Efficient Implementation of Ensemble-Based Methods in Data Assimilation

Elias David Nino Ruiz

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Adrian Sandu, Chair
Calvin J. Ribbens
Scott C. Leman
Yang Cao
Jeffrey L. Anderson

12/21/2015
Blacksburg, Virginia

Keywords: Ensemble-based methods, ensemble Kalman filter, ensemble square root filter, hybrid data assimilation, background error covariance matrix estimation, parallel data assimilation

Efficient Implementation of Ensemble-Based Methods in Data Assimilation

Elias David Nino Ruiz

(ABSTRACT)

Ensemble-based methods have gained widespread popularity in the field of data assimilation. An ensemble of model realizations encapsulates information about the error correlations driven by the physics and the dynamics of the numerical model. This information can be used to obtain improved estimates of the state of non-linear dynamical systems such as the atmosphere and/or the ocean. This work develops efficient ensemble-based methods for data assimilation.

A major bottleneck in ensemble Kalman filter (EnKF) implementations is the solution of a linear system at each analysis step. To alleviate it an EnKF implementation based on an iterative Sherman Morrison formula is proposed. The rank deficiency of the ensemble covariance matrix is exploited in order to efficiently compute the analysis increments during the assimilation process. The computational effort of the proposed method is comparable to those of the best EnKF implementations found in the current literature. The stability analysis of the new algorithm is theoretically proven based on the positiveness of the data error covariance matrix.

In order to improve the background error covariance matrices in ensemble-based data assimilation we explore the use of shrinkage covariance matrix estimators from ensembles. The resulting filter has attractive features in terms of both memory usage and computational complexity. Numerical results show that it performs better that traditional EnKF formulations.

In geophysical applications the correlations between errors corresponding to distant model components decreases rapidly with the distance. We propose a new and efficient implementation of the EnKF based on a modified Cholesky decomposition for inverse covariance matrix estimation. This approach exploits the conditional independence of background errors between distant model components with regard to a predefined radius of influence. Consequently, sparse estimators of the inverse background error covariance matrix can be obtained. This implies huge memory savings during the assimilation process under realistic weather forecast scenarios. Rigorous error bounds for the resulting estimator in the context of data assimilation are theoretically proved. The conclusion is that the resulting estimator converges to the true inverse background error covariance matrix when the ensemble size is of the order of the logarithm of the number of model components.

We explore high-performance implementations of the proposed EnKF algorithms. When the observational operator can be locally approximated for different regions of the domain, efficient parallel implementations of the EnKF formulations presented in this dissertation can be obtained. The parallel computation of the analysis increments is performed making use of domain decomposition. Local analysis increments are computed on (possibly) different processors. Once all local analysis increments have been computed they are mapped back onto the global domain to recover the global analysis. Tests performed with an atmospheric general circulation model at a T-63 resolution, and varying the number of processors from 96 to 2,048, reveal that the assimilation time can be decreased multiple fold for all the proposed EnKF formulations.

Ensemble-based methods can be used to reformulate strong constraint four dimensional variational data assimilation such as to avoid the construction of adjoint models, which can be complicated for operational models. We propose a trust region approach based on ensembles in which the analysis increments are computed onto the space of an ensemble of snapshots. The quality of the resulting increments in the ensemble space is compared against the gains in the full space. Decisions on whether accept or reject solutions rely on trust region updating formulas. Results based on a atmospheric general circulation model with a T-42 resolution reveal that this methodology can improve the analysis accuracy.

This work was supported in part by awards NSF CCF–1218454, AFOSR FA9550–12–1–0293–DEF, and by the Computational Science Laboratory at Virginia Tech.

Dedication

To My Lord Jesus Christ

Acknowledgments

I would like to thank to my wife Maria Ines and daughters Maria Gabriela and Maria Daniela for their love through these years. Also, many thanks to my parents Elias and Arely, and sister Carmen Judith for their continuous support and motivation. To my academical advisor, Dr. Adrian Sandu, for this unique opportunity. Lastly, thanks to my committee members Dr. Calvin Ribbens, Dr. Scotland Leman, Dr. Yang Cao and Dr. Jeffrey Anderson for their support and help through my Ph.D. studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

State estimation of highly non-linear dynamical systems is of great importance many fields of science. For instance, in weather forecasting, one wants to estimate the state of the atmosphere in order to prevent natural disasters, to schedule agriculture processes, and to avoid crosswinds in the aviation context. Along the years the scientific community has developed numerical models which mimic the real behavior of complex systems such as the ocean and/or the atmosphere. Typically, physics simplifications and dynamics reductions are employed when numerical models are developed. The underlying equations are initial-value problems where the exact initial conditions are unknown. Consequently, the forecast of numerical models can poorly estimate the real state of a system for long simulations. This problem can be mitigated by injecting observations from the real dynamics of the system into the predictions of imperfect numerical models.

Data assimilation is the a process in which real noisy observations are incorporated into the forecast of an imperfect numerical model in order to adjust the model trajectory according to the real state of a system. In this context, errors in the forecast and the observations are treated as random variables, typically with Gaussian distributions. In the general framework, one assumes that noisy observations are provided with the corresponding data error distributions, while the moments of the background error distribution associated with the model state must be estimated. *Background error estimation* has been an active field in the context of data assimilation for several decades. In practice an ensemble of model realizations can be used in order to estimate the moments of the underlying background error distribution. The number of model realizations is constrained by computational resources; typically this number does not exceed the hundreds, while the dimension of the state space is in the order of millions. Consequently, the moments of the ensemble can poorly estimate the unknown moments of the underlying error distribution, leading to poor forecasts of the system of interest. The statistical literature seeks robust approaches in order to deal with this situation, for instance, one can choose covariance matrix estimators with better asymptotic properties than the ensemble covariance matrix. However, some existing estimators rely on computing explicitly high-dimensional covariance matrices, while others depend on having large numbers of samples with respect to the dimension of the probability distribution. Both approaches are impractical for realistic weather forecast scenarios.

