-root scheme is only used as a correction term to the analysis ensemble, similarly to the random perturbations added in the stochastic EnKF. In particular the resampling step ensures that interesting non-Gaussian properties of the analysis distribution are represented in the ensemble.

5.3.1 Efficient computation

In principle there are different ways to compute W efficiently, but we chose to follow the procedure of the ETKF as closely as possible for easy implementation in the COSMO data assimilation framework. The starting

point is to compute the spectral decomposition of S, the weighted covariance matrix of the deviations of the model equivalents HX^b in ensemble space, or more precisely:

$$S = (HX^b)'R^{-1}(HX^b) = U\delta(\boldsymbol{\lambda})U', \tag{5.11}$$

where U is the matrix of eigenvectors and $\delta(\boldsymbol{\lambda})$ denotes the diagonal matrix constructed with the vector of eigenvalues $\boldsymbol{\lambda}$. Because X^b is centered, 0 is an eigenvalue of S with eigenvector $\mathbf{1}$. If the number of observations d is larger than k, S typically has $(k-1)$ non-zero eigenvalues.

The influence of the observations enters through the following vector:

$$c = (HX^b)'R^{-1}(y - H\bar{x}^b). \tag{5.12}$$

Using Woodbury's formula multiple times and working out the algebra, it is possible to compute W from these elements. For W^μ we obtain the following expression:

$$W^\mu = U\delta(f^\mu(\boldsymbol{\lambda}))U' + U\delta(f^{\bar{\mu}}(\boldsymbol{\lambda})))U'c\mathbf{1}', \tag{5.13}$$

where f^μ and $f^{\bar{\mu}}$ are rational functions and $f(\boldsymbol{\lambda})$ denotes the vector with components $f(\lambda_i)$. More details about the derivation of this and the following expressions and explicit formulas can be found in Section 5.B.

The matrix W^α does not have to be constructed explicitly, only the weights $\alpha^{\gamma,i}$ and the vector of resampled indices I are needed. Going through the algebra, one can find that the weights are proportional to the following expression:

$$\exp\left(-\frac{1}{2}\Big(U\delta(\boldsymbol{\lambda}f^\alpha(\boldsymbol{\lambda}))U'\Big)_{ii} + \Big(U\delta(f^\alpha(\boldsymbol{\lambda}))U'c\Big)_i \right),$$

where f^α is also a rational function.

Both the stochastic EnKPF and the ETKPF need the ensemble space covariance $\tilde{P}^{a,\gamma}$ to be computed. Similarly to the calculation of W^μ one can find that

$$\tilde{P}^{a,\gamma} = U\delta(f^\gamma(\boldsymbol{\lambda}))U',$$

where f^γ is another rational function. For the stochastic EnKPF, the symmetric matrix square root can thus be computed easily as:

$$\Big((k-1)\ \tilde{P}^{a,\gamma}\Big)^{1/2} = \sqrt{k-1}\cdot U\delta(\sqrt{f^\gamma(\boldsymbol{\lambda})})U'.$$

For the ETKPF one still needs to solve the CARE of Eq. (5.10), which is described in Section 5.A, but all its elements can be computed efficiently from the above expressions. From these equations we can recover the special cases of the ETKF and the PF in the limit $\gamma \to 1$ and $\gamma \to 0$. Details are given in Section 5.B.

5.3.2 Localization

If the ensemble size is much smaller than the system dimension, all methods described so far perform poorly. The EnKF suffers from spurious long range correlations that result from low rank background covariances. With PFs the problem is even more pronounced, as the ensemble collapses if the number of particles does not grow exponentially with the problem size (see Snyder et al. (2008) for more detail). These problems can be overcome by localization, which essentially consists in doing a separate analysis at each site and then *gluing* them together. For the EnKF this is well established, leading to the LETKF and similar methods, However it is not straightforward to use localization for PFs because of discontinuities introduced by resampling different particles at neighboring sites. We now discuss how we address these issues with the EnKPF, which leads to the LETKPF.

The basic idea of localization is to compute different W matrices at every site. For the EnKPF, the W^μ matrices associated with the component means of the analysis cause no problem as they vary smoothly between adjacent sites, provided that the localization radius is sufficiently large. In practice, one further enforces smooth transitions by tapering the inverse of the observation covariance matrix R^{-1} as a function of distance. For the EnKPF, however, the biggest issue comes from the resampling matrix W^α and the perturbation matrix W^ϵ. The problems with the latter are relatively easy to be dealt with, but the ones with the former can only be partially addressed.

In the case of the stochastic EnKPF one simply uses the same noise matrix E to construct W^ϵ in Eq. (5.7) at every site. Because the covariance matrix $\tilde{P}^{a,\gamma}$ varies smoothly in space, the W^ϵ matrix constructed in this way does not introduce additional discontinuities. For the ETKPF there is nothing special to do as the algorithm to find W^ϵ is deterministic and its solution varies smoothly between sites.

The main problem comes from the resampling matrix W^α, which reflects

the PF part of the algorithm. The weights $\alpha^{\gamma,i}$ vary smoothly in space, but the resampling of particles is discrete in nature and can thus vary abruptly from one location to another. We now consider three steps to limit the number of discontinuities introduced in this way.

The first step is to reduce the noise added during the choice of the resampled indices vector I from the weights $\alpha^{\gamma,i}$. Clearly, using independent sampling with replacement would be a very poor choice, as even if two adjacent sites had the exact same weights it would result in very different I vectors. The balanced sampling scheme that we use for choosing I is much better as it ensures that the multiplicities of each particle is at most one unit away from their expected value. A simple way to further reduce the added randomness is to use the same random seed at every site. This solution is still suboptimal, but we cannot do better without global communication between sites, which is prohibitive for high-dimensional applications.

The second step to limit the number of discontinuities is to permute the vector of resampled indices I. Indeed, the indexing of particles is arbitrary and can thus be changed without any influence on the local analysis. Unfortunately, finding the optimal permutation of every local I such that the number of discontinuities is minimal is an optimal assignment problem which cannot be solved without using global communication between sites. However, putting as many 1 as possible on the diagonal of the W^α matrix, and then filling in the remaining cases in a determined order, is simple and reduces discontinuities by a large extent.

The third step to limit the number of discontinuities is to compute the local analysis on a coarse grid and then to interpolate the matrix W to a finer grid. This is routinely done with the LETKF in practice, but for different reasons. In the case of the LETKF the main goal is to reduce the computational cost of the analysis, whereas in our case we want to smooth out discontinuities. Let us say we need to match particle i at one coarse grid point with particle j at the next coarse grid point. By interpolating the weights on the finer grid in between, we obtain particles which mix particles i and j progressively, and thus create a smooth transition between both.

It is worthwhile to mention that not all discontinuities are necessarily bad, and it is easy to imagine cases where they are actually positive. In particular if the physical field of interest is not continuous, such as a cloud field, it makes sense to match together different particles at different sites.

The problems arise when the estimated derivatives in the propagation step become large, which can result in gravity waves or other spurious dynamical effects. The extent to which such harmful discontinuities are avoided with our algorithms needs to be studied in practice.

5.3.3 Adaptive choice of γ

The parameter γ determines the proportion of the analysis done with the EnKF and with the PF. There is no reason to fix it a priori and we would like a criterion to select its value adaptively. Frei and Künsch (2013) proposed to choose the smallest γ such that the equivalent sample size (ESS) (Liu, 1996), computed from the mixture proportions as $1/\sum(\alpha^{\gamma,i})^2$, is within a given bound, for example no less than 50% of the original ensemble size. This idea is reasonable and particularly cheap to implement, but the problem of choosing γ is transfered to the problem of choosing the desired reduction in equivalent sample size, and it does not provide us with a clear criterion for the latter. In Chapter 6 we use this criterion with a targeted ESS of 50% as a reference to which we compare the alternative solution proposed below.

Another approach that seems attractive at first sight is to make γ a function of the "non-Gaussianity" of the distribution. The motivation is that if the background ensemble is truly Gaussian, one should choose $\gamma = 1$ and recover the EnKF, while the more non-Gaussian the distribution, the more PF should be used. However, there are at least two reasons why this idea is not applicable in practice. First, the concept of non-Gaussianity is not well defined, as there are infinitely many ways for a distribution to be non-Gaussian, especially in higher dimensions; but even with a measure of non-Gaussianity, one would still have to map its value to a choice of γ between 0 and 1, for which we would still have no guidance. Second, there are cases where the background distribution is clearly non-Gaussian but it might be preferable to choose a γ close to 1. Indeed, if the observation y is situated outside of the convex hull formed by the ensemble, the weights $\alpha^{\gamma,i}$ will be very skewed and thus lead to sample depletion. In such a case we would be better off choosing a large γ even if the background is non-Gaussian.

To address the various points above we propose to base the choice of γ on the mean squared error (MSE) of the predictive mean of y. From Eq. (5.1)

it follows that the predictive distribution of y is the following mixture:

$$\pi^a(y) = \sum_{i=1}^{k} \alpha^{\gamma,i} \mathcal{N}(H\mu^{\gamma,i}, HP^{a,\gamma}H' + R). \tag{5.14}$$

In order to take into account the error coming from the resampling step, we condition on the multiplicities N^i (the number of times component i is resampled), and consider the following predictive distribution:

$$\pi^a(y|\{N^i\}) = \sum_{i=1}^{k} \frac{N^i}{k} \mathcal{N}(H\mu^{\gamma,i}, HP^{a,\gamma}H' + R), \tag{5.15}$$

whose mean is $H\bar{\mu}^\gamma$ given in Eq. (5.8).

We then choose γ such that the MSE of the predictive mean, $H\bar{\mu}^\gamma$, is minimal. Because the observations do not all have the same variance, it is necessary to scale the MSE with R^{-1}, or more precisely:

$$\mathrm{MSE}(H\bar{\mu}^\gamma, y) = (y - H\bar{\mu}^\gamma)' R^{-1} (y - H\bar{\mu}^\gamma), \tag{5.16}$$

where the predictive mean $H\bar{\mu}^\gamma$ depends on γ. Writing $\bar{\mu}^\gamma$ as $\bar{x}^b + X^b m^\gamma$, where m^γ is the weight vector defined by $\frac{1}{k} W^\mu W^\alpha \mathbf{1}$, the MSE above can be written as:

$$\mathrm{MSE}_\gamma(H\bar{\mu}^\gamma, y) = \mathrm{MSE}(H\bar{x}^b, y) + (m^\gamma)' S m^\gamma - 2(m^\gamma)' c \tag{5.17}$$

where S and c are defined in Eq. (5.11) and Eq. (5.12). Since the first term is independent of γ, we can choose γ adaptively by minimizing $(m^\gamma)' S m^\gamma - 2(m^\gamma)' c$, for example with a grid search.

The scheme for choosing γ proposed above is objective and does not need any additional tuning parameter. On the other hand, it might lead to overfitting as it uses the observations y twice: once for computing the analysis given γ and once for computing the MSE. Practical experiments are needed to evaluate if this is a non-negligible effect. One potential remedy to mitigate the problem is to use the jackknife, a bias reduction technique, to estimate the expected MSE. A more radically different approach would be to use a cross-validation scheme with surrogate data created from the background ensemble. The latter approach is attractive from a theoretical point of view, but it is computationally expensive and implicitly relies on the assumption that the ensemble and the truth are exchangeable, which might be violated in case of systematic model biases.

Instead of the MSE of the analysis mean, we could also use the energy score (ES), a strictly proper multivariate generalization of the continuous ranked probability score (CRPS) (Gneiting and Raftery, 2007). We developed an algorithm to approximate the ES in ensemble space but the resulting choice of γ was not significantly different from using the MSE criterion above, and we thus prefer the latter method for its simplicity. Optimal selection of the parameter γ depends on many different variables such as the number of observations compared to k, the distribution of the background and the assimilation strength, and should be the object of further research.

Appendix 5.A Riccati equation for the transform filter

First let us write Eq. (5.10) replacing W^ϵ with X and $(k-1)\tilde{P}^{a,\gamma}$ with C for more clarity:

$$AX' + XA' + XX' - C = 0, \tag{5.18}$$

where we transposed the first X, which we can do as we seek a symmetric solution. Using Newton's method to solve this equation we find the candidate X_{n+1} recursively by solving

$$(A + X_n)X'_{n+1} + X_{n+1}(A' + X'_n) = X_n X'_n + C. \tag{5.19}$$

Theorems 9.1.1 and 9.1.2 in Lancaster and Rodman (1995) show that if the starting value X_0 is symmetric and large enough, then Eq. (5.19) has a unique positive definite solution for all n, and the sequence (X_n) converges quadratically to the largest positive definite solution of Eq. (5.18).

At each step of the algorithm we solve Eq. (5.19) using the $O(k^3)$ algorithm of Bartels and Stewart (1972), until a desired level of accuracy is reached, which in our application typically occurs after less than 10 steps. There are other algorithms besides Newton's method which are more efficient when a high degree of accuracy is desired, but for the present case we are satisfied with this method as it is straightforward to understand and to implement, and it converges in a few steps to a solution accurate enough for our purpose.

To verify that the solution W^ϵ is such that $W^\epsilon \mathbf{1} = 0$, first notice that we can pull out a factor $(X^b)'$ from $\tilde{L}^\gamma$ and thus $\mathbf{1}'\tilde{L}^\gamma = \mathbf{0}'$ and $\mathbf{1}'W^\mu = \mathbf{1}'$. W^α has only one 1 per column and thus $\mathbf{1}'W^\alpha = \mathbf{1}'$. Therefore $\mathbf{1}'A = \mathbf{0}'$ and $A\mathbf{1} = \mathbf{0}$. Because we can pull out a factor $(X^b)'$ on the left and a factor X^b on the right of $\tilde{P}^a$ we can also see that $\mathbf{1}'\tilde{P}^{a,\gamma}\mathbf{1} = 0$. Multiplying Eq. (5.18) by $\mathbf{1}'$ from the left and by $\mathbf{1}$ from the right, it follows that $X'\mathbf{1} = \mathbf{0}$ and by symmetry, also $X\mathbf{1} = 0$.

Appendix 5.B Efficient computation of weight matrices

The derivation of the algorithm in ensemble space starts by applying Woodbury's formula to compute the inverse in the Kalman gain $\tilde{K}(\gamma P^b)$

and results in the following expression after some further simplifications:

$$\tilde{K}(\gamma P^b) = \gamma\Big((k-1)I + \gamma S\Big)^{-1}(HX^b)'R^{-1}.$$

Using the definition of Q we can then write

$$\tilde{Q} = \gamma S\Big((k-1)I + \gamma S\Big)^{-2},$$

which is correct because the matrices on the right commute. To compute $\tilde{K}((1-\gamma)Q)$ we substitute the expression for $\tilde{Q}$ in the definition and apply again Woodbury's formula. After some further simplifications we can find that:

$$\tilde{K}((1-\gamma)Q) = (1-\gamma)\Big(I + (1-\gamma)\tilde{Q}S\Big)^{-1}\tilde{Q}(HX^b)'R^{-1}.$$

Splitting the ensemble into mean and deviations one can rewrite the W^μ matrix in Section 5.3 as

$$W^\mu = I - \tilde{L}^\gamma HX^b + \tilde{L}^\gamma(y - H\bar{x}^b)\mathbf{1}',$$

where the first part will be computed using the S matrix and the last part using the S matrix and the c vector. Using the expressions for $\tilde{K}(\gamma P^b)$ and $\tilde{K}((1-\gamma)Q)$ to compute $\tilde{L}^\gamma$ and some further simplifications, we can derive the first part as

$$I - \tilde{L}^\gamma HX^b = \Big(I + (1-\gamma)\tilde{Q}S\Big)^{-1}(k-1)\Big((k-1)I + \gamma S\Big)^{-1}.$$

Finally, using the spectral decomposition of S and basic rules of algebra we can find the rational function

$$f^\mu(\lambda) = \frac{(k-1)\gamma\lambda + (k-1)^2}{\gamma\lambda^2 + 2(k-1)\gamma\lambda + (k-1)^2}. \tag{5.20}$$

The second part of the matrix W^μ can be derived similarly as:

$$\tilde{L}^\gamma(y - H\bar{x}^b) = \Big(I + (1-\gamma)\tilde{Q}S\Big)^{-1}\Big((1-\gamma)\tilde{Q} + \gamma\Big((k-1)I + \gamma S\Big)^{-1}\Big)c,$$

from which we can find the function $f^{\bar{\mu}}$ after plugging in the spectral decomposition of S:

$$f^{\bar{\mu}}(\lambda) = \frac{1}{(k-1)+\gamma\lambda} \cdot \left(\gamma + \frac{(k-1)\gamma(1-\gamma)\lambda}{\gamma\lambda^2 + 2(k-1)\gamma\lambda + (k-1)^2} \right). \tag{5.21}$$

Using the expression for $\tilde{K}((1-\gamma)Q)$ and $\tilde{Q}$ we can similarly find that

$$\tilde{P}^{a,\gamma} = \Big(I + (1-\gamma)\tilde{Q}S \Big)^{-1} \gamma S \Big((k-1)I + \gamma S \Big)^{-2},$$

from which f^{γ} can easily be found as

$$f^{\gamma}(\lambda) = \frac{\gamma\lambda}{\gamma\lambda^2 + 2(k-1)\gamma\lambda + (k-1)^2}. \tag{5.22}$$

For the weights $\alpha^{\gamma,i}$ the derivation is similar and we find that they are proportional to

$$\exp\Bigg(-\frac{1}{2} \Big((k-1)^2(1-\gamma) \Big((k-1)I + \gamma S \Big)^{-2} \cdot \Big(I + (1-\gamma)\tilde{Q}S \Big)^{-1} (S - c\mathbf{1}') \Big)_{ii} \Bigg).$$

The final expression can be found by developing the last product in the exponential and by substituting the spectral decomposition of S, which results in the following:

$$f^{\alpha}(\lambda) = \frac{(k-1)^2(1-\gamma)}{\gamma\lambda^2 + 2(k-1)\gamma\lambda + (k-1)^2}. \tag{5.23}$$

One can easily see what happens in the limiting cases of $\gamma = 0$ and $\gamma = 1$. Setting $\gamma = 0$ gives $f^{\mu}(\lambda) = f^{\alpha}(\lambda) = 1$ and $f^{\bar{\mu}}(\lambda) = f^{\gamma}(\lambda) = 0$. Hence $W^{\mu} = UU' = I$, $W^{\epsilon} = 0$ and $\alpha_i \propto \exp(-\frac{1}{2}U_{ii} + c_i) \propto \ell(x_i|y)$. Hence the resulting analysis is equivalent to the PF.