We have identified several opportunities to enhance the state of the art in ensemble-based data assimilation methodologies. They are summarized below.

1. In sequential data assimilation methods such as the ensemble Kalman filter suffer from sampling errors due to the dimension of the underlying background error distribution. Several methodologies are proposed in the current literature in order to deal with this situation. Localization methods are commonly used in practice. In this approach model components are surrounded by local boxes that are assimilated independently. The size of the local box has considerable impact on the quality of the analyses. Large local boxes typically yield to poor analysis corrections. The size of the local box is driven by two aspects, the correlations of observed components and the type of observational operator. In operational data assimilation one can find data error correlations on large sub-domain sizes. In such cases it is desirable to have local estimates of background error correlations that are not sensitive to the size of the local box. The local analyses should be inexpensive, and lead to an accurate global analysis. The new filters based on shrinkage covariance matrix estimators and on modified Cholesky decomposition are developed herein to achieve this purpose.

2. The data assimilation algorithms based on local approximations of the observational operator and/or error correlations should be parallelizable. This is important not only to speedup the assimilation process, but also to have enough memory to distribute ensemble members and/or sub-domains across different processors. Our parallel methodologies developed herein address this point.

3. In strong-constraint four dimensional variational data assimilation implementations ensembles can be used to approximate local sub-spaces consistent with the background error distribution. This avoids the need for adjoint models that are difficult to implement. The optimal analysis increments are recovered from the weights computed onto the ensemble space. Current methodologies assume that the ensemble spaces capture the dominant directions of background errors. We propose to use of the trust region framework in order to corroborate that such assumption is true.

1.1 Research Objectives

The overall goal of this research is to push forward the state of the art in ensemble-based data assimilation methodologies.

Since spurious correlations do impact the quality of the analyses we need better estimates of the background error correlations. *The first objective of this research is to develop an* *efficient implementation of the ensemble Kalman filter based on shrinkage covariance matrix estimation.*

One attractive feature of the ensemble covariance matrix is its flow-dependency that captures the dynamics of the model. Due to physical properties one expects a rapid decay of background error correlations between spatially remote model variables such as temperature, specific humidity, and wind components. For low-rank ensemble covariance matrices localization methods are used to mimic this decay. The modified Cholesky decomposition exploits this kind of natural decorrelation and provides robust (asymptotically with regard to the ensemble size) estimators of the inverse covariance matrix. *Our second objective is to develop an efficient implementation of the ensemble Kalman filter based on a modified Cholesky decomposition for inverse covariance matrix estimation.* Furthermore, we want to study theoretically the asymptotic properties of the modified Cholesky decomposition for inverse covariance matrix estimation in the context of ensemble-based assimilation.

Since the analysis must be computed efficiently, *the third objective of this research is to develop an efficient implementation of the ensemble Kalman filter based on an iterative Sherman Morrison formula.* This method bounds linearly the computational effort of the analysis step in the ensemble Kalman filter formulation with regard to the model dimension. Furthermore, the iterative Sherman Morrison formula can be easily extended to other ensemble-based formulations.

Whenever possible, parallel resources should be exploited for speeding up the computations and for allowing large memory volumes to represent model realizations. *Our fourth objective is to develop efficient parallel implementations of the ensemble Kalman filter based on shrinkage covariance matrix estimation and on modified Cholesky decomposition for inverse covariance matrix estimation.*

The capabilities of ensemble-based methods can be extended to four dimensional approximations. *Our last objective is to provide an efficient implementation of a four dimensional ensemble Kalman smoother method based on a trust region framework.*

1.2 Research Accomplishments

This research accomplishment of the current dissertation are highlighted next:

1. An ensemble Kalman filter based on an iterative Sherman Morrison formula is proposed [NRSA14, NSA12]. The computational effort of this method is comparable to those of the best EnKF implementations found in the literature. The stability of the method is theoretically proved. No matrix decompositions are performed by this methodology, unlike what is usually required by current implementations.

2. An ensemble Kalman filter formulation based on shrinkage covariance matrix estima-

tion is proposed [NRS15]. The methodology exploits the rank deficiency of the sample covariance matrix in order to obtain an estimator that is computationally and memory inexpensive. Several experiments reveal that this background estimator increases the degrees of freedom of the ensemble consistently. The computational effort of this method is linearly bounded with regard to the number of observed components and the model dimension. When observational operators can be locally approximated the use of parallel resources provides improvements to the method in terms of efficiency.

3. We exploit the (expected) natural decorrelation of errors in distant model components for some model variables such as wind-components, specific humidity and temperature. We propose an efficient implementation of the ensemble Kalman filter based on a modified Cholesky decomposition for inverse covariance matrix estimation [NRSD]. This estimator exploits the decorrelation of model components based on their physical distance. We prove theoretically the convergence of the estimator in the context of data assimilation when background and observational errors are Gaussian. The computational effort of this method is bounded linearly with respect to the model dimension.

4. We also propose a four dimensional ensemble-based method in the trust region framework [NRS16, NS15]. The method avoids the need for adjoint models, and offers an efficient approach to compute the analysis increments in the ensemble space by comparing the improvement of the cost function in the ensemble and model spaces. The computational effort is similar to those of the best ensemble four dimensional data assimilation methods found in the literature, but it requires an additional model propagation during the assimilation process.

1.3 Dissertation Layout

This dissertation consist of seven chapters, as summarized below.