In the case where $\gamma = 1$, $f^{\alpha}(\lambda) = 0$ and thus $\alpha^{\gamma,i} = \frac{1}{k}$ and $W^{\alpha} = I$. Furthermore, $f^{\mu}(\lambda)$ simplifies to $(k-1)/((k-1)+\lambda)$, and $f^{\gamma}(\lambda)$ to $\lambda/((k-1)+\lambda)^2$. The CARE in Eq. (5.10) has thus the positive semidefinite solution $W^{\epsilon} = U\delta(f^{\epsilon}(\boldsymbol{\lambda}))U'$ where

$$2f^{\epsilon}(\lambda)f^{\mu}(\lambda) + f^{\epsilon}(\lambda)^2 = (k-1)f^{\gamma}(\lambda)$$

or

$$f^{\epsilon}(\lambda) = \frac{\sqrt{(k-1)((k-1)+\lambda)} - (k-1)}{((k-1)+\lambda)} \tag{5.24}$$

The sum $W^{\mu} + W^{\epsilon}$ is thus given by

$$U\delta\Big(\sqrt{\frac{(k-1)}{(k-1)+\boldsymbol{\lambda}}}\Big)U',$$

which is the formula for the transformation matrix in the ETKF.

Chapter 6

Application of the LETKPF to COSMO

The content of this chapter has been submitted as the second part of Robert et al. (2017). Numerical experiments with a setup similar to the one used operationally at MeteoSwiss were conducted with the LETKPF introduced in Chapter 5. We first describe the experimental setup in Section 6.1 and then discuss the main results in Section 6.2. In Section 6.3 we conclude with a summary of Chapters 5 and 6 and give some perspectives. In Appendix 6.A we present additional unpublished experimental results.

6.1 Experimental setup

In this section we briefly introduce the KENDA system used at MeteoSwiss before describing the test period and the experiments.

6.1.1 The KENDA system

The numerical experiments in this study were carried out using the KENDA (Kilometer-Scale Ensemble Data Assimilation) system as described in Schraff et al. (2016). It is based on the COSMO model (Baldauf et al., 2011) with a setup similar to the operational implementation at MeteoSwiss.

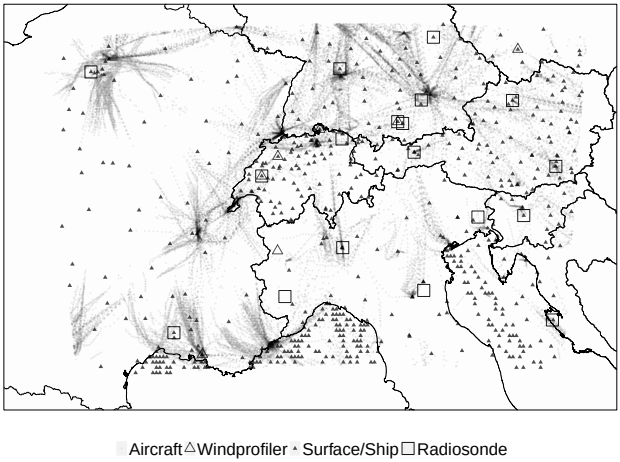

Figure 6.1: COSMO model domain and geographical distribution of the observations actively assimilated at least once during the 12-day test period.

The COSMO model is a convective-scale, non-hydrostatic NWP model developed within the COSMO consortium (http://cosmo-model.org) and operated at many national weather services worldwide. The atmospheric prognostic variables are the three-dimensional wind, temperature, pressure, turbulent kinetic energy and specific contents of water vapor, cloud water, cloud ice, rain, snow and graupel. The equations for the dynamic variables are solved using a Runge-Kutta time-splitting scheme. Deep convection is explicitly computed, whereas shallow convection is parametrized. A one-moment Lin-type cloud microphysics scheme is responsible for the conversions among all cloud and hydrometeor types. The turbulence parameterization is based on the prognostic Turbulent Kinetic Energy (TKE) equation and radiative effects are parametrized using a δ-two-stream scheme. A multi-layer soil model provides the lower boundary condition at the ground. For more details of the COSMO model we refer to Baldauf et al. (2011).

The MeteoSwiss COSMO implementation covers a geographical domain of central Europe (see Fig. 6.1) with a horizontal mesh-size of 2.2km and 60 terrain-following vertical levels up to a model top at roughly 22km.

The reference analysis algorithm is the LETKF based on Hunt et al. (2007) with a configuration similar to that described in Schraff et al. (2016). This

algorithm is operationally used at MeteoSwiss and serves as a reference for comparisons of the new LETKPF methods. For all algorithms, localization is done in observation space using a constant vertical and horizontal localization radius resulting in a varying effective number of observations being assimilated throughout the analyses. A multiplicative, adaptive covariance inflation scheme is used to account for unrepresented model error. In the operational MeteoSwiss implementation, additional additive covariance inflation in form of the relaxation to prior perturbation (RTPP) (Zhang et al., 2004) method is applied. As RTPP cannot be transferred immediately to the LETKPF we did not use it in this study for comparison reasons.

The KENDA system produces hourly ensemble analyses with 40 members. Lateral boundary conditions are taken from the first 40 global ECMWF EPS forecast members interpolated to the COSMO model grid. Ensemble perturbations are then calculated by subtracting the ensemble mean from each member. These perturbations are then added to the latest interpolated ECMWF HRES forecast valid at the same time to build a new ensemble. In order to get a reasonable spread-error relationship at the lateral boundaries, members from an older global ensemble forecast with lead times from +30h to +42h and thus a larger spread are used. The initial ensemble at the start of the test period are obtained from the pre-operational MeteoSwiss KENDA cycle.

The observations used for the experiments are similar to that used operationally at MeteoSwiss: radiosonde (TEMP) temperature, wind and humidity data, wind profiler wind data, surface (SYNOP) and ship surface pressure data and aircraft temperature and wind data. The geographical locations of all observations that were actively assimilated at least once during the 12-day test period are shown in Fig. 6.1.

The observation error covariance R is assumed to be diagonal with values estimated from innovation statistics following Desroziers et al. (2005) and Li et al. (2009) and are listed in Table 6.1.

6.1.2 Test period

The 12-day test period for the experiments from 4 to 16 June 2015 was chosen to include both convective and stratiform precipitation events over the domain of interest. From 4 to 9 June the weather in central Europe was dominated by high pressure systems over northern Europe leading to

Level [hPa]	Wind [m/s]	Temperature [K]	Rel. Humidity [%]
300	2.1 / 1.9 / 1.6	0.6 / 0.6	13.8
400	1.8 / 1.6 / 1.4	0.5 / 0.5	13.1
500	1.6 / 1.4 / 1.2	0.6 / 0.6	12.9
700	1.6 / 1.4 / 1.2	0.7 / 0.7	12.2
850	1.7 / 1.5 / 1.3	1.0 / 0.8	12.8
1000	1.7 / 1.5 / -	1.1 / 1.1	9.3

Table 6.1: Observation errors $\sqrt{\sigma_0^2}$ for wind, temperature and relative humidity at different heights in the atmosphere. The first value is for radiosonde, the second value for aircraft and the third value for wind profiler observations.

high surface temperatures and a diurnal cycle of convection over the Alpine Ridge. From 9 to 16 June, a cut-off low west of France and its associated fronts caused several bands of both stratiform and embedded convective precipitation sweeping over the Alps.

6.1.3 Assimilation methods

In all our experiments we compare four assimilation algorithms. The LETKF is close to the operational setup and serves as a reference. For the LETKPF we test two variants of the algorithm with the different adaptive γ schemes described in Section 5.3, which we refer to as LETKPF-ess50 for the scheme targeting a ESS of 50%, and as LETKPF-minMSE for the scheme minimizing the MSE of the analysis mean. The fourth algorithm is the local PF (LPF), defined as our LETKPF with γ set to zero.

6.2 Results

First we show how the LETKPF works and how it differs from the LETKF in a particular one-step analysis case study. Then we present results on the verification of radiosonde observations during the cycling assimilation phase. Finally we look at the 24-hour forecasts and contrast the performance of the different algorithms.

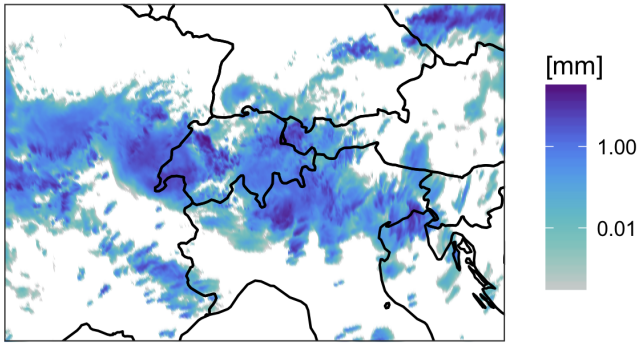

Figure 6.2: Maps of $\bar{x}^b$ for total precipitation in [mm] during the last hour before the analysis in the case study.

6.2.1 One-step analysis

We now look in more detail at a one-step analysis on the 14th of June at 1700 UTC. The meteorological situation at analysis time is summarized in Fig. 6.2 with the total precipitation of the background mean $\bar{x}^b$ in [mm]. A large storm is going through the domain with strong convection happening in many different areas.

To illustrate how the LETKPF differs from the LETKF we look at maps of the analysis weight matrix W (to be precise, we look at the values of $W = \tilde{W}^{\mu} W^{\alpha} + W^{\epsilon}$, where $\tilde{W}^{\mu}$ is the left side of Eq. (5.13) only, to remove the effect on the mean and focus on the particle deviations). For simplicity, we choose to focus on the contribution of the first two particles to form the analysis particle $x^{a,1}$. These contributions are summarized in the first two elements of the first column of the W matrix, W_{11} and W_{21}. Because the analysis is done locally, these values change at every grid point. Averaging over the lower atmosphere (pressure larger than 700 hPa) we can show the results for different algorithms as maps in Fig. 6.3. The particle $x^{a,1}$ is mainly composed of itself – $x^{b,1}$ – when the value mapped is close to 1, while it is recomposed from other particles when it is close to 0. When this is the case, other particles are resampled instead and glued together to form the analysis.

In the first row of Fig. 6.3 we can see what happens in the case of the

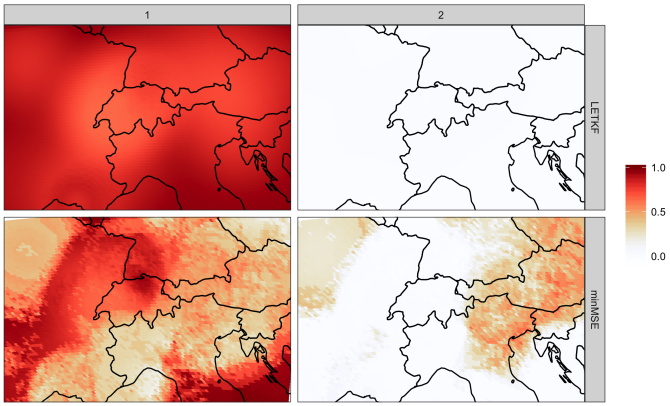

Figure 6.3: Maps of W_{i1}, the contribution of particles $i = 1, 2$ to the analysis particle 1 in the lower atmosphere during the case study, when using LETKF in the first row and LETKPF-minMSE in the second.

LETKF: $x^{a,1}$ is mainly composed of $x^{b,1}$, with the other particles only marginally influencing the analysis through their covariance with $x^{b,1}$. In the second row, however, the same maps for the LETKPF-minMSE shows a more interesting behavior: particle $x^{a,1}$ is composed of itself in some areas, for example in North-East France and Switzerland, but in some places $x^{a,1}$ is composed in a large part of $x^{b,2}$ as in Austria and the North-East of Italy, or of other particles not shown here as in the North-West of Italy. These maps illustrate well how the LETKPF produces an analysis by combining different particles locally, resampling particles where they fit the data well and discarding them where other candidates fit better.

Not only the weights W, but also the value of γ vary locally. In Fig. 6.4, the γ chosen in the lower atmosphere with different adaptive criteria is displayed together with the ESS. The value of γ shows where the algorithm prefers to stay closer to the LETKF (where γ is large) and where it chooses an update much closer to the PF (where γ is small). The functional relationship between γ and ESS is non-linear and depends locally on the background ensemble distribution and the observations. If the ESS is close to 1, little resampling occurs and most particles are reused, while if it is close to 0, a few particles are resampled many times.

The maps in Fig. 6.4 are quite different for the two algorithms: the γ chosen by the ESS criterion varies less in space, while the γ chosen by LETKPF-

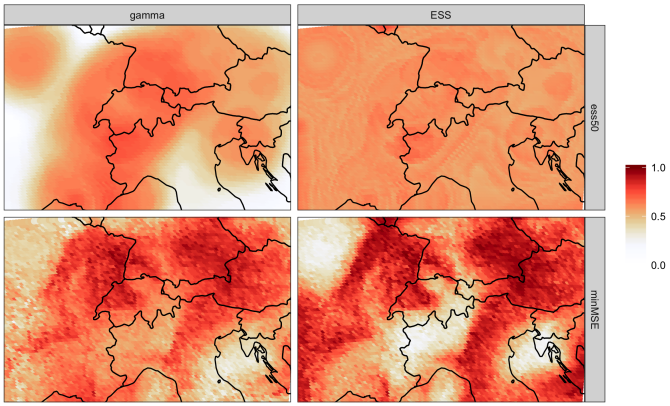

Figure 6.4: Adaptive choice of γ (left panel) and corresponding ESS (right panel) in the lower atmosphere during the case study, with LETKPF-ess50 in the first row and LETKPF-minMSE in the second.

minMSE has a rougher pattern. Both methods agree in some regions of the domain, but in others they make opposite choices, as for example in the region around Paris. Unfortunately, there is no ground truth to compare the chosen γ with, and one has to rely on the overall performance of a particular algorithm to see if it fared well. We attempted to find correlations between the choice of γ and the meteorological situation, for example by looking at measures of non-Gaussianity, but arrived at no clear result. Furthermore, with the current operational setup the number of observations varies quite a lot in the domain (from 0 to 100), which seems to have a strong influence on the choice of γ (see for example in the the region of high-density observations around Paris). Further research will be necessary to understand the interplay between the different parameters and the optimal choice of γ.

Looking closely at the patterns of ESS in Fig. 6.4 and comparing them to those of W_{i1} in Fig. 6.3 we can notice interesting patterns. One puzzling fact is that there are regions where ESS is high, but W_{11} is low (for example in Austria), which seems contradictory as there should not be much resampling occurring. In Fig. 6.6 we show the diagonal of W, the contribution of each particle to its own analysis, for the 40 ensemble members. We can now see that W_{11} being low in Austria was rather an exception than the rule, and a large proportion of particles are resampled, thus ex-

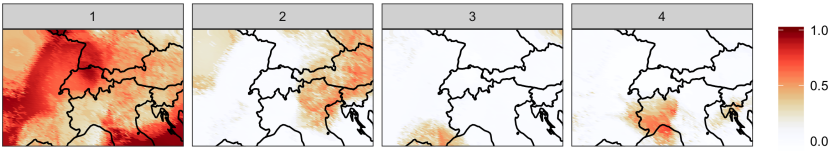

Figure 6.5: Maps of W_{i1}, the contribution of particles $i = 1, \ldots, 4$ to the analysis particle 1 in the lower atmosphere during the case study.

plaining the corresponding high value of ESS which can be seen in Fig. 6.4. Conversely, there are regions with a particularly low ESS, as for example in the Piedmont (Italy). Observing the maps in Fig. 6.6 we can see that indeed most particles are not resampled in this area, except a few. We cannot show the whole 40 figures of 40 maps which would be necessary to verify it, but it makes sense that these few particles are probably resampled instead of the one that are not selected in this area. In Fig. 6.5 we show the same as in Fig. 6.3 but for more particles. There we can see that indeed particle 4, which was resampled in Piedmont (high value of W_{44} in Fig. 6.6) is used to form the analysis of particle 1 in this area.

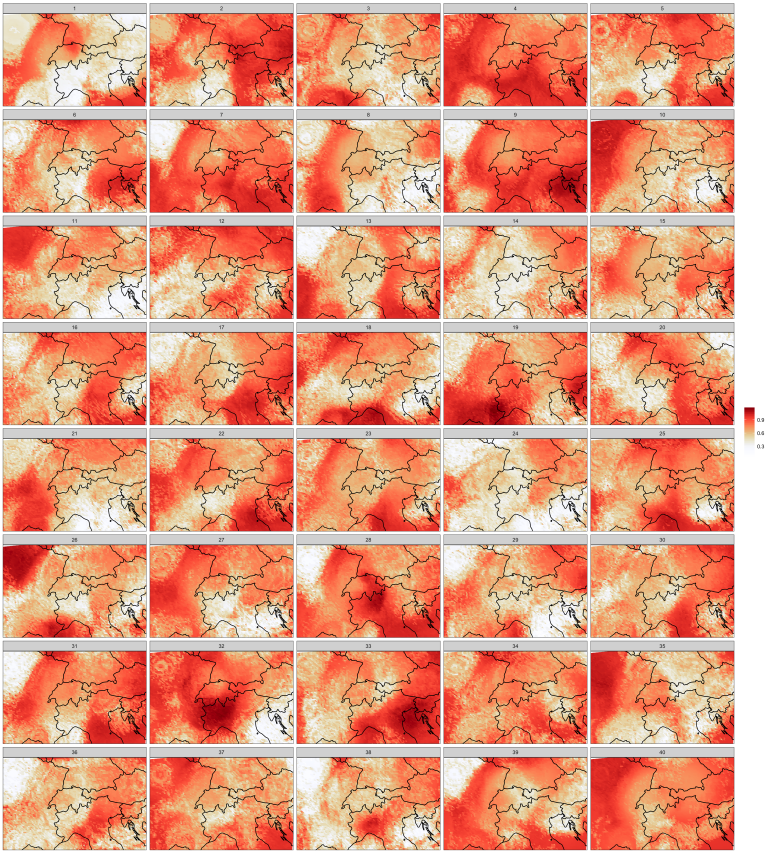

Figure 6.6: Maps of W_{ii}, the contribution of particles i to its own analysis in the lower atmosphere during the case study.

6.2.2 Cycled experiment

In order to assess the quality of the analysis during the assimilation, we verify the one-hour-ahead forecast produced by different algorithms against all radiosonde observations. As error metrics we use the bias of the forecast mean and the CRPS, a strictly proper scoring rule which takes into account both the sharpness and the calibration of the ensemble (Gneiting and Raftery, 2007). More scores will be considered for the forecast experiment described below, but for the analysis they are sufficient to evaluate the overall performances of the algorithms.

The error metrics are aggregated over the whole period and over different pressure levels. In Fig. 6.7, the difference of the bias and CRPS of the new algorithms with the bias and CRPS of the LETKF are displayed as vertical profiles. For the bias, the difference of the absolute value is displayed, such that for both the bias and the CRPS a negative value indicates an improvement over the LETKF. In each panel there is a smaller plot included to show the profile of the LETKF error.