In Chapter 2 the general theory of data assimilation is presented. Sequential and variational methods from the literature are discussed. The ensemble Kalman filter formulation and its variations are detailed. Efficient EnKF implementations are studied as well. Covariance localization and domain localization are presented as alternatives to background covariance matrix estimation. The impact of spurious correlations is introduced. The extension of ensemble-based methods to the four dimensional case is discussed as well.

In Chapter 3 we propose an efficient implementation of the ensemble Kalman filter based on iterative Sherman Morrison formula. It exploits the rank-deficiency of the ensemble covariance matrix in order to bring an efficient implementation which scales linear with regard to the number of observed components as well as the model dimension. The stability analysis of the method is theoretically proven. The iterative Sherman Morrison formula does not depend on performing a matrix decomposition as is usually done in the context of other efficient EnKF implementations found in the current literature.

In Chapter 4 we present efficient implementations of the ensemble Kalman filter based on shrinkage covariance matrix estimation. The proposed method exploits the rank-deficiency of the ensemble covariance matrix in order to estimate a well-conditioned background error covariance matrix. The resulting estimator provides huge savings in terms of memory and computational time. The new filtering method can be efficiently performed in parallel with minimum latency when observational operators can be locally approximated.

In Chapter 5 an ensemble Kalman filter implementation based on a modified Cholesky decomposition for inverse covariance matrix estimation is proposed in order to exploit the natural decorrelation or errors in distant model components. The estimator is computed based on the solution of inexpensive least square problems. The methodology exploits the conditional independence of distant model components according to a given radius of influence. This results in sparse covariance matrix estimators of the inverse background error covariance matrix. Even more, the full covariance matrix is not needed under this formulation, only the Cholesky factor (a sparse lower triangular matrix) is. The convergence of the estimator is theoretically proven in the context of data assimilation.

In Chapter 6 ensemble-based methods are extended to the context of strong constraint four dimensional data assimilation. The proposed method is based on the trust region framework in order to approximate the derivatives of the strong constraint 4D-Var in the ensemble space. Under this formulation, the computation of adjoint models are not needed and even more, the analysis increments computed in the ensemble space are tested against the actual profit on the 4D-Var cost function in the model space.

In Chapter 7 the conclusions of this research and future directions are discussed.

Chapter 2

Ensemble Data Assimilation

Data assimilation is the process of estimating the state of a dynamical system at the current time given a history of prior evolution and noisy observations of the state at previous times [SC11]. Typically, A dynamical model encapsulating our knowledge of the physical laws approximates the evolution of the dynamical system [CNF11]. Two families of methods, statistical filters and variational, are widely used to solve data assimilation problems. Representative methods of those classes are *the Ensemble Kalman Filter* (EnKF) and *the Four-Dimensional Variational Method* (4D-Var), respectively. In EnKF an ensemble of model runs is propagated in time; when data is available the filtering step generates an *analysis ensemble* whose empirical mean is an estimator for the actual state of the system. Strong constraint 4D-Var seeks an *analysis initial state* such that the corresponding forecast best fits the observations within the assimilation window. It is well-accepted that both methods face specific challenges in practical applications where the number of model components is in the order of millions. For instance, ensemble-based filters suffer from statistical sampling errors, while variational methods require adjoint models which are labor-intensive to develop and computationally expensive to run.

Hybrid methods are a natural extension of ensemble-based methods into the context of 4D-Var data assimilation in order to combine the strengths of EnKF and 4D-Var methods. A decomposition of the background errors in components that are analyzed and components that are ignored has been used to estimate posterior covariances [CMMS10], and the theoretical similarities between the two approaches have been used to construct look-ahead assimilation techniques [SH13]. Other hybrid approaches are based on model reduction and/or space reduction [CNF11, DPW04, SHC$^+$13]. A discussion of model reduction techniques is given in [Oli09].

This chapter is organized as follows, in section 2.1, ensemble-based methods in the context of sequential data assimilation are discussed, their strengths and limitations are presented as well, section 2.3 discusses the extension of ensemble-based methods to the context of four dimensional variational data assimilation, current hybrid methodologies are discussed.

2.1 Sequential data assimilation

Formaly, the goal of sequential data assimilation is to estimate the state $\mathbf{x}^* \in \mathbb{R}^{n \times 1}$ of a dynamical system which (approximately) evolves according to some model operator [CJAS10a, LS12, LS14]:

$$\mathbf{x}_k = \mathcal{M}_{t_{k-1} \rightarrow t_k}\left(\mathbf{x}_{k-1}\right) \in \mathbb{R}^{n \times 1}, \text{ for } \mathbf{x} \in \mathbb{R}^{n \times 1}, \tag{2.1}$$

where, for instance, $\mathcal{M}$ can be a numerical model which mimics the dynamics of the ocean and/or atmosphere, and k denotes time index. Likewise, n is the model dimension (i.e., number of model grid components). For Gaussian errors, the assimilation is based on a prior estimation $\mathbf{x}_k^b \in \mathbb{R}^{n \times 1}$ of $\mathbf{x}^*$:

$$\mathbf{x}_k^b \sim \mathcal{N}\left(\mathbf{x}_k^*, \mathbf{B}_k\right) \in \mathbb{R}^{n \times 1}, \tag{2.2a}$$

and the noisy observation (measurement) $\mathbf{y}_k \in \mathbb{R}^{m \times 1}$:

$$\mathbf{y}_k \sim \mathcal{N}\left(\mathcal{H}\left(\mathbf{x}_k^*\right), \mathbf{R}_k\right), \in \mathbb{R}^{m \times 1}, \tag{2.2b}$$

where m is the number of observed components, $\mathcal{H} : \mathbb{R}^{n \times 1} \rightarrow \mathbb{R}^{m \times 1}$ is the observational operator, $\mathbf{R}_k \in \mathbb{R}^{m \times m}$ is the estimated data error covariance matrix and $\mathbf{B}_k \in \mathbb{R}^{n \times n}$ is the unknown background error covariance matrix. Making use of Bayesian statistics and matrix identities, the assimilation of the observation (2.2b) can be performed as follows:

$$\mathbf{x}_k^a = \mathbf{x}_k^b + \mathbf{B}_k \cdot \mathbf{H}_k^T \cdot \left[\mathbf{H}_k \cdot \mathbf{B}_k \cdot \mathbf{H}_k^T + \mathbf{R}_k\right]^{-1} \cdot \left[\mathbf{y}_k - \mathbf{H}_k \cdot \mathbf{x}_k^b\right] \in \mathbb{R}^{n \times 1}, \tag{2.3}$$

where $\mathcal{H}' \approx \mathbf{H}_k \in \mathbb{R}^{m \times n}$ is the linearized observational operator about $\mathbf{x}_k^b$, and $\mathbf{x}_k^a \in \mathbb{R}^{n \times 1}$ is well-known as the analysis state. After the assimilation, the analysis state is propagated until a new observation $\mathbf{y}_{k+1}$ is available,

$$\mathbf{x}_{k+1}^b = \mathcal{M}_{t_k \rightarrow t_{k+1}}\left(\mathbf{x}_k^a\right) .$$

wherein a new background state $\mathbf{x}_{k+1}^b$ is obtained. The assimilation process relies in the estimation of the error distributions (2.2) Typically, observations are provided with estimates of their underlying error distributions while background errors must be estimated [WH12].

2.2 Ensemble-based methods

According to equation (2.3) the variances of $\mathbf{B}_k$ in $\left[\mathbf{H}_k \cdot \mathbf{B}_k \cdot \mathbf{H}_k^T + \mathbf{R}_k\right]$ contribute to the proper scaling of the innovations $\mathbf{y}_k - \mathbf{H}_k \cdot \mathbf{x}_k^b \in \mathbb{R}^{m \times 1}$ on the observed model components (e.g., consider $\mathbf{R}_k$ and $\mathbf{B}_k$ diagonal) while, the background error correlations of $\mathbf{B}_k$ in $\mathbf{B}_k \cdot \mathbf{H}_k^T$ ensures that, the unobserved model components are properly adjusted with respect to the innovations on the observed model components. Thus, the successful assimilation of the observation (2.2b) will rely, in part, on how well the background error statistics are approximated. In the context of ensemble-based methods, an ensemble of model realizations

$$\mathbf{X}^b = \left[\mathbf{x}^{b[1]},\, \mathbf{x}^{b[2]},\, \ldots,\, \mathbf{x}^{b[N]}\right] \in \mathbb{R}^{n\times N}, \tag{2.4}$$

is used in order to estimate the unknown moments of the background error distribution [Bra11]:

$$\mathbf{x}^b \approx \overline{\mathbf{x}}^b = \frac{1}{N}\cdot\sum_{i=1}^{N}\mathbf{x}^{b[i]} \in \mathbb{R}^{n\times 1}, \tag{2.5a}$$

and

$$\mathbf{B} \approx \mathbf{P}^b = \mathbf{S}^b \cdot \mathbf{S}^{b^T} \in \mathbb{R}^{n\times n}, \tag{2.5b}$$

where N is the number of ensemble members, $\mathbf{x}^{b[i]} \in \mathbb{R}^{n\times 1}$ is the i-th ensemble member, for $1 \le i \le N$, $\overline{\mathbf{x}}^b \in \mathbb{R}^{n\times 1}$ is the background ensemble mean, $\mathbf{P}^b$ is the background ensemble covariance matrix, and $\mathbf{S}^b \in \mathbb{R}^{n\times N}$ is the matrix of member deviations scaled by $\sqrt{N-1}$

$$\mathbf{S}^b = \frac{1}{\sqrt{N-1}}\cdot\left[\mathbf{X}^b - \overline{\mathbf{x}}^b\cdot\mathbf{1}_N^T\right] \in \mathbb{R}^{n\times N}, \tag{2.5c}$$

For ease of notation we have omitted the time index superscripts. One attractive feature of ensemble covariance matrices is that the estimated background error correlations are driven by the dynamics of the numerical model (2.1). In operational data assimilation, the use of ensemble-based methods have been motivated by several factors:

1. To reduce the computational effort of the conventional Kalman filter formulation in large dimensions [HDM$^+$13].

2. To obtain error estimates on the solutions obtained via the assimilation of observations [HDM$^+$13, SB08, KMN11]. This can be easily estimated by the sample covariance matrix (2.5b)

3. To conduct sensitivity experiments at a low computational effort [MNBZ15, EYO15, Tor10].

Well-known approximations in the context of ensemble-based methods are the ensemble Kalman filter and the ensemble square root filter methods. Briefly, we discuss about them in the next sections.