For the relative humidity (RH), the pressure level and the type of method have a strong influence on the CRPS. It seems that the LETKPFs are worse than the LETKF for the lower atmosphere, but they are sometimes better for the middle and upper atmospheres. There is no clear ranking between the variants of LETKPFs, with LETKPF-minMSE performing best around 700 [hPa] while LETKPF-ess50 seems better around 400 [hPa]. In terms of bias, we can also see some large gains for the LETKPFs in the middle and upper atmospheres.

The LETKF predicts temperature (T) better for almost all pressure levels both in terms of CRPS and bias. This comes as no surprise as temperature is the most Gaussian of all the variables. LPF is clearly worse than the other algorithms in terms of CRPS, while it is fares relatively well in terms of bias, particularly at 1000 [hPa].

The LETKPFs improve over the LETKF for predicting the wind speed (WIND) at middle to lower atmosphere, as can be seen from the CRPS and bias profiles. The LETKPF-minMSE seems to have the most consistent advantage, if not always the largest. The LPF, on the other hand, has trouble with WIND observations and is the worst method in terms of CRPS while its performance in terms of bias is erratic.

In Fig. 6.8 we compare the root mean squared error (RMSE) to the spread of the background ensemble, which should be equal if the ensemble is well

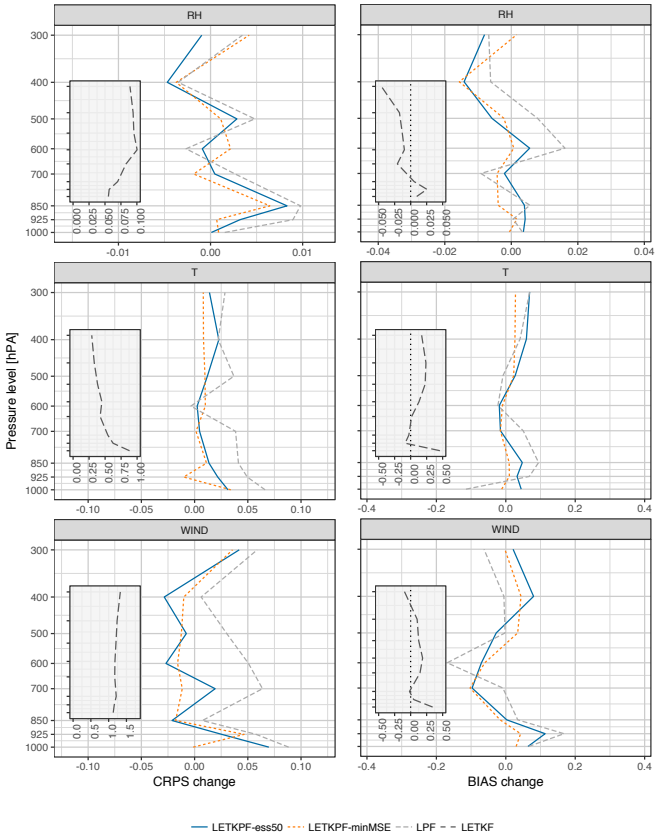

Figure 6.7: Change in CRPS and bias relative to LETKF analyses during the cycled experiment. More precisely CRPS(LETKPF) - CRPS(LETKF) and |bias(LETKPF)| - |bias(LETKF)|. A negative change indicates a reduction of CRPS or of bias, respectively. Note the different scales on the x-axis. In the small plot is the CRPS and bias of the reference LETKF.

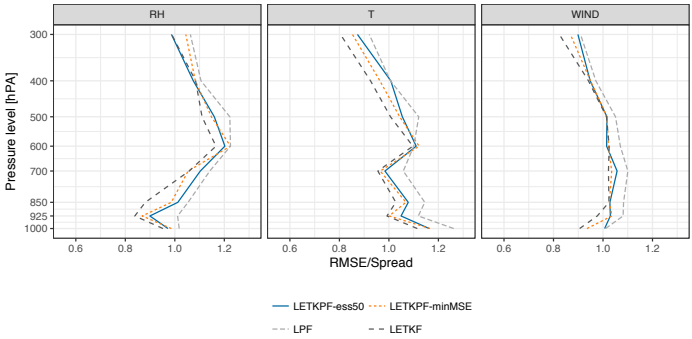

Figure 6.8: Ratio of RMSE over the spread of the background predictive distribution during the cycling experiment. A ratio larger than one indicates that the ensemble is too concentrated.

calibrated (see for example Fortin et al. (2014)). To take into account the observation error, we actually compare the observed RMSE to the spread of the predictive distribution $\pi^b(y)$. In the case of a diagonal R, we can compute this spread squared separately for each observation by adding the variance of the forecast ensemble to the corresponding diagonal element of R. We then aggregate by averaging over all observations, and take the square root before comparing to the RMSE. The profiles in Fig. 6.8 show that overall the ensembles are well calibrated. In terms of relative humidity it seems that the ensembles lack spread in the middle atmosphere, while they are too dispersed in the upper atmosphere in terms of temperature and wind. In general the LETKPFs have a larger ratio than the LETKF, due mainly to a reduction in spread because of resampling. Better calibration could be achieved in the future by fine tuning of the R matrix and by using refined covariance inflation schemes.

6.2.3 Forecast experiment

Twice a day, at 0000 and 1200 UTC, a 24-hour forecast was launched from the current analysis ensemble. In Fig. 6.9 we look at the CRPS and bias of predicting radiosonde observations averaged over the whole domain and the whole forecast horizon (i.e. the scores of all forecast lead times were aggregated to one single score), similar to the cycled experiments in Fig. 6.7. The absolute CRPS is usually larger in the forecast than in the cycled experiment, with the strongest growth in the upper atmosphere for

the temperature and wind variables. However, the differences between the methods are much less pronounced than during the analysis and disappear almost completely at the end of the 24-hour forecast. The LPF is clearly worse than the other algorithms, particularly in terms of relative humidity and wind. For the relative humidity and temperature the LETKF is generally slightly better, while among the LETKPFs the LETKPF-minMSE is the best performer and even beats the LETKF for the wind variables at most levels.

More relevant for the forecast users, we now look at the hourly precipitation recorded at 121 stations over the Swiss domain (SYNOP data). In Fig. 6.10 we can see the evolution of the ensemble forecast means (the first 12 lead time hours of all forecasts are chunked together to build a continuous time series) over the whole period as compared to the actual observations (dots). It is interesting to notice how the different algorithms coincide most of the time but differ substantially for some events. For example, around the 8 June a large precipitation event is best predicted by LETKPF-minMSE forecasts, while the LPF forecasts overestimate it, and the other method underestimate it. At other times, all methods seem to miss or produce spurious events.

The evolution of the skills of the ensemble to predict hourly precipitation larger than 0.1 [mm] as a function of lead time is illustrated in Fig. 6.11, where we see the equitable threat score (ETS, see Eq. 7.18 in Wilks (2006)), the frequency bias index (FBI, see Jolliffe and Stephenson (2003)) and the Brier skill score (BSS, see Eq. 7.35 in Wilks (2006)) of the forecast ensembles. In terms of ETS, the LETKPFs and the LETKF are more or less equivalent, with some lead time where one or another is better. The LPF on the other hand is clearly worse during the first 12 hours of forecast but then stabilizes. The FBI plot shows that all methods tend to overforecast the event, while the LETKPF-ess50 has the best overall performance. For the BSS, the Brier score normalized by the climatology forecast score (as computed from the test period), the LETKPF-ess50 is again the best performer, while the LPF has no skill in the first half of the forecast but reaches similar level to the others in the second half.

The calibration of the methods is shown in the reliability diagrams of Fig. 6.12. One can see that all algorithms have some skill except maybe the LPF during the first 12 hours of forecast. The LETKPF-ess50 is once again the best performer and the LETKF is generally less well calibrated than the LETKPFs, but the differences are small and depend on the forecast probabilities. The rank histograms indicate an overall positive bias, but

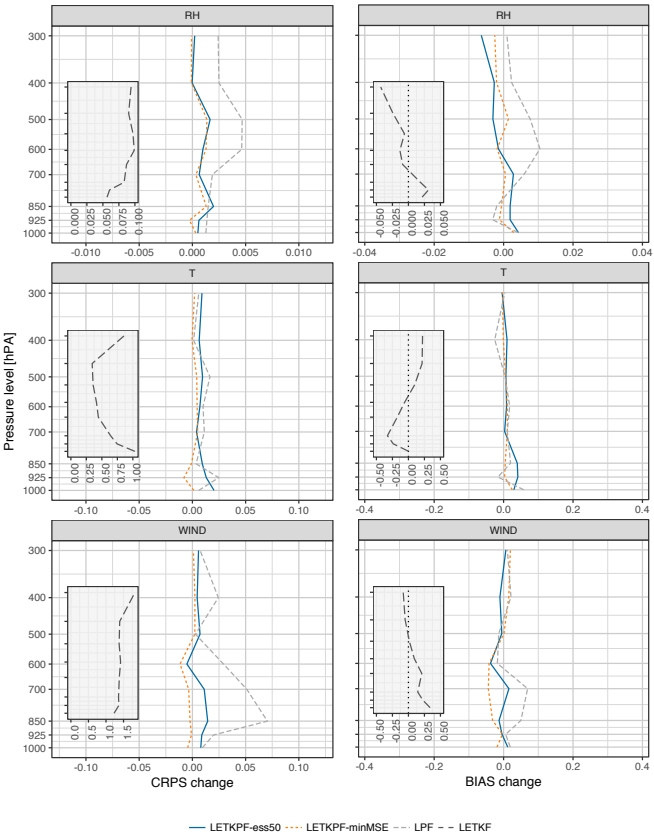

Figure 6.9: Change in forecast CRPS and bias relative to LETKF-driven forecasts, similar to Fig. 6.7. The scores of all forecast lead times were aggregated. A negative change indicates a reduction of CRPS or of bias, respectively. Note the different scales on the x-axis. In the small plot is the CRPS and bias of the reference LETKF.

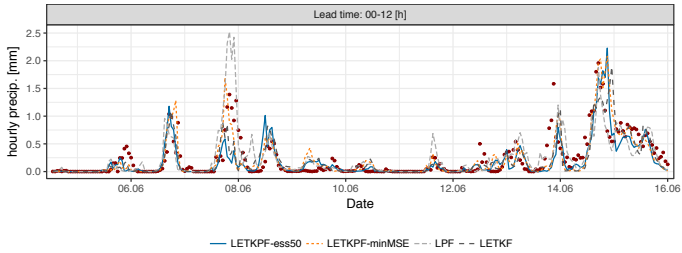

Figure 6.10: Time evolution of the ensemble mean (various lines) compared to the observations (dots) of hourly precipitations from 121 Swiss SYNOP stations.

no particular differences between the methods (not shown).

6.2.4 Discussion

The results of the cycled and forecast experiments show that the LETKPFs perform similarly to the LETKF. The new algorithms bring some improvements for some variables at some pressure levels – for example for wind in the middle and upper atmosphere – but they also perform worse in other cases. As expected, these improvements over the LETKF occur for the most non-Gaussian variables, while for Gaussian variables like temperature the LETKF is usually better. During the forecast in particular, the LETKPFs show some benefit in predicting hourly precipitation, which is a highly non-Gaussian variable. The better ability of the EnKPF to deal with rain fields confirms previous results with a toy model of cumulus convection (Robert and Künsch, 2017b).

The LPF is surprisingly not as bad as one could expect given its simplicity, which shows that localizing the PF is a viable strategy, but the ability to combine it with the LETKF seems to bring clear advantages. However, the question of which criterion to use for choosing the proportion of PF and of LETKF in the analysis is still not clear from the empirical results. The LETKPF-minMSE seems to be slightly better for the model variables (temperature, relative humidity and wind), but the LETKPF-ess50 typically performs better for forecasting hourly precipitation.

These results are promising and indicate that the LETKPF can be used in practice. However, further experiments should be conducted with longer periods and during different meteorological situations.

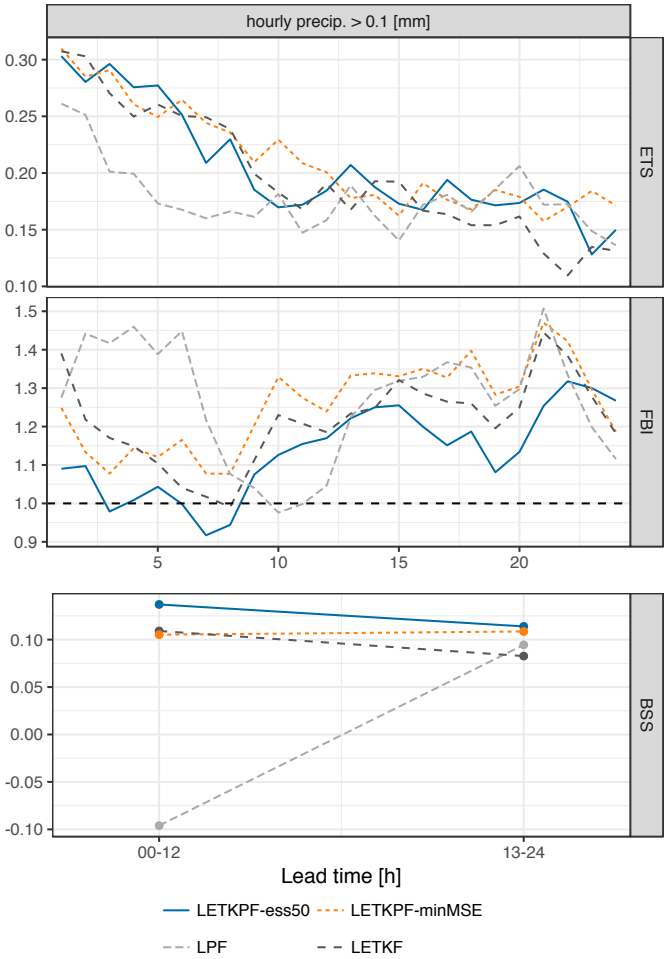

Figure 6.11: Evolution of various scores for predicting hourly precipitation larger than 0.1 [mm] as a function of lead time during forecast, and aggregated over all forecasts of the period under consideration. Reference observations are accumulated hourly precipitation from 121 Swiss SYNOP stations. For ETS and BSS the higher the better (maximum 1), and for FBI the closer to 1 the better. Because of system constraints, the BSS is aggregated every 12 hours, whereas the other scores are aggregated hourly.

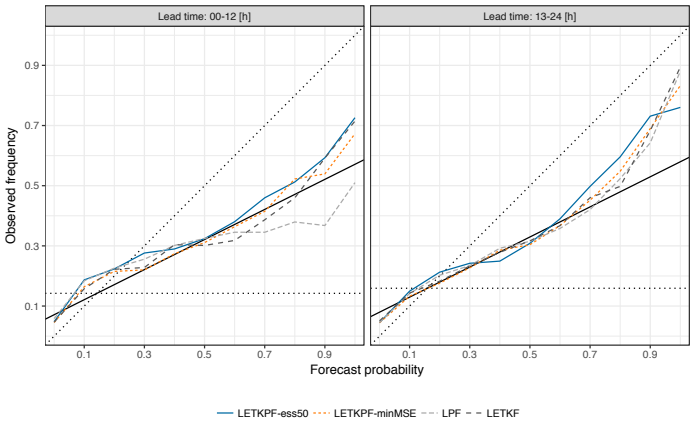

Figure 6.12: Reliability diagram for predicting more than 0.1 [mm] of hourly precipitation in the first 12 hours (left panel) and the second 12 hours (right panel). The solid black line indicates no skill while the diagonal is for perfect reliability.

6.3 Summary and conclusions

High-dimensional non-Gaussian filtering problems, such as encountered in convective scale data assimilation, call for the development of new algorithms. In the present paper we proposed the LETKPF, which builds on the EnKPF to make it more efficient and applicable in practice. In particular, we reformulated the whole algorithm in ensemble space and derived a deterministic scheme such that it now has the ETKF instead of the stochastic EnKF as a limiting case. The same approach as that of the LETKF was taken for localizing the algorithm, with a few additional steps to deal with the PF nature of the analysis. While this may not be the optimal localization strategy, it is widely used in practice and made the implementation in the existing framework feasible. Furthermore, a new criterion for choosing the proportion of analysis to be done with the PF and the ETKF was proposed based on the idea of minimizing the predictive MSE.

The new algorithm was implemented in the COSMO data assimilation framework and tested on a 12-day period of hourly assimilation in a region surrounding Switzerland. These experiments showed that the newly proposed algorithm is applicable in practice and can perform similarly to the LETKF, which is the algorithm used operationally at MeteoSwiss. In

particular, the LETKPF brings some remarkable improvements for non-Gaussian variables such as wind and hourly precipitation. These results are promising and we hope that they will stimulate further experiments and research with the LETKPF and other types of localized hybrid algorithms.

In the present study, we have relied on the setup used for the LETKF, but some questions concerning the particularity of the LETKPF – or more generally any hybrid algorithm – need to be further investigated. The optimal choice of γ is still poorly understood and the experimental results were not conclusive, showing that both proposed methods work better in some situations. The alternatives discussed in Section 5.3.3 might be promising and could be tested in practice if efficient implementations are found. In general, it would be of great interest to better understand the interplay between the optimal choice of γ and the non-Gaussianity of the distribution, the number of observations assimilated, the model error, etc. As we have seen in the experiments, the choice of γ should certainly vary for every grid point, as different situations call for different decisions. One could push this idea further and choose a different γ for different types of observations or even for different model variables. For example, one could imagine using a γ close to 1 for temperature while using a small γ for wind or relative humidity.

Another aspect that should be explored further is how to control the ensemble spread for the LETKPF. Among other means, to do so the LETKF relies on covariance inflation (multiplicative and additive) and RTPP. However, both of these methods derive their rationale from the idea that the analysis consists in moving a little bit each particle such that the new ensemble has a correct mean and covariance. RTPP controls the loss of spread by recombining each analysis particle with its corresponding background particle, while covariance inflation somehow increases the analysis ensemble covariance. Because the LETKPF analysis consists partly in resampling particles, one cannot just transfer these techniques blindly. One obvious solution to this issue would be to work with the mixture representation of the analysis and control the spread by adding more covariance to the mixture components. Similarly, for RTPP one could use the idea of combining the analysis particle with the background ensemble, but by taking into account the resampling step of the analysis.

Appendix 6.A Additional results

In this section we present a few additional results from the COSMO experiments. In Section 6.A.1 we show the effect of the analysis by comparing ensembles against radiosondes observations in two different situations. In Section 6.A.2 we investigate in more detail the error growth of model variables during the forecast. In Section 6.A.3 we compare the performance of the stochastic EnKPF with the ETKPF on the cycled experiment.

6.A.1 Ensemble versus radiosonde observations

As another illustration of the different assimilation methods we can look at the ensemble in observation space for a particular set of observations. In this case we select the radiosonde of Payerne on the 15 June at 0000 UTC. In Fig. 6.13 we display the background and analysis ensemble together with the observation as a red line. However, because the filtering is done in 4D, as explained in Hunt et al. (2007), the observations and the analysis do not occur at the same time. Thus part of the shift between the background and analysis ensembles comes from the natural evolution of the system during the interval and not from the analysis itself. Furthermore, one should keep in mind that other observations are assimilated at the same time, which have an influence on the analysis seen here but are not displayed.

The illustration of Fig. 6.13 contrasts well the algorithms. In particular the resampling nature of the LPF is clearly highlighted, with the analysis ensemble showing a strong sample depletion problem. Considering this figure it is somehow surprising that the LPF manages relatively well overall. The most striking difference between the LETKPFs and the LETKF is maybe that they have more non-Gaussian features in their background distribution. For example in the upper atmosphere, the LETKPF-ess50 has a rather skewed distribution, almost bimodal, while the LETKPF-minMSE background ensemble is also generally more skewed than the LETKF one.

Another interesting feature of these plot is that while the background particles always have relative humidity smaller or equal to 1, the analysis of the LETKF and to a lesser extent of the LETKPF-minMSE, are sometime larger than 1. While it is technically not unfeasible, as there are phenomena of supersaturation, in the given case it seems to be an artificial feature introduced during the analysis that is most likely undesirable.

To disentangle the influence of the background ensemble from that of the

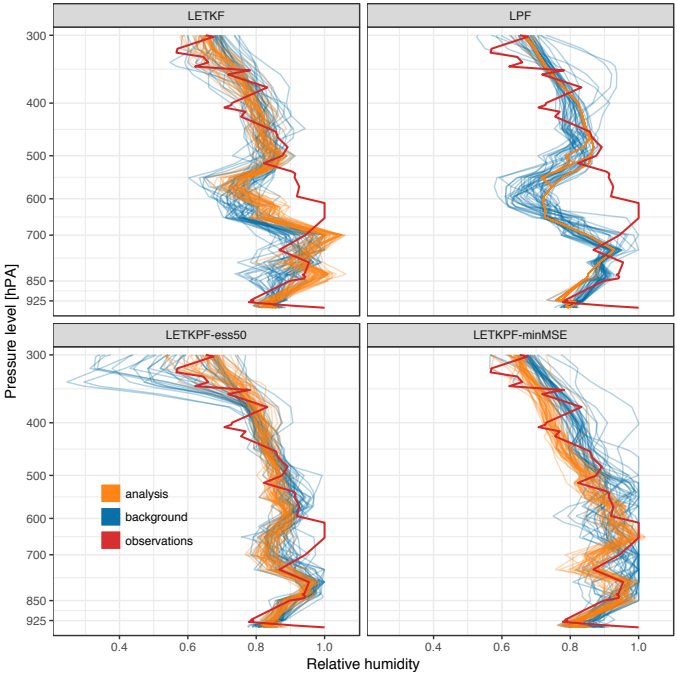

Figure 6.13: Effect of the analysis verified against the radiosonde measurements of Payerne on the 15 June at 0000 UTC. The blue lines are for the background ensemble members, while the orange lines show the analysis. The red line represents the observations.

assimilation algorithm we run another example where we always use the same background ensemble (produced by the LETKF cycling experiment). In Fig. 6.14 we can see the resulting plot of the background and analysis ensembles, this time for the radiosonde of Kümmersbruck in Bavaria on the 14 June at 1700 UTC (the same time as in the case study of Section 6.2.1). We did not choose the same example as above as we did not have the background ensemble available then. In this case the differences between the algorithms are much less visible, except for the LPF. Therefore, we can conclude that it is important to look at cycled experiments to really see the difference between the algorithms taking shape.

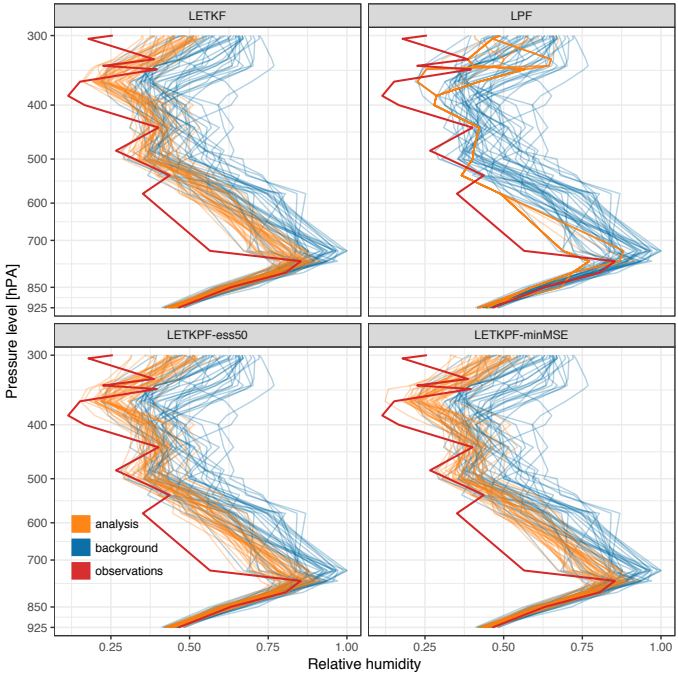

Figure 6.14: Effect of the analysis verified against the radiosonde measurements of Kümmersbruck on the 14 June at 1700 UTC. The blue lines are for the background ensemble members, while the orange lines show the analysis. The red line represents the observations.

6.A.2 Error growth with lead time

In Section 6.2, we only looked at the radiosonde observations aggregated over the whole period, space and forecast lead time. For the surface data of 121 swiss stations we investigated in more detail the evolution of the error as a function of lead time. We now do the same for other sources of data.

In Fig. 6.15 we can see the CRPS error growth during the forecast for all SYNOP stations (not only Switzerland but the whole domain). In Fig. 6.16 the same is plotted but for PILOT observations (for the different types of observations see WMO (2015)), aggregated at three different atmospheric levels (lower atmosphere means pressure larger than 700 [hPa], upper means less than 500 [hPa], and middle is in between). One could certainly draw more conclusions from these plots but we restrict ourselves to three points. First, all the algorithms perform comparatively well, except the LPF which is notably worse. Second, the difference between the algorithms disappear after 24 hours, but persist for quite some time, for example for the pressure variable. Third, the error growth at the beginning of the forecast is seemingly faster for the PF algorithms. In particular, looking at the error growth for surface pressure we can see that the LPF actually starts better than all other algorithms, but quickly deteriorates. This is symptomatic of imbalances introduced during the analysis that are dissipated in the forecast, but can actually be detrimental as it seems to be the case for the LPF. More work would need to be done to actually study the fields produced by the analysis and monitor their evolution during the forecast, but it would require more knowledge of the COSMO numerical model and falls out of the scope of this thesis. However, investigating the problems related to the balance of the analysis fields is extremely important and should definitely been pursued further in the future (Zeng and Janjić, 2016).

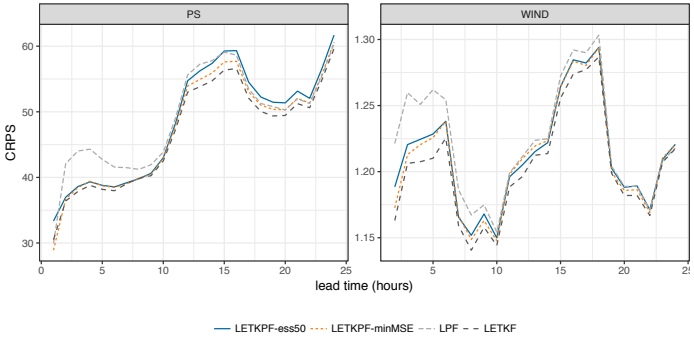

Figure 6.15: CRPS error growth of surface pressure and wind speed against SYNOP observations aggregated over the whole period and domain.

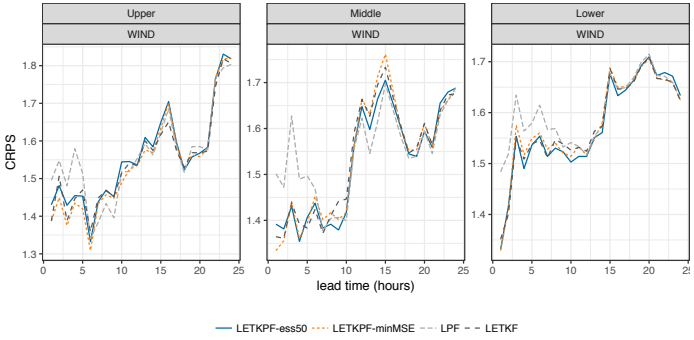

Figure 6.16: CRPS error growth of wind speed at different atmospheric levels against PILOT observations aggregated over the whole period and domain.

6.A.3 Stochastic versus transform filter

One aspect that we have not looked at in detail is the impact of the transform scheme of the LETKPF on the analysis quality. No systematic experiments have been ran because the goal was not to enter the open debate between stochastic and square-root filters but only to show that a transform EnKPF was possible and could work. As a reference we nonetheless ran the LEnKPF with the ESS set to 50% as a criterion and compare it to the LETKPF-ess50.

In Fig. 6.17 we show the same plot as in Fig. 6.7 but with the LEnKPF-ess50 replacing the LPF. Probably one should not look into it too much, but there seems to be some differences. In terms of relative humidity in the upper atmosphere the stochastic scheme is worse, while it is better in the middle and lower atmosphere. For temperature and wind it seems to be slightly better in the lower and upper atmosphere, but a bit worse in the middle. These differences are not particularly strong and should not be interpreted too much. However, from these plots one could conclude that the transform version of the filter is not worth the extra computational cost and that a stochastic EnKPF would be fine. A more systematic study should be conducted in the future to confirm or disprove this conclusion.

An additional remark concerning the stochastic EnKPF in ensemble space is that we do not use the same observation perturbations at every location, but the same ensemble perturbations. To be more precise, we generate W^ϵ in Eq. (5.7) using the same E matrix at all sites. Therefore if the observations assimilated are not the same at two neighboring sites, the perturbations corresponding to them will not be the same. To use the same perturbations everywhere one would have to generate two independent vectors $\epsilon^1, \epsilon^2 \sim \mathcal{N}(0, R)$ globally. Then, for each local analysis we would select the subvectors of ϵ^1 and ϵ^2 corresponding the observations assimilated locally, and compute a local W^ϵ from these subvectors using the same procedure as in Frei and Künsch (2013). It is a more expensive computation if $2d$ is larger than k, which is typically the case, but there might be gains in terms of smoothness of the global analysis particles. We have not implemented this alternative, but it could easily be done in the future.

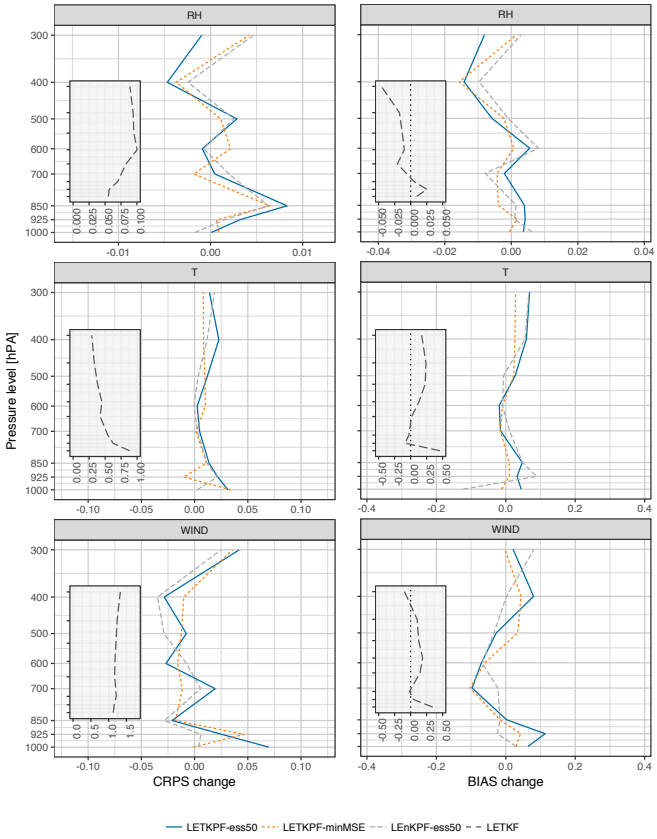

Figure 6.17: Change in CRPS and bias relative to LETKF analyses during the cycled experiment, similar to Fig. 6.7 except that it shows Local Ensemble Kalman Particle Filter (LEnKPF)-ess50 instead of LPF. A negative change indicates a reduction of CRPS or of bias, respectively. Note the different scales on the x-axis. In the small plot is the CRPS and bias of the reference LETKF.

Chapter 7

Other topics related to the LETKPF

In this chapter we present four topics related to the LETKPF which were not published elsewhere. In Section 7.1 we present a geometric intuition for the derivation of the EnKF and ETKF in ensemble space. In Section 7.2 we discuss a visualization technique which highlights interesting differences between various algorithms in ensemble space. In Section 7.3 we explore in depth the possibility to choose γ adaptively based on objective scores, as mentioned in Section 5.3.3. Finally, in Section 7.4 we briefly present how the algorithms were implemented in the COSMO assimilation framework and link to the available code.

7.1 Geometric derivations of the EnKF

Deriving an algorithm in ensemble space means finding the matrix W such that the analysis can be written as in Eq. (1.26), which we repeat here:

$$\boldsymbol{x}^a = \bar{x}^b \mathbf{1}' + X^b W, \tag{7.1}$$

In Chapter 5 we derived the EnKPF – and the EnKF as a special case – in ensemble space by using basic algebra. In this section we discuss in more detail the geometric intuition behind the EnKF and ETKF along the lines of Hunt et al. (2007). The discussion below is not meant as a rigorous

derivation but rather as an alternative perspective on the ensemble space formulation of the EnKF and ETKF.

We start by projecting y onto the ensemble space. Because the observations can have different variance and can even be correlated, we use the orthogonal projection with respect to the scalar product scaled by R^{-1}. To compare the ensemble to the observations we need the operator H. For ease of notation we define $Y^b = HX^b$ and $\bar{y}^b = H\bar{x}^b$, but using X^b or Y^b does not change anything to the argument. When we refer to the ensemble space we refer to the space spanned by $\boldsymbol{x}^b$ or by $H\boldsymbol{x}^b$ depending on the context. Let us write

$$y = \bar{y}^b + \hat{y} + r, \tag{7.2}$$

where $\hat{y}$ is the projection of y onto the ensemble space and r the residual such that $r'R^{-1}\hat{y} = 0$. The projection $\hat{y}$ can be found as

$$\hat{y} = Y^b S^{-1} c =: Y^b \beta,$$

where $S = (Y^b)'R^{-1}Y^b$ and $c = (Y^b)'R^{-1}(y - \bar{y}^b)$. The k-dimensional vector β contains the coordinates of $\hat{y}$ in the ensemble space, using Y^b as basis. It is possible to use a $(k-1)$-dimensional orthogonal basis instead, but it is more convenient for the derivation to work directly with Y^b, even if the columns of Y^b are colinear. Because we work in a $(k-1)$-dimensional space with a k-dimensional basis, S^{-1} should be understood as a Moore-Penrose pseudoinverse.

Now let us consider the problem from the point of view of the projection in ensemble space using Y^b as a (non-orthogonal) basis. In the end we will show that the resulting algorithm is equivalent to the full-space EnKF. The data to assimilate in this problem is β, while the background ensemble is nothing other than the $k \times k$ identity matrix I. The background covariance in ensemble space is thus simply $\frac{1}{k-1}I$ and the observation operator H is the identity. To find the equivalent of R in ensemble space, we derive the covariance of the projection onto this subspace of a random vector with covariance R. To do so consider $\epsilon \sim \mathcal{N}(0, R)$ and project it onto the ensemble space as for y in Eq. (7.2). Then the covariance of $\hat{\epsilon} = Y^b S^{-1}(Y^b)'R^{-1}\epsilon$ is $Y^b S^{-1}(Y^b)'$. Therefore, the observation error covariance matrix associated with β^ϵ, the coordinates of $\hat{\epsilon}$ in ensemble space, is S^{-1}.

Using Woodbury's formula it is possible to rewrite the Kalman gain $K(P) =$

$PH'(HPH'+R)^{-1}$ in the more convenient form

$$K(P) = \left(P^{-1} + H'R^{-1}H\right)^{-1} H'R^{-1}.$$

Substituting the ensemble space elements $P = \frac{1}{k-1}I$, $H = I$ and $R = S^{-1}$ we find our Kalman gain to be:

$$\tilde{K} = \left((k-1)I + S\right)^{-1} S. \tag{7.3}$$

The analysis ensemble of the EnKF is produced as

$$\boldsymbol{x}^a = \boldsymbol{x}^b + K(P)(y\mathbf{1}' + R^{1/2}E - H\boldsymbol{x}^b),$$

where E is a matrix with i.i.d. columns $\epsilon \sim \mathcal{N}(0, I)$. In our ensemble space assimilation problem, this simplifies to

$$\boldsymbol{x}^a = I + \tilde{K}(\beta\mathbf{1}' + S^{-1/2}E - I). \tag{7.4}$$

Substituting the definition of β and $\tilde{K}$ and using the spectral decomposition $S = U\delta(\boldsymbol{\lambda})U'$, as we did for the efficient computation of the ETKPF in Section 5.B, we can rewrite Eq. (7.4) as:

$$U\delta(\frac{(k-1)}{(k-1)+\boldsymbol{\lambda}})U' + U\delta(\frac{1}{(k-1)+\boldsymbol{\lambda}})U'c\mathbf{1}' + U\delta(\frac{\sqrt{\boldsymbol{\lambda}}}{(k-1)+\boldsymbol{\lambda}})U'E \tag{7.5}$$

This expression is the analysis ensemble resulting from assimilating the coordinates of the projection of y onto the ensemble space in the basis formed by the background ensemble Y^b. Looking at Eq. (7.5) we recognize that this is nothing else than the W matrix from the stochastic EnKF analysis. Indeed, the eigenvalues in the first element of Eq. (7.5) are the same as $f^{\mu}(\boldsymbol{\lambda})$ in Eq. (5.20) when setting $\gamma = 1$, the eigenvalues in the second elements are the same as $f^{\bar{\mu}}(\boldsymbol{\lambda})$ in Eq. (5.21) when $\gamma = 1$, and the value in the last elements are the same as $\sqrt{f^{\gamma}(\boldsymbol{\lambda})}$ in Eq. (5.22) when $\gamma = 1$. Therefore, the computation of the analysis in ensemble space of Chapter 5 can be reinterpreted geometrically from this perspective.