2.2.1 The ensemble Kalman filter

The ensemble Kalman filter (EnKF) is a sequential Monte Carlo method for state and parameter estimation of highly non-linear models such as those found in atmospheric and oceanic sciences [SO08b, Eve03, GMC+06]. The EnKF importance relies on its basic theoretical formulation and relatively ease implementation [Eve03]. In the EnKF, given the *background* ensemble (2.4) a posterior (*analysis*) ensemble can be built as follows:

$$\mathbf{X}^a = \mathbf{X}^b + \mathbf{P}^b \cdot \mathbf{H}^T \cdot \left[\mathbf{R} + \mathbf{H} \cdot \mathbf{P}^b \cdot \mathbf{H}^T\right]^{-1} \cdot \boldsymbol{\Delta} \in \mathbb{R}^{n \times N}, \tag{2.6a}$$

where:

$$\boldsymbol{\Delta} = \mathbf{Y}^s - \mathbf{H} \cdot \mathbf{X}^b \in \mathbb{R}^{m \times N}\,, \tag{2.6b}$$

and the matrix of perturbed observations $\mathbf{Y}^s \in \mathbb{R}^{m \times N}$ reads:

$$\mathbf{Y}^s = \left[\mathbf{y} + \boldsymbol{\epsilon}^{[1]},\, \mathbf{y} + \boldsymbol{\epsilon}^{[2]},\, \dots,\, \mathbf{y} + \boldsymbol{\epsilon}^{[N]}\right] \in \mathbb{R}^{m \times N}\,, \tag{2.6c}$$

for $1 \le i \le N$, with

$$\boldsymbol{\epsilon}^{[i]} \sim \mathcal{N}\left(\mathbf{0},\, \mathbf{R}\right)\,. \tag{2.6d}$$

The analysis equation (2.6a) can be written as follows:

$$\mathbf{X}^a = \mathbf{X}^b + \mathbf{S}^b \cdot \mathbf{V}^{b^T} \cdot \mathbf{Z}_{\mathbf{X}^a},$$

where $\mathbf{V}^b = \mathbf{H} \cdot \mathbf{S}^b \in \mathbb{R}^{m \times N}$ and $\mathbf{Z}_{\mathbf{X}^a} \in \mathbb{R}^{m \times N}$ can be obtained by solving the next linear system:

$$\left[\mathbf{R} + \mathbf{V}^b \cdot \mathbf{V}^{b^T}\right] \cdot \mathbf{Z}_{\mathbf{X}^a} = \boldsymbol{\Delta} \in \mathbb{R}^{m \times N}\,. \tag{2.7}$$

The literature proposes many matrix-free methodologies in order to avoid the direct solution of (2.7) A well-known approximation is proposed by Evensen in [Eve06]. The estimated data error covariance matrix is approximated by the empirical covariance

$$\mathbf{R} = \frac{1}{N-1} \cdot \mathbf{E} \cdot \mathbf{E}^T\,,$$

where $\mathbf{E} \in \mathbb{R}^{m \times m}$ is the matrix holding samples from the data error distribution (2.6d):

$$\mathbf{E} = \left[\boldsymbol{\epsilon}^{[1]},\, \boldsymbol{\epsilon}^{[2]},\, \dots,\, \boldsymbol{\epsilon}^{[N]}\right] \in \mathbb{R}^{m \times N},$$

then, the linear system (2.7) can be re-written as follows:

$$\left[\mathbf{E} \cdot \mathbf{E}^T + \mathbf{V}^b \cdot \mathbf{V}^{b^T}\right] \cdot \mathbf{Z}_{\mathbf{X}^a} = \boldsymbol{\Delta} \in \mathbb{R}^{m \times N}\,.$$

Consider the singular value decomposition (SVD) of $\mathbf{V}^b + \sqrt[-2]{N-1} \cdot \mathbf{E} = \mathbf{U}_{\mathbf{V}^b\mathbf{E}} \cdot \mathbf{\Sigma}_{\mathbf{V}^b\mathbf{E}} \cdot \mathbf{V}_{\mathbf{V}^b\mathbf{E}}$, the solution of the linear system (2.7) can be approximated as follows:

$$\mathbf{Z}_{\mathbf{X}^a}^{\mathrm{SVD}} = \mathbf{U}_{\mathbf{V}^b\mathbf{E}} \cdot \mathbf{\Sigma}_{\mathbf{V}^b\mathbf{E}} \cdot \mathbf{\Sigma}_{\mathbf{V}^b\mathbf{E}}^T \cdot \mathbf{U}_{\mathbf{V}^b\mathbf{E}}^T \cdot \mathbf{\Delta} \,.$$

For the efficient computation of $\mathbf{Z}_{\mathbf{X}^a}^{\mathrm{SVD}}$, the matrix multiplications should be performed from right to left. The leading-order complexity of the analysis making use of the SVD decomposition reads:

$$\mathcal{O}\left(n \cdot N^2 + m \cdot N^2 + m \cdot N^2 + m \cdot N + m\right) \,. \tag{2.8}$$

Yet another manner to approximate the solution of (2.7) is proposed by Madel in [Man06]. In this implementation, the data error covariance matrix is approximated by

$$\mathbf{R} = (N-1) \cdot \mathrm{diag}\left(\mathbf{E} \cdot \mathbf{E}^T\right) .$$

Consider

$$\mathbf{M}_{\mathbf{P}^a} = (N-1) \cdot \mathbf{I} + \mathbf{V}^{b^T} \cdot \mathbf{R}^{-1} \cdot \mathbf{V}^b \,,$$

and its Cholesky factorization $\mathbf{M}_{\mathbf{P}^a} = \widehat{\mathbf{L}}_{\mathbf{P}^a} \cdot \widehat{\mathbf{L}}_{\mathbf{P}^a}^T \in \mathbb{R}^{N \times N}$, then, the linear system (2.7) can be solved as follows:

$$\mathbf{Z}_{\mathbf{X}^a}^{\mathrm{Chol}} = \mathbf{V}^{b^T} \cdot \mathbf{R}^{-1} \cdot \left[\mathbf{I} - \mathbf{V}^b \cdot \widehat{\mathbf{L}}_{\mathbf{P}^a}^{-T} \cdot \widehat{\mathbf{L}}_{\mathbf{P}^a}^{-1} \cdot \mathbf{V}^{b^T} \cdot \mathbf{R}^{-1} \cdot \mathbf{\Delta}\right]$$