In Hunt et al. (2007) this geometrical approach is adopted to derive the ETKF analysis. As discussed in Chapter 5, square-root filters separate the analysis into a mean, $\bar{x}^a$, and a matrix of perturbations, $X^a = \boldsymbol{x}^a - \bar{x}^a\mathbf{1}'$ analogous to the background perturbations X^b. The ensemble space analysis of Eq. (7.1) can be split into two parts as

$$\boldsymbol{x}^a = \bar{x}^b\mathbf{1}' + X^bW^{ETKF} = \bar{x}^b\mathbf{1}' + X^bm\mathbf{1}' + X^bW^a,$$

where $\bar{x}^a = \bar{x}^b + X^b m$ and $X^a = X^b W^a$. Then Hunt et al. (2007) use the same geometric approach as above to derive the ETKF directly. Remembering that $\bar{y}^b$ is $\mathbf{0}$ in ensemble space and that the Kalman gain is given by Eq. (7.3), we find that

$$m = \mathbf{0} + \tilde{K}(\beta - \mathbf{0}) = \big((k-1)I + S\big)^{-1} c.$$

Using the spectral decomposition of S again, it is clear that this is the same as the second term in Eq. (7.5). Therefore, we see that the mean part of both the EnKF and ETKF are the same. In a square-root filter X^a is defined such that $\frac{1}{k-1} X^a (X^a)' = P^a$, where P^a is the analysis covariance. Let us rewrite Eq. (1.13) in the more convenient form

$$P^a = \big((P^b)^{-1} + H' R^{-1} H\big)^{-1}.$$

Replacing the elements with the ensemble space equivalents we find

$$\tilde{P}^a = \big((k-1)I + S\big)^{-1}.$$

The ETKF defines the perturbations X^a as the symmetric matrix square-root of $(k-1)P^a$, such that $\frac{1}{k-1} X^a (X^a)' = P^a$. Therefore, using the equivalents in ensemble space and the spectral decomposition of S, W^a is computed as

$$W^a = U\delta\Big(\frac{(k-1)}{(k-1)+\boldsymbol{\lambda}}\Big)^{1/2} U'. \tag{7.6}$$

For more detailed arguments about the derivation of the ETKF see Hunt et al. (2007). We only show this to give an additional geometrical intuition to the interested reader, but do not claim to make this derivation rigorous.

7.2 Ensemble space visualization

There is an interesting possibility to visualize the analysis from the ensemble space formulation of the algorithms (suggested by Andreas Rhodin, DWD), which sheds a new light on the interpretation of the ETKF and illustrates the difference between square-root and stochastic filters. Consider the singular value decomposition (SVD) of the scaled background ensemble perturbations $\tilde{Y}^b = R^{-1/2} Y^b = V\delta(\sqrt{\boldsymbol{\lambda}})U'$, where the scaling by $R^{-1/2}$ ensures that the different observations are comparable. V is the $d \times k$ matrix of left singular vectors (SVs), while U is the $k \times k$ matrix of

right SVs, and $\sqrt{\boldsymbol{\lambda}}$ is the vector of the singular values. Note that U and $\boldsymbol{\lambda}$ are the same as for the spectral decomposition of $S = U\delta(\boldsymbol{\lambda})U'$. The SVD can be used as a dimensionality reduction technique, similar to the use of empirical orthogonal functions in meteorology (Wilks, 2006). For visualization purpose we decide to use the first two SVs to explain the data, so that $\tilde{Y}^b \approx V^{(1,2)}\delta(\sqrt{\boldsymbol{\lambda}^{(1,2)}})(U^{(1,2)})'$, where the $(1,2)$ superscript indicate that we take the first 2 column (or elements). The reconstruction is not perfect if more than two singular values are non-zero, but it is optimal in the sense that it preserves as much of the variance of the data as it is possible with this type of decomposition (see for example Wilks (2006)).

Using this low-dimensional projection, we can represent the background ensemble as a cloud of points in a 2-dimensional space. For the purpose of visualization we can use scaled axes to account for the scaling by $\sqrt{\lambda_i}$. If one is interested more precisely in comparing multiple directions of projections, however, it is then better to fix the scale of the axis (for example in Fig. 7.2). The coordinates of the background particles in this projections are therefore $\beta^{b,j} = (U_{j1}, U_{j2})'$, such that $\tilde{Y}^{b,j} \approx V^{(1,2)}\delta(\sqrt{\boldsymbol{\lambda}^{(1,2)}})\beta^{b,j}$, where the subscript ji means the row j and column i. The percentage of variance explained by each component is computed as $\lambda_i / \sum \lambda_j$ and is shown to give an idea of the approximation involved in the low-dimensional projection.

Now for the analysis, we note the decomposition $\boldsymbol{y}^a = \boldsymbol{y}^b + Y^b m\mathbf{1}' + Y^b W^a$, where W^a depends on the algorithm used. m is the mean shift, such that $\bar{x}^a = \bar{x}^b + X^b m$; it is the same for the EnKF and ETKF, but differs for the EnKPF where it depends on resampling too. For the present visualization we ignore it and look at W^a only. The coordinates of the analysis ensemble in this projection are given by $\beta^{a,j} = (((W^a)'U)_{j1}, ((W^a)'U)_{j2})'$. For better legibility, we can rewrite the first coordinate as $\beta_1^{a,j} = \sum_{i=1}^k W_{ij}^a U_{i1}$.

For the ETKF this leads to a nice interpretation: plugging in the definition of W^a of Eq. (7.6) and remembering that $UU' = I$, we rewrite

$$\beta_1^{a,j} = \Big(\frac{(k-1)}{(k-1)+\lambda_1}\Big)^{1/2} \cdot U_{j1},$$

and $\beta_2^{a,j}$ analogously. Therefore, one can interpret the function of λ in Eq. (7.6) as the constant value by which the analysis scales the background ensemble perturbations to obtain the analysis deviations, in the basis formed by the SVs of $\tilde{Y}^b$.

For the stochastic EnKF the interpretation is a bit more complicated. The

W^a of the EnKF corresponds to the first and last elements in Eq. (7.5). Substituting those in the expression for $\beta^{a,j}$, we obtain the following for the first coordinates:

$$\beta_1^{a,j} = \frac{(k-1)}{(k-1)+\lambda_1} \cdot U_{j1} + \frac{\sqrt{\lambda_1}}{(k-1)+\lambda_1} \cdot \sum_{i=1}^{k} U_{i1} E_{ij}. \tag{7.7}$$

The first part is similar as for the ETKF, except that there is no square-root and thus it scales the coordinates by a smaller factor. The second part is a random combination of the $\beta_1^{b,i}$ coordinates of the whole background ensemble, rescaled by a constant factor. The distribution of this random coordinate is normal with mean 0 and standard deviation $\frac{\sqrt{\lambda_1}}{(k-1)+\lambda_1}$.

It should be noted that for the ETKF the SVs of the analysis ensemble are the same as the SVs of the background ensemble, but this is not true of other assimilation algorithms. Therefore while the visualization shows the direction of greatest explained variance for the ETKF, for other algorithms there are other possible projections in which the analysis ensemble has larger variance. This should be kept in mind when interpreting the visualization, in particular when considering the reduction of spread during the analysis.

In Fig. 7.1 we illustrate with an example the visualization technique and the differences between the stochastic EnKF and the ETKF. The background ensemble of 20 members and the truth were drawn from a 50-dimensional Gaussian with a Gaspari-Cohn covariance structure (Gaspari and Cohn, 1999). The figure shows how both methods do the same thing overall, namely contracting the ensemble towards its mean. This visualization highlights the difference between the stochasticity of the EnKF and the deterministic nature of the ETKF. The ETKF simply scales the coordinates by a constant factor, while the EnKF adds random perturbations to the coordinates.

The total variance in the background ensemble explained by the first two components is 42%. However, only in the case of the ETKF the first singular vector corresponds to the direction of maximal variance in the analysis ensemble. This explains why the analysis ensemble of the EnKF seems less spread compared to that of the ETKF. Depending on the particular random realization of the analysis, however, this effect might not be so clearly visible. In Fig. 7.2 we show the same visualization but for more projections and with fixed axes. By doing so we can now notice that as we consider projections with smaller and smaller singular values the back-

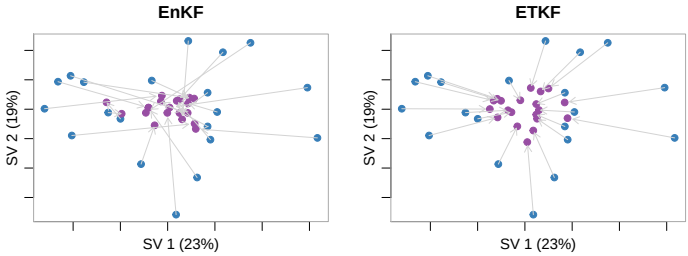

Figure 7.1: Comparison of the stochastic EnKF (left) and the ETKF (right) using the visualization technique described in Section 7.2, where the x-axis is the projection on the first SV, and the y-axis on the second SV. The percentage in the axes labels indicate the percentage of variance explained by each SV. The blue points in the outer rim are the background particles, $\beta^{b,j}$, the purple points towards the center the analysis particles, $\beta^{a,j}$, and the arrows show the transformation from one to another.

ground ensemble is indeed less and less spread and the shrinkage effect of the analysis gets less strong.

For the ETKPF the interpretation is a bit more complex. The part that produces the mean components $\mu^{a,j}$ is equivalent to the deterministic part of the EnKF coordinates in Eq. (7.7) and is given by $f^{\mu}(\lambda_1))U_{j1}$. The value of the function $f^{\mu}(\lambda_1)$ is the same as for the EnKF in the case where $\gamma = 1$ and is equal to 1 for $\gamma = 0$, while in between it varies from one to another monotonously. The resampling step is straightforward and consists in resampling the deterministic part of the coordinates. For the W^{ϵ} component, its effect depends on γ in a non-linear way. If the stochastic version of the EnKPF is used we would get a result similar to the left of Fig. 7.1, but here we focus on the deterministic version of the filter.

In Fig. 7.3 we can see the resulting analysis for four different values of γ, including the extreme case of $\gamma = 1$. In the right column we show a heatmap representation of the W matrix while in the left column we use the visualization technique described above, with the additional feature that we color the individual particles according to which background particle was resampled. For $\gamma = 0$ we can clearly see how the ensemble collapses and only one particle is selected (in pink on the left). For $\gamma = 0.1$ there is a mix of resampling and pulling: some particles are resampled many times, but corrections coming from W^{ϵ} adds some mass on the diagonal, which brings more diversity in the ensemble. Interestingly the particle selected in the case of $\gamma = 0$ is now only selected twice, while other particles (green and blue) are more reused. This comes from the fact that the particles

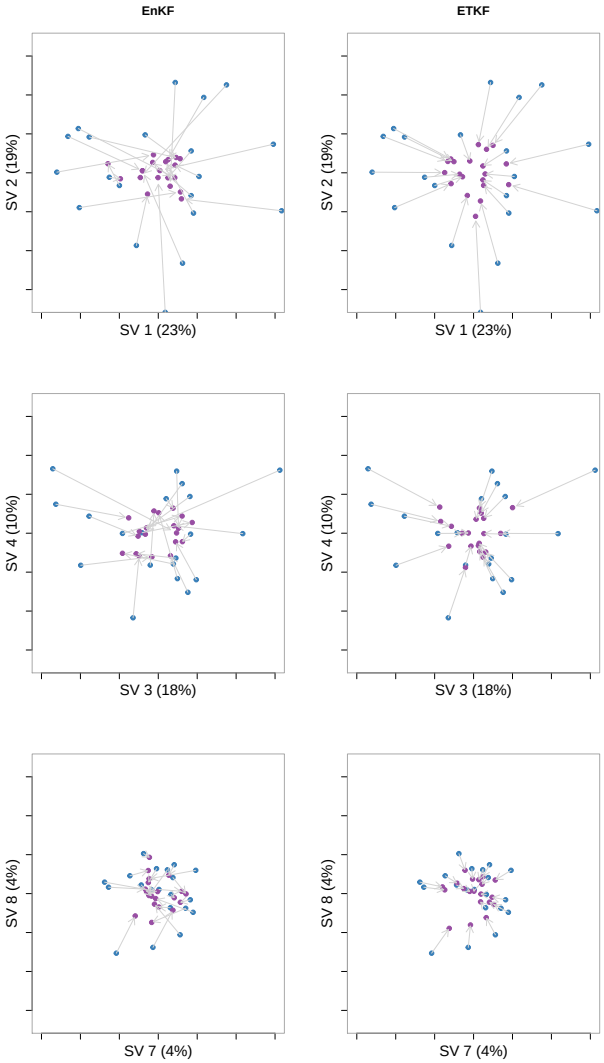

Figure 7.2: Comparison of the stochastic EnKF (left) and the ETKF (right) using fixed scales and more projections, but otherwise the same as Fig. 7.1.

are first pulled before being reweighted, which can change their relative fit to the observation. When $\gamma = 0.25$ the analysis is already very close to what would be obtained with the ETKF, but there is still some resampling going on as is visible in the few off-diagonal elements on the right figure. The final row shows the analysis of the ETKF for comparison, where we see that the weight is more concentrated on the diagonal. It is interesting to notice that the ETKF seems to push the analysis particles away from the center, maintaining thus a greater spread.

So far we have ignored the effect of the observation on the analysis, but let us now look at what happens for different random realization of y. From the decomposition of the analysis as $\boldsymbol{y}^a = \boldsymbol{y}^b + Y^b m \mathbf{1}' + Y^b W^a$ we can find the coordinates of the projection of $\bar{y}^a$ onto the first SV of S as $\bar{\beta}^{a,1} = (m'U)'_1$, and similarly for the other components. In Fig. 7.4 we show the analysis ensemble together with an arrow showing the effect of the analysis on the mean with this technique. As expected, y influences the analysis of the ETKF only through the mean, while the deviation of the particles from their mean is unchanged for different realization of y (compare different columns in Fig. 7.4). The ETKPF, however, is influenced by y both for the mean shift and for the *shape* of the analysis, because of the resampling step (the analysis ensemble is different in each column of Fig. 7.4). One last thing to mention is that the mean shift of the ETKPF is not identical to the one of the ETKF but has a similar orientation for a given realization of y.

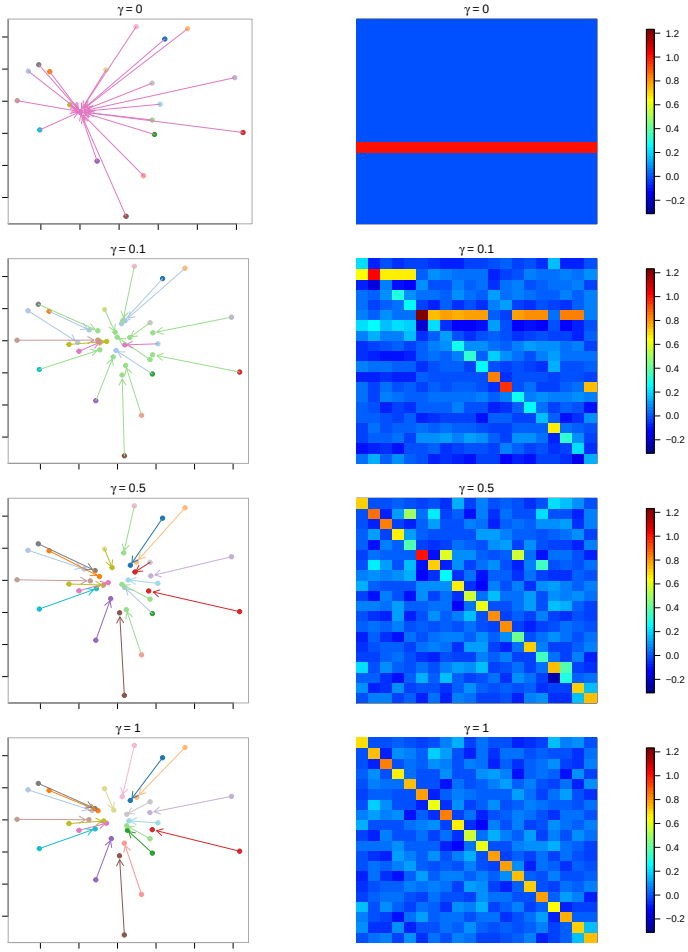

Figure 7.3: Visualization of the ETKPF analysis for different γ in each row. The left column uses the same visualization technique with an additional coloring of the particles. Each background particle has a color, and the analysis particles and arrows take the color of their "parent" during resampling. The right column pictures W with a color scale.

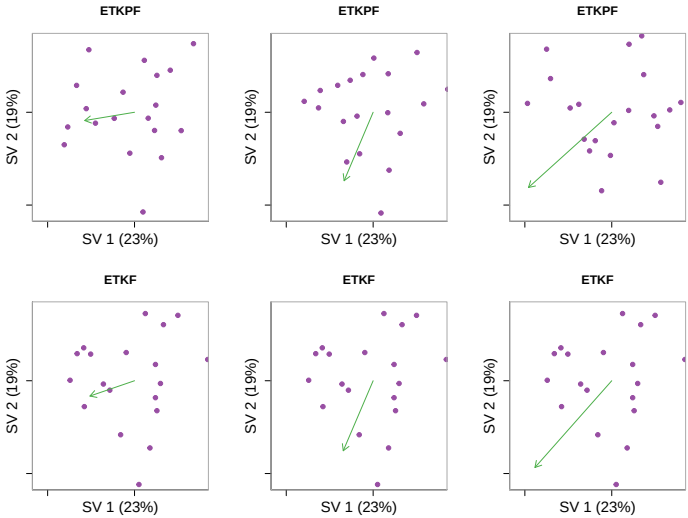

Figure 7.4: Visualization of the influence of different realization of y (each column) on the analysis for the ETKPF (upper row) and the ETKF (lower row). The ETKPF uses $\gamma = 0.1$.

7.3 Optimal choice of γ

The parameter γ determines how much of the analysis should be done with the EnKF, and how much with the PF. However, it is not clear how one should select its value in practice, or even what should be a criterion for its optimal choice in theoretical situations. In the asymptotic case of an infinite ensemble size, the analysis tends to to true posterior only if $\gamma = 0$ or if the background distribution is Gaussian. However, this is of poor guidance to choose γ in the finite sample case, where γ is more of a bias-variance trade-off parameter.

The original proposition for selecting γ in Frei and Künsch (2013) was to target a bracket for the ESS, ensuring that the ensemble does not collapse while trying to stay as close as possible to the PF. However, this criterion is not entirely satisfactory as it leaves open the question of choosing the targeted ESS. In Section 5.3.3 we discussed the issue and proposed as a solution to find the γ such that the MSE of the analysis predictive distribution is minimized. We now review this idea and expand it further by proposing different error measures and alternative estimators. There are three main aspects to take into consideration: first, which error measure or scoring rule we use as objective to minimize; second, how do we compute such a score efficiently in ensemble space, if possible at all; and third, what do we condition on and what do we integrate out to make our estimator more robust and less likely to over-fit the data. We now address each of these points successively in Sections 7.3.1 to 7.3.3, before conducting a simulation study on a simple toy example in Section 7.3.4.

7.3.1 Scoring rule

For more clarity, henceforth we refer to the actual observations as y^o instead of the usual y. Basically we want an objective criterion to evaluate how good our prediction is for a given γ. If we condition on the resampling step (setting the multiplicities N^i), the analysis predictive distribution of the EnKPF is given by

$$\pi^a(y|\gamma) = \sum_{i=1}^{k} \frac{N^i}{k} \mathcal{N}(H\mu^{\gamma,i}, HP^{a,\gamma}H' + R), \tag{7.8}$$

reproduced from Eq. (5.15), but here we removed the conditioning on $\{N^i\}$ for better legibility and made the dependence on γ explicit. Now,

assuming a loss function L, our error in predicting y^o with Eq. (7.8) is the functional $L(y^o, \pi^a(y|\gamma))$. For example, in the case of minimizing the MSE of the ensemble mean that we discussed in Section 5.3.3, this functional is

$$MSE(y^o, \pi^a(y|\gamma)) = ||y^o - E_{\pi^a(y|\gamma)}(y)||_R^2, \tag{7.9}$$

where our loss is $||y||_R^2 = y'R^{-1}y$, the squared Euclidean norm scaled by R^{-1}, which takes care of properly averaging the observations in y. This criterion was selected for convenience and because the MSE of the ensemble mean is often used as a diagnostic in forecast verification. However, this choice might not be the best and other possibilities could be considered.

First, taking the same loss function, an alternative suggested by the formalism exposed here would be to take the MSE (EMSE) over the predictive distribution instead:

$$EMSE(y^o, \pi^a(y|\gamma)) = \int_{-\infty}^{\infty} ||y^o - y||_R^2 \cdot \pi^a(y|\gamma)dy, \tag{7.10}$$

One can decompose this expected score into a squared bias component, equal to the score of Eq. (7.9), and a variance term. Some special care has to be taken though, because of the norm scaled by R. More precisely:

$$EMSE(y^o, \pi^a(y|\gamma)) = ||y^o - E_{\pi^a(y|\gamma)}(y)||_R^2 + E_{\pi^a(y|\gamma)}\big(||y - E_{\pi^a(y|\gamma)}(y)||_R^2\big).$$

Instead of considering the MSE, which is not a strictly proper scoring rule (Gneiting and Raftery, 2007), one could instead use the CRPS. However, we have a vector of observations y and it would not be correct to compute the CRPS component-wise and then to aggregate with something like the Euclidean norm scaled by R^{-1}, as we did for the MSE. Instead, to take into account the joint prediction of all the components of y we should use the Energy Score (ES), a multivariate generalization of the CRPS (Gneiting et al., 2008). In our case it can be written as

$$\begin{aligned} ES(y^o, \pi^a(y|\gamma)) = &\int ||y^o - y||_R \cdot \pi^a(y|\gamma)dy - \\ &\frac{1}{2}\int\int ||y' - y||_R \cdot \pi^a(y|\gamma)\pi^a(y'|\gamma)dydy', \end{aligned}$$

Remembering that we can write the CRPS as

$$\begin{aligned} CRPS(y^o, \pi^a(y|\gamma)) = &\int |y^o - y| \cdot \pi^a(y|\gamma)dy - \\ &\frac{1}{2}\int\int |y' - y| \cdot \pi^a(y|\gamma)\pi^a(y'|\gamma)dydy', \end{aligned}$$

makes the connection between ES and CRPS clear. If y is 1-dimensional, the Euclidean norm is the same as the absolute value, the scaling by R^{-1} does not play a role, and thus both are equivalent. One could pick other error measures to use as scoring rule, but we do not consider any other option in the present work.

7.3.2 Scoring rule in ensemble space

It is one thing to choose an error measure to minimize, such as the MSE or the ES, but it is yet another thing to be able to compute it using only information provided by the data assimilation scheme. In Eq. (5.17) of Section 5.3.3 we showed how one can minimize the MSE of the ensemble mean fully in ensemble space, thus making the criterion cheap to evaluate. For the EMSE in Eq. (7.10) the score can be computed in ensemble space in the same way as Eq. (5.17), but adding an extra term $\frac{1}{k}tr((W^a)'SW^a)$, where W^a is $W^\mu W^\alpha - \frac{1}{k}\cdot W^\mu W^\alpha \mathbf{1}\mathbf{1}' + W^\epsilon$; this term is an estimator of the variance of the predictive distribution scaled by R but in ensemble space.

However, computing the ES in ensemble space is most likely not possible without relying on some approximation. The first reason for this limitation is that to the best of our knowledge there is no analytical formula available for mixture of Gaussians predictive distribution. The second reason is that we do not have a definitive argument to show that considering the projection of y onto the ensemble space leads to the same relative score as in the full space (as is the case for the MSE). Therefore, we propose to use the following CRPS-based score, for which we then proceed to give a justification:

$$\hat{ES}(y^o, \pi^a(y|\gamma)) = \sum_{j=1}^{l} CRPS(\sum_{i=1}^{k} \frac{N^i}{k}\mathcal{N}(\beta_j^{\mu,i}, \lambda_j f^\gamma(\lambda_j) + 1), \beta_j^o), \quad (7.11)$$

where $\beta^{\mu,i} = \delta(\sqrt{\boldsymbol{\lambda}_+})U_+' W^\mu e^i$, e^i is the vector of all 0 except at position i where it is 1, $\beta^o = \delta(\boldsymbol{\lambda}_+^{-1/2})U_+' c$, the columns of U_+ are the eigenvectors of S corresponding to the l strictly positive eigenvalues $\boldsymbol{\lambda}_+$, and the subscripts j indicate the vector components.

The general idea to construct the score in Eq. (7.11) is to express everything in terms of coordinates in ensemble space using an orthonormal basis. We first project y onto this space and then compare its projected value to the predictive distribution. The advantage of using an orthonormal ba-

sis is that we can then neglect covariance terms between components and compute an average of the marginal CRPS.

We want to compare y to the model equivalents $Hx^{b,i}$, which we denote by $y^{b,i}$ and the matrix of their deviations to their mean by Y^b. Let us define a basis for the ensemble space spanned by Y^b as

$$Z^b = Y^b U_+ \delta(\boldsymbol{\lambda}_+^{-1/2}), \tag{7.12}$$

The columns of Z^b are orthonormal with respect to the scalar product scaled by R^{-1}. The number of columns of Z^b is l, which is typically equal to $k-1$ if $k < d$ and the background ensemble has rank k. If $d < k$ then $l = d - 1$, but the arguments below stay the same.

Now we can rewrite the model equivalents $y^{b,i}$ in terms of their coordinates in the basis Z^b as:

$$y^{b,i} = \bar{y}^b + Y^b e^i = \bar{y}^b + Z^b \beta^{b,i}, \tag{7.13}$$

where $\beta^{b,i} = \delta(\boldsymbol{\lambda}_+^{1/2}) U_+' e^i$ are the coordinates of the $y^{b,i}$ in this basis. This is correct because $U_+ U_+' = I - U_- U_-'$ and $Y^b U_- = \mathbf{0}$, where U_- are the eigenvectors of S corresponding to zero eigenvalues of $S = (Y^b)' R^{-1} Y^b$ and are thus orthogonal to Y^b. Using a similar argument, we can rewrite the component means $H\mu^{\gamma,i}$ in the predictive distribution Eq. (7.8) as

$$H\mu^{\gamma,i} = \bar{y}^b + Y^b W^\mu e^i = \bar{y}^b + Z^b \beta^{\mu,i}, \tag{7.14}$$

where $\beta^{\mu,i} = \delta(\boldsymbol{\lambda}_+^{1/2}) U_+' W^\mu e^i$ are the coordinates of $H\mu^{\gamma,i}$ in the basis Z^b.

The first part of the covariance in Eq. (7.8) can be written as

$$HP^{a,\gamma}H' = Z^b \delta(\boldsymbol{\lambda}_+ f^\gamma(\boldsymbol{\lambda}_+))(Z^b)', \tag{7.15}$$

where $f^\gamma(\cdot)$ was defined in Eq. (5.22). The second part of the covariance, R, has rank larger than the ensemble space if $d > k$. However because we restrict our analysis to the ensemble space we can ignore the remaining part and consider only the covariance of a projection of $\xi \sim \mathcal{N}(0, R)$ onto that space, which is nothing other than the identity matrix (see Section 7.1 for detailed reasoning).

The observation y^o can be similarly decomposed into a part which lies in ensemble space, $\bar{y}^b + Z^b \beta^o$, and a part which is orthogonal to it, r^o. More precisely, we can write:

$$y = \bar{y}^b + (y - \bar{y}^b) = \bar{y}^b + Z^b \beta^o + r^o, \tag{7.16}$$

where $\beta^o = \delta(\boldsymbol{\lambda}_+^{-1/2})U_+'c$ are the coordinates of the projection of y^o onto Z^b, and $c = (Y^b)'R^{-1}(y - \bar{y}^b)$. Showing this relies on the simple fact that $S = U_+\delta(\boldsymbol{\lambda}_+)U_+'$.

Expressing everything in terms of coordinates in our new basis, the predictive distribution Eq. (7.8) has the following equivalent in ensemble space

$$\tilde{\pi}^a(\beta|\gamma) = \sum_{i=1}^{k} \frac{N^i}{k} \mathcal{N}\big(\beta^{\mu,i}, \delta(\boldsymbol{\lambda}_+ f^\gamma(\boldsymbol{\lambda}_+) + 1)\big). \tag{7.17}$$

The covariances involved are all diagonal, but some dependencies between dimensions remain due to the resampling of mixture components. As an approximation of a fully multivariate score, we ignore these additional dependencies and compute the sum of the marginal CRPS in this coordinate system Z^b, which results in the score of Eq. (7.11). Furthermore, to compute the CRPS in the case of a Gaussian mixture predictive distribution we can use the analytical formula mentioned in Eq. (5) of Grimit et al. (2006).

7.3.3 Cross-validation and jackknife

The scoring rules that we have proposed so far can be criticized on multiple grounds, the most serious of them being that they are dependent on the observation y^o and might thus lead to over-fitting. This criticism comes from the fact that we are using the observations twice: once for minimizing the score when selecting γ, and once for computing the analysis given the chosen γ.

A first attempt to remedy to this problem is to use a technique akin to the jackknife. More precisely, we hold out each ensemble member one after the other, compute the score based on this truncated ensemble, and then average out. The idea is that by doing so we get rid of some variability due to the conditioning on the ensemble and avoid the situation where we could over-fit the observation with one particular particle being close to y^o. However, recomputing the estimated predictive distribution for each held-out $x^{b,i}$ is costly and inefficient. In the case of the MSE for the ensemble mean of Eq. (7.9), one cheap way to approximate the jackknife estimator would be to take P^b from the whole ensemble and then estimate the MSE

as

$$\hat{MSE}(y^o, \pi^a(y|\gamma)) = \frac{1}{k}\sum_{j=1}^{k} ||y^o - H\bar{x}^{a(-j)}||_R^2,$$

where $\bar{x}^{a(-j)}$ is the estimate of the analysis mean when taking out particle j but using the full covariance P^b. In a sense, what we do here is taking the expectation of the MSE of Eq. (7.9) with respect to the empirical distribution of the background. It it is not clear if this is really the thing that we should do, but it can for sure help to remove some of the variability in the estimate of the MSE as a function of γ.

The jackknife procedure gets rid of some of the problems mentioned above but it still depends heavily on the observation y^o and can lead to overfitting. A common statistical procedure to avoid this issue is cross-validation (CV). However, it cannot be applied directly in our case as we essentially have only one observation y^o. To circumvent this issue, we propose to create surrogate observations

$$y^j = Hx^{b,j} + \eta^j, \quad \eta^j \sim \mathcal{N}(0, R), \tag{7.18}$$

on which we can then apply a cross-validation scheme. By doing so we implicitly assume that the true state that we are trying to estimate and our predictive ensemble are exchangeable. Using this idea, we successively take out each particle j from the ensemble, generate a surrogate data y^j, assimilate it and then compare our prediction to the surrogate value. Let us call $\pi^{a(-j)}(y|\gamma)$ the analysis predictive distribution computed without particle j. Then, we can describe the procedure as follows:

1. For $j = 1, \ldots, k$
   (a) Generate $y^j = Hx^{b,j} + \eta^j$.
   (b) Take $x^{b,j}$ out of the ensemble.
   (c) Assimilate y^j and compute the analysis $\pi^{a(-j)}(y|\gamma)$.
   (d) Compute the score $L(y^j, \pi^{a(-j)}(y|\gamma))$.
2. Average the scores.

We now describe in more detail how one can conduct the computation of the various steps above.

CV computation

The starting point of the CV procedure is to create the surrogate data y^j. To do so we need to generate η^j, which is a d-dimensional vector in observation space. However, we only care about its projection onto the ensemble space. Let us define the following matrix

$$\Pi = Y^b S^{-1} (Y^b)' R^{-1}.$$

Here S^{-1} is the Moore-Penrose inverse $U\delta(\boldsymbol{\lambda}^{-1})U'$ where one sets $0^{-1} = 0$. We could also take the same approach as in Section 7.3.2 and restrict our analysis to the directions associated with strictly positive eigenvalues, but we do it this way for simplicity. The matrix Π projects any vector in observation space onto the subspace spanned by the columns of Y^b. Then, one can decompose η^j as

$$\eta^j = \Pi\eta^j + (I - \Pi)\eta^j =: \hat{\eta}^j + r^j.$$

Because we can pull out from the right of $K(\gamma P^b)$ a factor $(Y^b)'R^{-1}$, $K(\gamma P^b)r^j = \mathbf{0}$. By the same argument, $L^\gamma r^j = \mathbf{0}$, and therefore using η^j or $\hat{\eta}^j$ gives the same analysis. Moreover, because η^j is Gaussian with mean 0 and covariance R, $\hat{\eta}^j$ is also Gaussian with mean 0 and covariance

$$\Pi R \Pi' = Y^b S^{-1} S S^{-1} (Y^b)' = Y^b S^{-1} (Y^b)'.$$

Therefore we can generate $\hat{\eta}^j$ as follows:

1. draw $\epsilon^j \sim \mathcal{N}_k(0, I)$.
2. Set $\xi^j = S^{-1/2}\epsilon^j$.
3. Set $\hat{\eta}^j = Y^b \xi^j$.

Once we have y^j we need to take out $x^{b,j}$ from the ensemble and compute $\pi^{a(-j)}(y|\gamma)$. This is actually the most problematic part, because computing the whole analysis at every step is computationally heavy. One could try to keep the P^b estimated from the whole ensemble, as we proposed for the jackknife, in which case we can compute $\pi^{a(-j)}(y|\gamma)$ relatively cheaply. However, this approach does not work for CV as there is too much information about $x^{b,j}$ contained in P^b, which the EnKF part of the algorithm is able to exploit; it thus leads to large γ values appearing unfairly advantageous.

Therefore, the only possibility is to recompute the analysis at each time step. One way to do so is to define the $(k-1) \times k$ matrix $A^{(-j)}$ such

that $Y^{b(-j)} = Y^b A^{(-j)}$. Basically $A^{(-j)}$ takes off the column j in Y^b and readjust the centering around the ensemble mean without particle j, $\bar{y}^{b(-j)}$. Then we can compute the equivalent of S when j is taken as

$$S^{(-j)} = A^{(-j)'} S A^{(-j)}.$$

To do the same for c we need to see that

$$\bar{y}^{b(-j)} = \bar{y}^b - \frac{1}{k-1} Y^b e^j.$$

Then we can write

$$(y^j - \bar{y}^{b(-j)}) = \frac{k}{k-1} Y^b e^j + \eta^j,$$

from which we obtain

$$c^{(-j)} = A^{(-j)'} S \left(\frac{k}{k-1} e^j + \xi^j\right),$$

where ξ^j was defined above in the algorithm to generate η^j.

Therefore, to compute the analysis distribution $\pi^{a(-j)}(y|\gamma)$, one proceeds as follows:

1. generate $\xi^j = S^{-1/2}\epsilon^j, \quad \epsilon^j \sim \mathcal{N}(0, I)$.
2. compute $S^{(-j)}$ and $c^{(-j)}$.
3. compute the spectral decomposition of $S^{(-j)}$.
4. compute the elements $W^{\mu(-j)}$, $\alpha^{\gamma(-j)}$, etc.
5. compute the score based on these elements.

Depending on which score is used, different elements need to be computed from $\pi^{a(-j)}(y|\gamma)$. For example, for the MSE of the ensemble mean, we only need the vector $m = \frac{1}{k} W^\mu W^\alpha \mathbf{1}$.

The CV scheme described here is interesting from a conceptual point of view, but it is too expensive to compute in practice, particularly because of the high cost necessary to take the spectral decomposition of $S^{(-j)}$ for each j. One way to make it cheaper would be to use B-fold CV instead of leave-one-out CV. Another approach would be to develop some kind of approximation to the CV in the spirit of generalized CV, but it is not clear how one could do so. Finally, there might be some algorithmic shortcut to compute the spectral decomposition of $S^{(-j)}$ from the decomposition of S, but we have not explored this further.

7.3.4 Simulation study

To investigate the abilities of the different schemes proposed above we now conduct a simulation study. Unfortunately, the simulations are not conclusive, but they help us understand some of the limitations of our approach. We consider the following simple example with a q-dimensional bimodal background distribution:

$$x^b \sim 0.5 \cdot \mathcal{N}(\mathbf{0}, I) + 0.5 \cdot \mathcal{N}(\mathbf{3}, I), \tag{7.19}$$

$$H = (e_1, e_2)', \quad R = \sigma^2 I, \quad \sigma^2 = 3, \tag{7.20}$$

where $\mathbf{0}$ and $\mathbf{3}$ are vectors of all 0 and 3 respectively, and e_i is the ith unit vector of 1 in the ith entry and 0 otherwise. As "truth" we set $x^o = \mathbf{0}$ and fix the observations to $y^o = Hx^o$.

The theoretical posterior is a new Gaussian mixture with new component means, covariances and weights which can be computed easily. The scenario is such that the component at $\mathbf{0}$ should be preferred, but the observations are not incompatible with the second mode either and the posterior is bimodal (the component weight of the first mode is approximately 0.75).

We set the ensemble size to $k = 50$ and the state dimension to $q = 100$, and generate a background ensemble from this model. A two-dimensional plot using only the first two coordinates is shown in Fig. 7.5. Additionally, in the left panel we see an ensemble from the theoretical posterior, while in the right panel is the analysis ensemble obtained with the EnKF. It is interesting to notice how the EnKF seems to smear the two modes while the true posterior is more clearly concentrated on the first mode.