where $\mathbf{Z}_{\mathbf{X}^a}^{\mathrm{Chol}}$ can be efficiently obtained when computations are performed from right to left. Note that, since $\widehat{\mathbf{L}}_{\mathbf{P}^a}^{-1} \in \mathbb{R}^{N \times N}$ is a lower triangular matrix, the direct inversion of $\widehat{\mathbf{L}}_{\mathbf{P}^a}^{-1}$ is not needed and instead, forward and backward substitutions can be used in order to compute $\mathbf{Z}_{\mathbf{X}^a}^{\mathrm{Chol}}$. The overall computational effort of the analysis step becomes:

$$\mathcal{O}\left(N^3 + n \cdot N^2 + m \cdot N^2\right) .$$

2.2.2 Ensemble square root filters

The use of perturbed observations in (2.6c) during the assimilation provides asymptotically correct analysis-error covariance estimates for large ensemble sizes and makes the formulation of the EnKF statistically consistent [WT02]. However, it also has been shown that the inclusion of perturbed observations introduces sampling errors during the computation of the analysis corrections [JFW14, And12]

Ensemble square root filters are deterministic filters in which the use of synthetic data is avoided during the assimilation process. The general idea is to apply an affine transformation to the ensemble (2.4) in such way that, the resulting ensemble is similar to (2.6a). In this context, we first compute the posterior mode of the error distribution:

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^b + \mathbf{S}^b \cdot \mathbf{V}^{b^T} \cdot \left[\mathbf{R} + \mathbf{V}^b \cdot \mathbf{V}^{b^T}\right]^{-1} \cdot \left[\mathbf{y} - \mathcal{H}\left(\overline{\mathbf{x}}^b\right)\right] , \tag{2.9a}$$

where we have let $\mathbf{V}^b = \mathbf{H} \cdot \mathbf{S}^b \in \mathbb{R}^{m \times N}$. The posterior ensemble is then built about (2.9a):

$$\mathbf{X}^a = \overline{\mathbf{x}}^a \cdot \mathbf{1}_N^T + \mathbf{S}^b \cdot \mathbf{\Gamma} \in \mathbb{R}^{n \times N}, \tag{2.9b}$$

where $\mathbf{\Gamma} \in \mathbb{R}^{N \times N}$ is known as the transformation matrix. The choice of $\mathbf{\Gamma}$ depends on the square root filter implementation. For instance , in the traditional EnSRF formulation [TAB+03], $\mathbf{\Gamma}$ is derived as follows, from the Kalman filter equations [Eve03], the relation between the analysis covariance matrix $\mathbf{P}^a$ and the background covariance matrix $\mathbf{P}^b$ is given by:

$$\begin{aligned}\mathbf{P}^a &= \left[\mathbf{I} - \mathbf{S}^b \cdot {\mathbf{V}^b}^T \cdot \left[\mathbf{R} + \mathbf{V}^b \cdot {\mathbf{V}^b}^T\right]^{-1} \cdot \mathbf{H}\right] \cdot \mathbf{P}^b, \\ &= \mathbf{S}^b \cdot \left[\mathbf{I} - {\mathbf{V}^b}^T \cdot \left[\mathbf{R} + \mathbf{V}^b \cdot {\mathbf{V}^b}^T\right]^{-1} \cdot \mathbf{V}^b\right] \cdot {\mathbf{S}^b}^T,\end{aligned}$$

therefore, the matrix $\mathbf{\Gamma}$ sought satisfies:

$$\mathbf{\Gamma} \cdot \mathbf{\Gamma}^T = \mathbf{I} - {\mathbf{V}^b}^T \cdot \left[\mathbf{R} + \mathbf{V}^b \cdot {\mathbf{V}^b}^T\right]^{-1} \cdot \mathbf{V}^b \in \mathbb{R}^{N \times N}. \tag{2.10}$$

Making use of the matrix identity

$$\mathbf{I} - {\mathbf{V}^b}^T \cdot \left[\mathbf{R} + \mathbf{V}^b \cdot {\mathbf{V}^b}^T\right]^{-1} \cdot \mathbf{V}^b = \left[\mathbf{I} + {\mathbf{V}^b}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{V}^b\right]^{-1},$$

and the singular value decomposition:

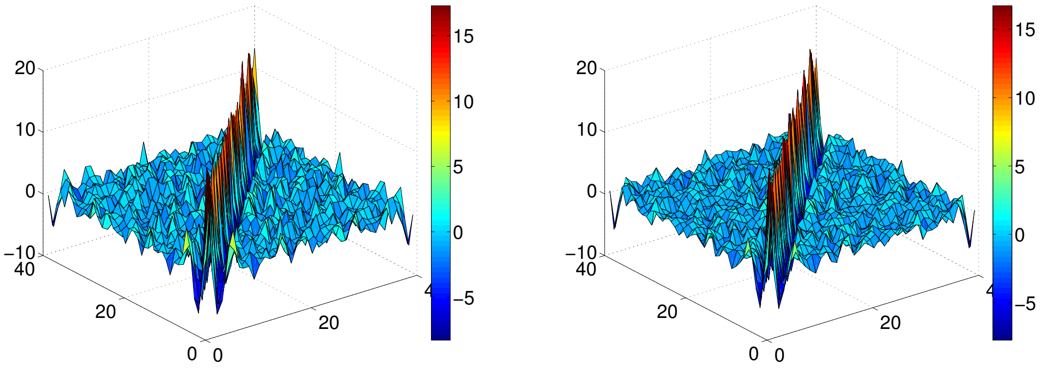
$${\mathbf{V}^b}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{V}^b = \mathbf{U}_{\mathbf{V}^b} \cdot \mathbf{\Sigma}_{\mathbf{V}^b} \cdot \mathbf{V}_{\mathbf{V}^b},$$

the ensemble tranform Kalman filter (ETKF) proposes the next choice for $\mathbf{\Gamma}$:

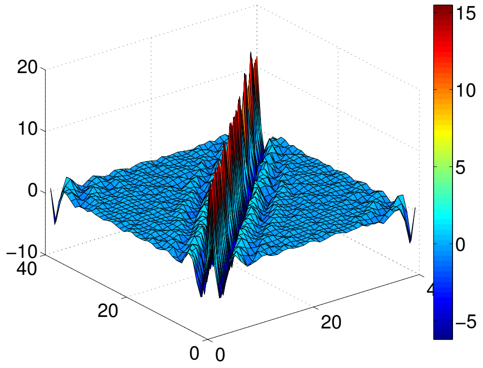
$$\mathbf{\Gamma} = \mathbf{U}_{\mathbf{V}^b} \cdot \left[\mathbf{I} + \mathbf{\Sigma}_{\mathbf{V}^b}\right]^{-1/2} \cdot {\mathbf{U}_{\mathbf{V}^b}}^T. \tag{2.11}$$

2.2.3 Avoiding filter divergence

In operational data assimilation, the number of model components n is much larger than the number of model realizations N (ensemble size) Typically, N ranges in the order of hundreds while n does it in the order of millions. In general, the analysis corrections for the n-th dimensional state are performed in a space with only $N - 1$ degrees of freedom. An immediate consequence is the presence of spurious correlations in the ensemble covariance matrix, for instance, figure 2.1 shows the effect of sampling errors in the ensemble covariance matrix for the Lorenz 96 model [Lor05] making use of different ensemble sizes. Other approximations of background error covariance matrix are found in the literature, a traditional approximation of $\mathbf{B}$ is the Hollingworth and Lonnberg method [HL86] in which the difference between observations and background states are treated as a combination of background and observations errors. However, this method provides statistics of background errors in observation space, and requires dense observing networks. Another method has been proposed by Benedetti and Fisher [BF07] based on forecast differences in which the spatial correlations of background errors are assumed to be similar at 24 and 48 hours forecasts. This method can be efficiently implemented in practice, however, it does not perform well in data-sparse regions, and the statistics provided are a mixture of analysis and background errors. A general manner to reduce the impact of spurious correlations is found in the context of localization methods. Two well-known approximations are proposed in the literature: covariance matrix localization and domain localization.



(a) $\mathbf{P}^b$, $N = 110$ (b) $\mathbf{P}^b$, $N = 210$



(c) $\mathbf{P}^b$, $N = 10^5$

Figure 2.1: Background error estimation via the ensemble covariance matrix $\mathbf{P}^b$ for different ensemble sizes N. The number of model components in the Lorenz 96 model [Lor05] is $n = 40$.

Covariance matrix localization artificially reduces correlations between distant model components via a Schur product with a localization matrix $\mathbf{\Pi} \in \mathbb{R}^{n \times n}$:

$$\widehat{\mathbf{P}}^{\mathrm{b}} = \mathbf{\Pi} \circ \mathbf{P}^{\mathrm{b}} \in \mathbb{R}^{n \times n}$$

and then $\mathbf{P}^{\mathrm{b}}$ is replaced by $\widehat{\mathbf{P}}^{\mathrm{b}} \in \mathbb{R}^{n\times n}$ in the EnKF analysis equation (2.6a). The entries of $\mathbf{\Pi}$ decrease with the distance between model components depending on the radius of influence ζ:

$$\left\{\mathbf{\Pi}\right\}_{i,j} = \exp\left(-\frac{\pi\left(m_i,\, m_j\right)}{f(\zeta)}\right)\,,\ \text{for } 1 \le i \le j \le n\,, \tag{2.12}$$

where $\pi\left(m_i,\, m_j\right)$ represents the physical distance between the model components m_i and m_j while, $f(\zeta)$ is a function of ζ (e.g., $f(\zeta) = 2\cdot\zeta^2$). The exponential decay allows to reduce the impact of innovations between distant model components. The use of covariance matrix localization alleviates the impact of sampling errors. However, the explicit computation of $\mathbf{\Pi}$ (and even $\mathbf{P}^{\mathrm{b}}$) is prohibitive owing to numerical model dimensions. Thus, domain localization methods [Bue11, Kep00] are commonly used in the context of operational data assimilation. One of the best EnKF implementations based on domain localization is the local ensemble transform Kalman filter (LETKF) [OHS$^+$04b]. In the LETKF the analysis increments are computed in the space spanned by the ensemble perturbations $\mathbf{U}^{\mathrm{b}} \in \mathbb{R}^{n\times N}$, where

$$\mathbf{U}^b = \mathbf{X}^b - \overline{\mathbf{x}}^b \cdot \mathbf{1}_N^T \in \mathbb{R}^{n\times N}\,. \tag{2.13a}$$

An approximation of the analysis covariance matrix in this space reads:

$$\widehat{\mathbf{P}}^{\mathrm{a}} = \left[(N-1)\cdot\mathbf{I} + \mathbf{Q}^T\cdot\mathbf{R}^{-1}\cdot\mathbf{Q}\right]^{-1} \in \mathbb{R}^{N\times N}, \tag{2.13b}$$

where $\mathbf{Q} = \mathbf{H}\cdot\mathbf{U}^{\mathrm{b}} \in \mathbb{R}^{m\times N}$ and $\mathbf{I}$ is the identity matrix consistent with the dimension. The analysis increments in the subspace are:

$$\mathbf{w}^{\mathrm{a}} = \widehat{\mathbf{P}}^{\mathrm{a}}\cdot\mathbf{Q}^T\cdot\mathbf{R}^{-1}\cdot\left[\mathbf{y} - \mathcal{H}(\overline{\mathbf{x}}^{\mathrm{b}})\right] \in \mathbb{R}^{N\times 1}, \tag{2.13c}$$