For testing our different criteria to choose γ we replicate the following simulation twenty times:

1. Generate a random x^b.
2. For a set of γ on a grid:
   (a) Assimilate the observation with parameter γ.
   (b) Compute the error criterion on x^o, y^o and β^o.

Because we are in a perfect model scenario we can actually compute the error on the truth, x^o, but in practice we are obviously restricted to y^o (or β^o if we want to do everything in ensemble space). As error criterion we will test the MSE, EMSE and ES. We also check empirically that our ensemble space computations in Section 7.3.2 are correct. Furthermore, we consider what happens to the CV and jackknife estimators proposed in

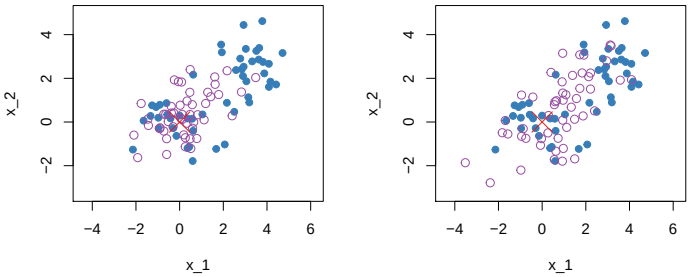

Figure 7.5: Plot of the first two coordinates of a draw from the toy model described in Section 7.3.4. The solid blue dots form the background ensemble, while the empty purple circles are sampled from the theoretical posterior distribution (left panel) or the EnKF analysis distribution (right panel). The red cross marks the observation y^o.

Section 7.3.3. For the ES computed on y^o and x^o we use a Monte Carlo approximation as described in Gneiting et al. (2008) (with one hundred Monte Carlo samples) and adapted to the Gaussian mixture case as in Grimit et al. (2006).

In Fig. 7.6 the three scores evaluated on x^o (top row), on y^o (middle row), and in ensemble space (bottom row), are plotted as a function of γ. Each line is one of the 20 different realizations of the process while the bold blue line is a smoother. It should be noted that in order to compare all the simulations draws in one plot we had to standardize the different curves such that they each have mean 0 and variance 1.

The first thing that we can see from this figure is that all curves have a strong minimum at 0 and then plateau at about $\gamma = 0.25$, whatever the scoring rule used or the underlying component on which it is computed. The second interesting conclusion from this plot is that the ensemble space version of the scores seems to be quite accurate (the score evaluated on y^o or on β^o are very similar), which confirms empirically the correctness of our derivations. The third conclusion is that the MSE and ES produce very similar curves, while the EMSE is slightly different in the sense that it has a tendency to become lower again between $\gamma = 0.25$ and 1. The fact that MSE and ES produce almost the same result when used as score is confirmed in Fig. 7.7 where we see the same figure as in Fig. 6.4 but comparing LETKPF-minES to LETKPF-minMSE. Differences are barely noticeable to the untrained eye.

In Fig. 7.8 we compare the MSE estimated on y^o with the CV and jackknife (JK) estimators (other scores are not shown but the same conclusions can

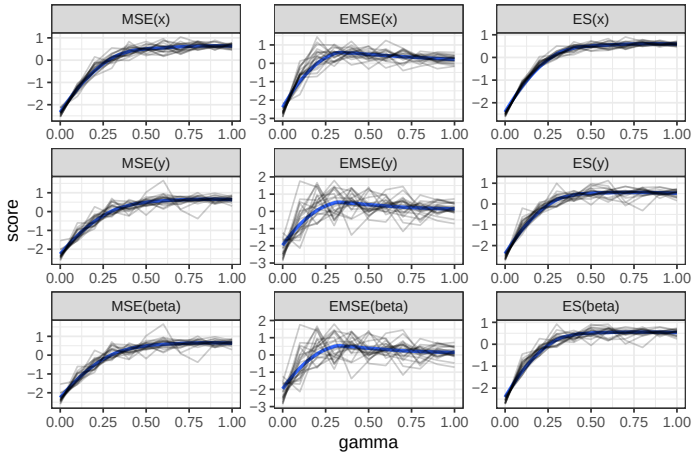

Figure 7.6: Standardized scores as a function of γ for 20 different realizations. In the top row the scores are evaluated on x^o, in the middle row on y^o, and in the bottom row on y^o in ensemble space (β^o).

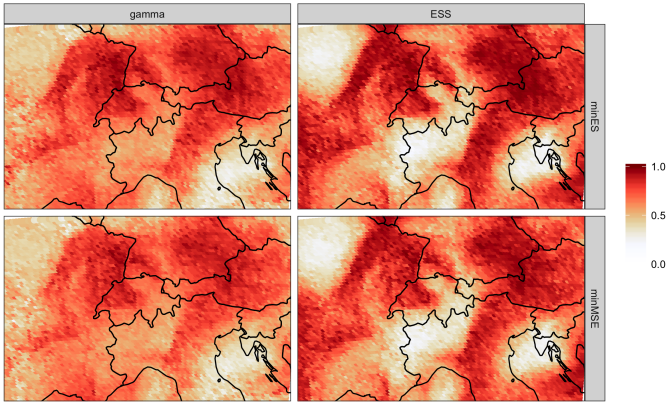

Figure 7.7: Adaptive choice of γ (left panel) and corresponding ESS (right panel) in the lower atmosphere during the case study (as in Fig. 6.4), with LETKPF-minES in the first row and LETKPF-minMSE in the second.

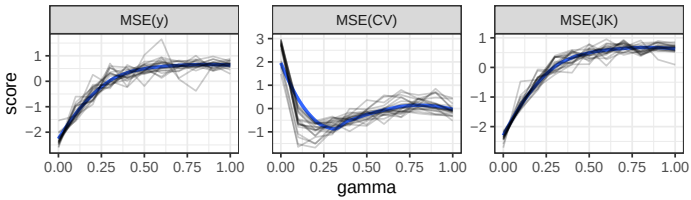

Figure 7.8: CV and jackknife (JK) estimates of MSE compared to the MSE on y^o directly, as a function of γ for 20 different realization. The curves have been standardized to have mean 0 and variance 1.

be drawn from them). The jackknife does not seem to have a strong effect except reducing a little bit the variability of the curves, but the CV leads to a qualitatively different curve with a minimum score achieved slightly away from 0. This might be an indication that indeed minimizing the score directly leads to overfitting, while CV is more conservative and would select a larger γ. On the other hand, it might also be due to the fact that the optimal γ is ultimately dependent on y and that because CV averages out the distribution of y it makes a compromise by choosing a γ of 0.25.

In Fig. 7.9 we compare the scores obtained in the bimodal example here with a reference scenario where we have only one mode at $\mathbf{0}$, everything else being the same. It is interesting to see that in the unimodal case the curve appears much flatter with a tendency towards selecting a γ value of 1, which makes intuitive sense as in this case a pure EnKF should in principle fare well. However in this case the individual curves (not the smoother) are quite noisy and do not follow this pattern as clearly. Furthermore, the curve for the CV error in the unimodal case has a strange bump close to $\gamma = 1$ that should be investigated further.

To conclude, the question of the optimal choice of γ is far from being answered. The score-based approach proposed here is interesting from a theoretical point of view but not completely satisfactory in practice. The underlying difficulty to test our algorithms is that we do not know what the optimal γ should be, even in simple situations. In the present case our score seem to indicate that a γ of 0 would yield the best results, which might be because the example is very clearly non-Gaussian or might be an indication of overfitting. It would be interesting to replicate this simulation for a spectrum of cases from very non-Gaussian to purely Gaussian to check the sensitivity of the different scores. Furthermore, it would be necessary to investigate further if the CV scheme should be preferred and in which

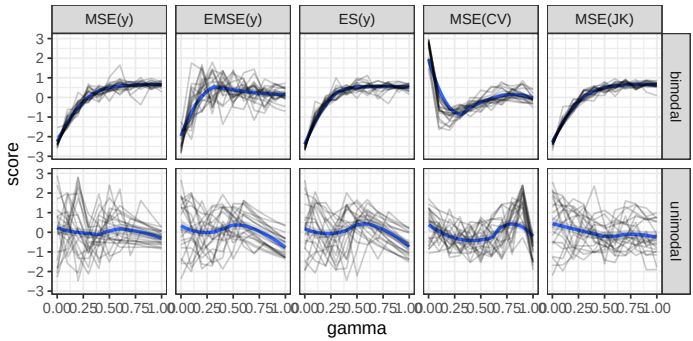

Figure 7.9: Standardized scores as a function of γ for the bimodal case (upper row) and the unimodal case (bottom row).

case if it can be made more efficient for practical applications. In the present section we started to study the optimal choice of γ but much more research needs to be done to understand what the critical parameters that influence γ are, and to develop algorithms to choose its value adaptively in applications.

7.4 Computation and implementation

The implementation of the code in the COSMO assimilation framework was an integral part of this work and played a significant role in the design of the algorithms by setting external constraints. We now give a rough account of the implementation details and development strategy.

The assimilation framework of COSMO is a complex software written in fortran90 for legacy reason. It is actively maintained and developed by the data assimilation team of DWD, in particular Andreas Rhodin, and by other COSMO users around the globe. Describing the full code would be out of the scope of this work and beyond the author's competence, but here are a few elements necessary to understand the context in which our algorithms had to be inserted:

1. Preprocessing of observations based on quality control criteria.
2. Computation of $Hx^{b,j}$ for each particle. Actually a non-linear operator $h(\cdot)$ is applied if available. Furthermore, it is computed at observation time, not at the analysis time, thus making the algorithm 4-dimensional (Hunt et al., 2007).
3. Accept or reject observations according to criteria relying on statistics of $y - Hx^{b,j}$.
4. For each grid point on a coarse grid:
   (a) Compute $S = (Y^b)'R^{-1}Y^b$ and $c = (Y^b)'R^{-1}(y - \bar{y}^b)$. At this stage tapering is applied, with a taper function decreasing with distance of the observation to the grid point under consideration.
   (b) Compute m and W such that $\boldsymbol{x}^a = \bar{x}^b\mathbf{1}' + X^b m\mathbf{1}' + X^b W$.
   (c) Covariance inflation: multiply W by the square-root of ρ.
5. For each grid point on the original fine grid:
   (a) Interpolate m and W from the coarse grid using Akima's interpolation algorithm (Akima, 1970).
   (b) Compute the local analysis $\boldsymbol{x}^a = \bar{x}^b\mathbf{1}' + X^b m\mathbf{1}' + X^b W$.
6. Construct the global analysis by gluing everything together.

Implementing a specific assimilation algorithm such as the ETKPF consists thus in writing the code for the step 4b. As input we have the eigenvectors and eigenvalues of S as well as the vector c, while as output we need to provide the vector m and the matrix W. The separation of the analysis

between mean and perturbations is necessary for the multiplicative covariance inflation scheme in step 4c. More precisely, by multiplying W by $\sqrt{\rho}$, the analysis covariance P^a will be inflated by a factor ρ. The choice of ρ is done adaptively in space and time according to some criterion which we have not considered.

The first stage of the ETKPF computation is to select the parameter γ. As discussed in Section 7.3, to do so we use a grid search to find the γ that fits a given criterion (either a target ESS or a minimal score). For each value of γ tested we typically need to compute $\alpha^{\gamma,i}$, the resampling indices and W^μ. All these quantities can be computed efficiently with the formulae derived in Section 5.B. Then, given a selected γ we compute W^ϵ, either by solving the CARE for the transform filter, or with a random draw for the stochastic EnKPF as in Eq. (5.7). The bottlenecks of the algorithm are the choice of γ (especially if one wants to use cross-validation) and the CARE solver.

In Table 7.1 we compare the computation times of different algorithms. The two variants of the LETKPF are slower than the reference LETKF. On the other hand, the LPF and the two LEnKPF are about twice faster than the LETKF. This highlights that the main computational cost comes from the computation of the square-root part of the filter. The ESS criterion for choosing γ is faster than the minimization of the MSE, because it is implemented as a bisection search (which cannot be done for the MSE minimization as the MSE is not monotonically changing with γ). The computation times for the CV schemes are not included as they are prohibitively expensive in this setup (more than 2 hours).

Therefore, if one was to use these algorithms in practice two options could be considered. First, one could give up the transform version of the filter and use the stochastic LEnKPF-minMSE or LEnKPF-ess50, which are both faster than the LETKF. With the remaining available resources one could choose a finer grid for γ resulting in a more precise estimate or even use a more expensive method such as CV (if improved from the current stage). Second, one could use the LETKPF and accept the higher cost (or improve the CARE solver for the transform filter by using a direct method instead of Newton's algorithm). We have not tried it but it should be easily feasible.

To write the code and build confidence in it, I used the following strategy. First, write the code in R, where it is easy to write tests and simple examples. Second, implement each separate function in fortran and create

Method	CPU time	Percentage relative to LETKF
LETKF	589	100%
LETKPF-ess50	688	117%
LETKPF-minMSE	766	130%
LEnKPF-ess50	259	44%
LEnKPF-minMSE	273	46%
LPF	258	44%

Table 7.1: CPU wall time in seconds and relative to the LETKF for the analysis on 14 June at 1700 UTC with 72 processors.

a R-wrapper to call it from R. Third, for each function create a test to check that the R version and the fortran version return the same result. Using this strategy, it is possible to quickly try new ideas in R and transfer them later in fortran, while at the same time making sure that none of the previous functionalities are broken. Obviously it works all the better if the code is well modularized. In the present case I implemented separate functions to compute the analysis elements ($\alpha^{\gamma,i}, W^{\mu}, W^{\epsilon}$, etc.) and one master function to call everything in order. The code is available on github: `https://github.com/robertsy/ETKPF`.

The implementation of the algorithm would not have been possible without the assistance of Andreas Rhodin from DWD, who modularized the original code such that the core of the assimilation algorithm is well separated from the rest. In this way it is possible to focus on the assimilation algorithm without having to worry about the pre- and post-processing steps. Such a standardization should make it possible to implement alternative algorithms and compare them in a common framework.

Chapter 8

Summary and outlook

The goal of this work was to contribute towards a full solution to the non-Gaussian data assimilation problem in high-dimensional applications. We did so by considering possible extensions to the EnKPF of Frei and Künsch (2013) and by testing the newly proposed algorithms on a hierarchy of models, culminating with a full-scale application to a quasi-operational NWP system.

High-dimensional filtering problems encountered in data assimilation for weather forecast are remarkably challenging. In practice, heuristic but feasible solutions such as the EnKF are typically favored. The PF, on the other hand, is a well-known algorithm which can be in principle applied to any non-Gaussian filtering problem, but which fails utterly in applications with limited ensemble sizes and large state spaces. Hybrid algorithms like the EnKPF try to alleviate some of the issues with the PF by combining it with the EnKF; however, they do not solve the fundamental issue with PFs in high-dimensions.

To address this problem, we proposed two new localized algorithms based on the EnKPF: the naive-LEnKPF and the block-LEnKPF. The first one conducts the analysis locally without taking particular care of the discontinuities introduced by resampling. The block-LEnKPF, on the other hand, assimilates the data by blocks while limiting their influence to local regions and smoothing out the introduced discontinuities. We showed in a simple conjugate normal example how localization can indeed beat the curse of dimensionality, at least superficially. At a deeper level, though, localizing

the update involves a bias-variance trade-off and there is no magical way out of the problem.

To show the feasibility of our algorithms we applied them to a toy model of cumulus convection based on a modified SWEQ. The idea was to have a model of intermediate complexity between the Lorenz96 and a full-scale weather system. The newly proposed algorithms fared well on this setup, with performance similar to the reference LEnKF. The largest improvements were observed for non-Gaussian variables such as rain. In particular, the localized EnKPFs were able to identify dry and wet areas with better accuracy. However, the results obtained with this model have limitations. As we have seen, the behavior of the different algorithms can vary depending on which set of parameters is used for the dynamical model, which reduces the universality of the conclusions reached through the experiments. Furthermore, the example highlighted the appearance of what we called the U-pattern when assimilating high-frequency observations. We showed that this phenomenon was not limited to the modified SWEQ model and that the EnKF was also susceptible to it, even though to a lesser extent. How much this problem can play a role in practice is unclear, but it should be kept in mind and further studied if the possibility of assimilating high-frequency data were to become a reality.

Besides localizing the EnKPF, the second main contribution of this work was to reformulate the algorithm in ensemble space and derive a new transform update scheme. The goal was to emulate the success of the LETKF in practice and to be able to implement our algorithm in the same framework. We showed that the EnKPF could be fully derived in ensemble space if we restrict ourselves to using the empirical covariance as estimate of P^b, which is the same constraint as for the ETKF. The derivation led thus to an efficient way to compute the analysis, which could be easily implemented in the existing data assimilation module of the COSMO model. Developing a deterministic version of the algorithm such that it has the ETKF and not the stochastic EnKF as a limiting case proved harder than expected, but was possible with the use of a continuous algebraic Riccati equation.

The newly proposed LETKPF algorithm was derived and implemented in the COSMO data assimilation framework. We then conducted numerical experiments during a 12-day period of intense convection in June 2015. The algorithm was shown to perform as well as the LETKF, and to be noticeably better for predicting non-Gaussian variables such as wind and hourly precipitation. The results were not completely black or white, but

rather grayish. In practice, which method is better seems to depend greatly on the variable considered, on the pressure level, and on the exact situation at hand. It should be noted that there are many parameters of the analysis that were tuned beforehand to the LETKF, such as the observation covariance matrix, the localization radius, etc. Therefore, in the future one might obtain larger improvements by tuning these various components to the LETKPF. However, before doing so there are some open issues to be dealt with and problems to be better understood.

One of the main questions about the EnKPF that we tried to answer is the question of the optimal choice of γ. Whereas in its original formulation a criterion based on targeting a range of ESS was proposed, we were not satisfied with it and seek a more objective criterion. We investigated the option to minimize a cost function, leading to a score-based criterion. We considered different score functions and various ways to estimate it such that over-fitting of the observations is avoided. The simplest form of our score-based criterion, which consisted in minimizing the MSE of the analysis mean compared to the observations, in ensemble space, was implemented and tested with the COSMO experiments. The results, however, were not clearly for or against it. We investigated in more depth the use of different criteria in a simulation study, but no clear conclusion was reached either. It seems to us that pursuing the question further is of great importance for applying the LETKPF in practice, as the choice of γ clearly has a strong impact on the algorithm performance. In the future, one could imagine choosing γ adaptively not only in space and time, but also for different variables or observations.

Ensemble data assimilation for geophysical applications is an extremely vast topic and we have only scratched the surface of a few problems in this work. For example, throughout this thesis we have assumed that the model that we use is the true model. Obviously, there are many sources of model error and this assumption is far from being correct. In principle one should take into account this model error during the assimilation. The various techniques to increase the spread of the ensemble such as covariance inflation, or RTPP, are one way to tackle this problem. An interesting solution which is easily combined with PFs and our LETKPF is to introduce stochasticity in the parametrization of the dynamical system (Arnold et al., 2013). However, the question of the validity itself of applying filtering methods in the presence of model error can be asked (Smith, 2001). In fact some might argue that a new framework should be used and that model error affects data assimilation in some fundamental ways:

for example the optimal analysis ensemble might be different for different forecast horizons (Smith, 2001). It would be interesting to study this type of questions further, but it was clearly out of the scope of this thesis.

Of more direct implication, a second aspect that we have neglected is the treatment of non-Gaussian observations. The EnKPF somehow alleviates the dependency on Gaussian assumptions for the background distribution, but we always assumed linear and Gaussian observations. One of the key advantages of the PF is that it can be easily applied to any observation distribution. There are some options to adapt the EnKPF to deal with non-Gaussian observations (Frei and Künsch, 2013), but we have not explored them in practice. It would be interesting to try these ideas, as it might be then that the PF would show its full advantage (Poterjoy and Anderson, 2016). A related issue, however, is that often the distribution of the observations is not known, and complicated forward operators linking the model variables to the observations have to be built from first principles. This limits greatly the use of non-conventional observations that are collected through new technologies, such as radars, webcams, smartphone sensors, etc. An interesting approach to use such data would be to learn the relationship between the model and the observations. Essentially we need to model the conditional distribution of y given x, which fits naturally into a regression framework. One could try to learn this relationship from past observations y and use as x the best estimate available from all other sources of classical observations. It is not clear at all if this approach would work in practice, but it would be an interesting path to pursue.

One of the fundamental difficulties when applying PFs to geophysical data assimilation problems is that we are trying to model a complex high-dimensional distribution with a small ensemble. In the usual PF we do not make any additional assumption about this distribution and model it completely non-parametrically, considering it as a mixture of point masses at our background ensemble. At the other end of the spectrum, the EnKF roughly assumes that the prior distribution is Gaussian and uses the ensemble to estimate its mean and covariance. The PF approach is appealing, as it ensures that the analysis ensemble consists of physically possible particles only. The EnKF, on the other hand, might well pull the ensemble members in non-physical regions of the state space. Ultimately, the quality of our sample from the posterior distribution depends on the quality of our estimate of the prior distribution. If we have a large sample compared to the dimensionality of the problem, we can afford to model the prior non-parametrically and the PF works well. In our applications, however,

we cannot afford large ensembles. One interesting alternative approach to improve, or design, assimilation algorithms would be to improve our estimator of the prior distribution.

A natural approach to do so would be to somehow attempt to use information coming from the *climatology*, or long-term distribution of the system. Weather forecast centers have typically spent a lot of time trying to model the so-called B matrix, which is the climatological background covariance matrix. There might be some ways to use this information to improve our estimate of the distribution at a given time. In the Gaussian case it would make sense to simply use the B matrix as some kind of regulator to estimate the background covariance P^b, either in a fully Bayesian framework or more empirically by combining somehow the empirical covariance of the background ensemble with B (Hamill and Snyder, 2000). In the non-parametric setting, a similar idea could be to use not the climatological covariance, but previous realizations of the systems, similar to the idea of *analogues* (Lorenz, 1969, 1982). These analogues could be used to form a kind of prior distribution for the prior, in other words a hyperprior. In principle one could thus estimate the prior at analysis time by combining this hyperprior with the current background ensemble, and then update it with the new observations. For the background ensemble to play this role of data one would need to have some kind of way to connect the prior to the hyperprior, which is far from obvious. However, this line of research may prove interesting and suggest some new algorithms for non-Gaussian data assimilation.

Another approach to improve our estimate of the prior distribution would be to make some additional assumptions about it. To do so one could draw some ideas from what people do in forecast verification, where one has to construct a probabilistic forecast from an ensemble by using post-processing techniques (Gneiting and Katzfuss, 2014; Weigel, 2011; Gneiting and Raftery, 2005). For example, to estimate the probability distribution of precipitation at a site one can use the distribution at this site and its neighbors. This is to account for the fact that sometimes the ensemble predicts the correct precipitation cells but misplaces them. One could imagine using a similar idea to construct an estimate of the marginal distribution at each site, by pooling the information coming from neighbors. Doing so would in effect increase the ensemble size available locally. Of course the samples would not be independent and it is not clear at all how one should take this into account properly, let alone the problem of modeling the joint distribution at different sites. However, the idea is somehow appealing and

might lead to some new algorithms if developed to maturity.

To conclude, developing algorithms for non-Gaussian high-dimensional data assimilation is fantastically exciting and interesting, but also extremely challenging. In this work, we presented a few contributions towards a full solution to this problem, but the way ahead is long and promises to be rigged with difficulties. Hopefully some of the ideas and experiments conducted during the course of this work will be helpful and valuable to other researchers in this long-term journey.

List of figures

Introduction

Localization

Ensemble Space

Bibliography

Ades, M. and P. J. van Leeuwen (2013). An exploration of the equivalent weights particle filter. *Quarterly Journal of the Royal Meteorological Society 139*(672), 820–840.

Akima, H. (1970). A new method of interpolation and smooth curve fitting based on local procedures. *Journal of the ACM (JACM) 17*(4), 589–602.

Alspach, D. L. and H. W. Sorenson (1972). Nonlinear Bayesian estimation using Gaussian sum approximations. *Automatic Control, IEEE Transactions on 17*(4), 439–448.

Arnold, H. M., I. M. Moroz, and T. N. Palmer (2013). Stochastic parametrizations and model uncertainty in the Lorenz '96 system. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences 371*(1991), 20110479.

Baldauf, M., A. Seifert, J. Foerstner, D. Majewski, and M. Raschendorfer (2011). Operational convective-scale numerical weather prediction with the COSMO model: Description and sensitivities. *Monthly Weather Review 142*, 3887–3905.

Bartels, R. H. and G. W. Stewart (1972). Solution of the matrix equation AX+ XB= C [F4]. *Communications of the ACM 15*(9), 820–826.

Bauer, P., A. Thorpe, and G. Brunet (2015). The quiet revolution of numerical weather prediction. *Nature 525*(7567), 47–55.

Bengtsson, T., P. Bickel, and B. Li (2008). Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In D. Nolan and T. Speed (Eds.), *Probability and Statistics: Essays in Honor of David A. Freedman*, Volume 2, pp. 316–334.

Bengtsson, T., C. Snyder, and D. Nychka (2003). Toward a nonlinear ensemble filter for high-dimensional systems. *Journal of Geophysical Research: Atmospheres 108*(D24).

Bickel, P., B. Li, and T. Bengtsson (2008). Sharp failure rates for the bootstrap particle filter in high dimensions. In B. Clarke and S. Ghosal (Eds.), *Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*, Volume 3, pp. 318–329.

Bickel, P. J. and E. Levina (2008). Regularized estimation of large covariance matrices. *The Annals of Statistics 36*(1), 199–227.

Bishop, C. H., B. J. Etherton, and S. J. Majumdar (2001). Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Monthly Weather Review 129*(3), 420–436.

Bishop, G. and G. Welch (2001). An introduction to the Kalman filter. *Proc of SIGGRAPH, Course 8*, 27599–3175.

Carpenter, J., P. Clifford, and P. Fearnhead (1999). Improved particle filter for nonlinear problems. *IEE Proceedings - Radar, Sonar and Navigation 146*(1), 2–7.

Chustagulprom, N., S. Reich, and M. Reinhardt (2016). A hybrid ensemble transform particle filter for nonlinear and spatially extended dynamical systems. *SIAM/ASA Journal on Uncertainty Quantification 4*(1), 592–608.

Crisan, D. (2001). Particle filters — a theoretical perspective. In A. Doucet, N. d. Freitas, and N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, pp. 17–41. Springer New-York.

de Wiljes, J., W. Acevedo, and S. Reich (2016). Second-order accurate ensemble transform particle filters. *arXiv:1608.08179 [math]*. arXiv: 1608.08179.

Desroziers, G., L. Berre, B. Chapnik, and P. Poli (2005). Diagnosis of observation, background and analysis-error statistics in observation space. *Quarterly Journal of the Royal Meteorological Society 131*, 3385–3396.

Doucet, A., N. De Freitas, and N. Gordon (2001). *Sequential Monte Carlo Methods in Practice*. Springer New-York.

Evensen, G. (1994). 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.

Evensen, G. (2003). The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics 53*(4), 343–367.

Evensen, G. (2009). *Data Assimilation: The Ensemble Kalman Filter*. Springer New-York.

Fortin, V., M. Abaza, F. Anctil, and R. Turcotte (2014). Why should ensemble spread match the rmse of the ensemble mean? *Journal of Hydrometeorology 15*(4), 1708–1713.

Frei, M. and H. R. Künsch (2013). Bridging the ensemble Kalman and particle filters. *Biometrika 100*(4), 781–800.

Furrer, R., M. G. Genton, and D. Nychka (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics 15*(3), 502–523.

Gaspari, G. and S. E. Cohn (1999). Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society 125*(554), 723–757.

Gneiting, T. and M. Katzfuss (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application 1*(1), 125–151.

Gneiting, T. and A. E. Raftery (2005). Weather forecasting with ensemble methods. *Science 310*(5746), 248–249.

Gneiting, T. and A. E. Raftery (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association 102*(477), 359–378.

Gneiting, T., L. I. Stanberry, E. P. Grimit, L. Held, and N. A. Johnson (2008). Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds. *Test 17*(2), 211–235.

Gordon, N., D. Salmond, and A. Smith (1993). Novel approach to nonlinear non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F 140*(2), 107–113.

Gottwald, G. A. and A. J. Majda (2013). A mechanism for catastrophic filter divergence in data assimilation for sparse observation networks. *Nonlinear Processes in Geophysics 20*(5), 705–712.

Grimit, E. P., T. Gneiting, V. J. Berrocal, and N. A. Johnson (2006). The continuous ranked probability score for circular variables and its application to mesoscale forecast ensemble verification. *Quarterly Journal of the Royal Meteorological Society 132*(621C), 2925–2942.

Hamill, T. M. and C. Snyder (2000). A hybrid ensemble Kalman filter–3d variational analysis scheme. *Monthly Weather Review 128*(8), 2905–2919.

Hamill, T. M., J. S. Whitaker, and C. Snyder (2001). Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Monthly Weather Review 129*(11), 2776–2790.

Harnisch, F. and C. Keil (2015). Initial conditions for convective-scale ensemble forecasting provided by ensemble data assimilation. *Monthly Weather Review 143*(5), 1583–1600.

Haslehner, M., T. Janjić, and G. C. Craig (2016). Testing particle filters on simple convective-scale models. part 2: A modified shallow-water model. *Quarterly Journal of the Royal Meteorological Society 142*(697), 1628–1646.

Hastie, T. J. and R. J. Tibshirani (1990). *Generalized additive models*, Volume 43. CRC press.

Houtekamer, P. L. and H. L. Mitchell (1998). Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review 126*(3), 796–811.

Houtekamer, P. L. and H. L. Mitchell (2001). A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review 129*(1), 123–137.

Houtekamer, P. L. and H. L. Mitchell (2005). Ensemble kalman filtering. *Quarterly Journal of the Royal Meteorological Society 131*(1), 3269—-3289.

Hunt, B. R., E. J. Kostelich, and I. Szunyogh (2007). Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena 230*(1-2), 112–126.

Jolliffe, I. T. and D. B. Stephenson (2003). *Forecast verification: a practitioner's guide in atmospheric science*. John Wiley & Sons.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering 82*(1), 35–45.

Kalman, R. E. and R. S. Bucy (1961). New results in linear filtering and prediction theory. *Journal of Basic Engineering 83*(1), 95–108.

Künsch, H. R. (2001). State space and hidden markov models. In O. Barndorff-Nielsen, D. Cox, and C. Klüppelberg (Eds.), *Complex stochastic systems*, Volume 87, Chapter 3, pp. 109–173. CRC Press.

Künsch, H. R. (2005). Recursive Monte Carlo filters: Algorithms and theoretical analysis. *The Annals of Statistics 33*(5), 1983–2021.

Lancaster, P. and L. Rodman (1995). *Algebraic Riccati Equations*. Oxford science publications. Clarendon Press.

Lei, J. and P. Bickel (2011). A moment matching ensemble filter for nonlinear non-Gaussian data assimilation. *Monthly Weather Review 139*(12), 3964–3973.

Lewis, J. M. (2005). Roots of ensemble forecasting. *Monthly Weather Review 133*(7), 1865–1885.

Li, H., E. Kalnay, and T. Miyoshi (2009). Simultaneous estimation of covariance inflation and observation errors within an ensemble kalman filter. *Quarterly Journal of the Royal Meteorological Society 135*, 523–533.

Liu, J. and M. West (2001). Combined parameter and state estimation in simulation-based filtering. In A. Doucet, N. d. Freitas, and N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, pp. 197–223. Springer New York.

Liu, J. S. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing 6*(2), 113–119.

Lorenz, E. (1982). Atmospheric predictability experiments with a large numerical model. *Tellus 34*(6), 505–513.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences 20*(2), 130–141.

Lorenz, E. N. (1969). Atmospheric predictability as revealed by naturally occurring analogues. *Journal of the Atmospheric sciences 26*(4), 636–646.

Lorenz, E. N. and K. A. Emanuel (1998). Optimal sites for supplementary weather observations: Simulation with a small model. *Journal of the Atmospheric Sciences 55*(3), 399–414.

Musso, C., N. Oudjane, and F. Le Gland (2001). Improving regularised particle filters. In A. Doucet, N. De Freitas, and N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice*, pp. 247–271. Springer New-York.

Nakano, S. (2014). Hybrid algorithm of ensemble transform and importance sampling for assimilation of non-Gaussian observations. *Tellus A 66*(1), 21429.

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

Papadakis, N., E. Memin, A. Cuzol, and N. Gengembre (2010). Data assimilation with the weighted ensemble Kalman filter. *Tellus A 62A*, 673–697.

Pitt, M. K. and N. Shephard (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association 94*(446), 590–599.

Poterjoy, J. (2016). A localized particle filter for high-dimensional nonlinear systems. *Monthly Weather Review 144*(1), 59–76.

Poterjoy, J. and J. L. Anderson (2016). Efficient assimilation of simulated observations in a high-dimensional geophysical system using a localized particle filter. *Monthly Weather Review 144*(5), 2007–2020.

Rebeschini, P. and R. v. Handel (2015). Can local particle filters beat the curse of dimensionality? *The Annals of Applied Probability 25*(5), 2809–2866.

Reich, S. (2013a). A guided sequential Monte Carlo method for the assimilation of data into stochastic dynamical systems. In A. Johann, H.-P. Kruse, F. Rupp, and S. Schmitz (Eds.), *Recent Trends in Dynamical Systems*, Number 35 in Springer Proceedings in Mathematics & Statistics, pp. 205–220. Springer Basel.

Reich, S. (2013b). A nonparametric ensemble transform method for Bayesian inference. *SIAM Journal on Scientific Computing 35*(4), A2013–A2024.

Robert, S. and H. R. Künsch (2017a). Localization in high-dimensional Monte Carlo filtering. In R. Argiento, E. Lanzarone, I. A. Villalobos, and A. Mattei (Eds.), *Bayesian Statistics in Action*, Volume 194, Chapter 8, pp. 79–89. Springer Proceedings in Mathematics and Statistics.

Robert, S. and H. R. Künsch (2017b). Localizing the ensemble Kalman particle filter. *Tellus A: Dynamic Meteorology and Oceanography 69*(1), 1–14.

Robert, S., D. Leuenberger, and H. R. Künsch (2017). A local ensemble transform Kalman particle filter for convective scale data assimilation. *Quarterly Journal of the Royal Meteorological Society*. accepted.

Sakov, P. and L. Bertino (2011). Relation between two common localisation methods for the enkf. *Computational Geosciences 15*(2), 225–237.

Schraff, C., H. Reich, A. Rhodin, A. Schomburg, K. Stephan, A. Perianez, and R. Potthast (2016). Kilometre-scale ensemble data assimilation for the COSMO model (KENDA). *Quarterly Journal of the Royal Meteorological Society 142*, 1453–1472.

Sigrist, F., H. R. Künsch, and W. A. Stahel (2012). A dynamic nonstationary spatio-temporal model for short term prediction of precipitation. *The Annals of Applied Statistics 6*(4), 1452–1477.

Smith, L. A. (2001). Disentangling uncertainty and error: On the predictability of nonlinear systems. In A. I. Mees (Ed.), *Nonlinear Dynamics and Statistics*, pp. 31–64. Birkhäuser Boston.

Snyder, C., T. Bengtsson, P. Bickel, and J. Anderson (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review 136*(12), 4629–4640.

Snyder, C., T. Bengtsson, and M. Morzfeld (2015). Performance bounds for particle filters using the optimal proposal. *Monthly Weather Review 143*(11), 4750–4761.

Stidd, C. K. (1973). Estimating the precipitation climate. *Water Resources Research 9*(5), 1235–1241.

Tippett, M. K., J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker (2003). Ensemble square root filters. *Monthly Weather Review 131*(7), 1485–1490.

van Leeuwen, P. J. (2009). Particle filtering in geophysical systems. *Monthly Weather Review 137*(12), 4089–4114.

van Leeuwen, P. J. (2010). Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society 136*(653), 1991–1999.

Weigel, A. P. (2011). Ensemble forecasts. In I. T. Jolliffe and D. B. Stephenson (Eds.), *Forecast Verification*, pp. 141–166. John Wiley & Sons, Ltd.

Whitaker, J. S. and T. M. Hamill (2002). Ensemble data assimilation without perturbed observations. *Monthly Weather Review 130*(7), 1913–1924.

Wikle, C. K. and L. M. Berliner (2007). A Bayesian tutorial for data assimilation. *Physica D: Nonlinear Phenomena 230*(1–2), 1–16.

Wilks, D. S. (2006). *Statistical methods in the atmospheric sciences* (2nd ed ed.). Number v. 91 in International geophysics series. Amsterdam ; Boston: Academic Press.

WMO (2015). *Manual on the Global Observing System, Volume I.* World Meteorological Organization No. 544.

Würsch, M. (2014). *Testing data assimilation methods in idealized models of moist atmospheric convection.* Ph. D. thesis, Ludwig-Maximilians-Universität München.

Würsch, M. and G. C. Craig (2014). A simple dynamical model of cumulus convection for data assimilation research. *Meteorologische Zeitschrift 23*(5), 483–490.

Zeng, Y. and T. Janjić (2016). Study of conservation laws with the local ensemble transform Kalman filter. *Quarterly Journal of the Royal Meteorological Society 142*(699), 2359–2372.

Zhang, F., C. Snyder, and J. Sun (2004). Impacts of initial estimate and observation availability on convective-scale data assimilation with an ensemble kalman filter. *Monthly Weather Review 132*, 1238–1253.