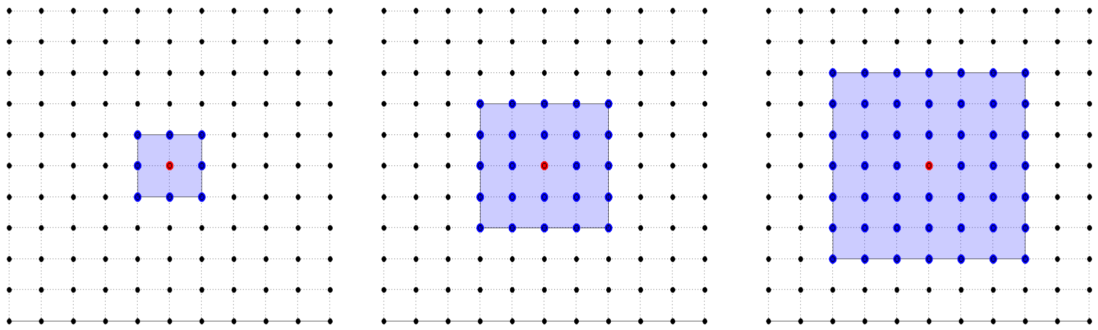
from which an estimation of the analysis mean in the model space can be obtained:

$$\overline{\mathbf{x}}^{\mathrm{a}} = \overline{\mathbf{x}}^{\mathrm{b}} + \mathbf{U}^{\mathrm{b}}\cdot\mathbf{w}^{\mathrm{a}} \in \mathbb{R}^{n\times 1}. \tag{2.13d}$$

Finally, the analysis ensemble reads:

$$\mathbf{x}^{\mathrm{a}} = \overline{\mathbf{x}}^{\mathrm{a}}\cdot\mathbf{1}_N^T + \mathbf{U}^{\mathrm{b}}\cdot\left[(N-1)\cdot\widehat{\mathbf{P}}^{\mathrm{a}}\right]^{1/2} \in \mathbb{R}^{n\times N}. \tag{2.13e}$$

The domain localization in the LETKF is performed as follows: each model component is surrounded by a local box of radius ζ. Within each local domain the analysis equations (2.13) are applied, and therefore a local analysis component is obtained. All local analysis components are mapped back onto the model space to obtain the global analysis state. Local boxes for different radii are shown in Figure 2.2. Note that, the LETKF implementations does not require any background error covariance model, and even more, since computations are performed onto a small space, the analysis increments can be efficiently computed. However, with non-local satellite observations, the effective box sizes become large and therefore, affordable ensemble sizes implies poor analysis resolution and hence accuracy. Recall that, the local sample covariance matrix (2.5b) is utilized as the covariance estimator of the local $\mathbf{B}$. This can perform well when small radii ζ are considered during the assimilation step. Nevertheless, for large values of ζ, the analysis corrections can be impacted by spurious correlations since the local sample covariance matrix can be rank deficient. Consequently, the local analysis increments can perform poorly. The same situation can happen when block-correlations in the estimated data error covariance matrix are of large sizes.



(a) $\zeta = 1$ (b) $\zeta = 2$ (c) $\zeta = 3$

Figure 2.2: Local domains for different radii of influence ζ. The red dot is the model component to be assimilated, blue components are within the scope of ζ, and black model components are unused during the local assimilation process.

2.3 Four dimensional ensemble data assimilation

Ensemble-based methods can provide flow-dependent error estimates of the background errors (with the Monte Carlo methods), but it does not have the ability to assimilate the observation data available at distributed times.

4D-Var considers cost functions of the form

$$\mathcal{J}(\mathbf{x}_0) \quad = \quad \underbrace{\frac{1}{2}\left\|\mathbf{x}_0 - \mathbf{x}_0^b\right\|^2_{\mathbf{B}_0^{-1}}}_{\mathcal{J}^b(\mathbf{x})} + \underbrace{\frac{1}{2}\sum_{k=0}^{M}\left\|\mathbf{y}_k - \mathcal{H}\left(\mathbf{x}_k\right)\right\|^2_{\mathbf{R}_k^{-1}}}_{\mathcal{J}^o(\mathbf{x})}, \tag{2.14}$$

where $\mathcal{J}^b(\mathbf{x})$ and $\mathcal{J}^o(\mathbf{x})$ are known as the background and observation cost functions, respectively. Likewise M is the number of observations within the assimilation window, and at time t_k, for $0 \leq k \leq M$, $\mathbf{y}_k$ denotes the observation, and $\mathbf{R}_k$ is the estimated data error covariance matrix. The cost function (2.14) is the negative logarithms of the a posteriori probability density when all the data and background errors are normally distributed. The maximum likelihood estimate of the initial state is then obtained by minimizing the cost function, i.e., the analysis step is computed by solving the optimization problem

$$\mathbf{x}_0^a = \arg\min_{\mathbf{x}_0} \mathcal{J}(\mathbf{x}_0) \qquad \text{subject to} \qquad \mathbf{x}_k = \mathcal{M}_{t_{k-1}\to t_k}\left(\mathbf{x}_{k-1}\right), \tag{2.15}$$

for $1 \le k \le M$. The formulation of (2.14) allows 4D-Var to assimilate data which appears at different observation times.

The computation of the gradient (2.14) with respect to the control variable $\mathbf{x}_0 \in \mathbb{R}^{n\times 1}$ requires one forward and one adjoint model integration. The construction of an adjoint model for real, large forecast models is an extremely labor-intensive process. In order to avoid the implementation of adjoint models four dimensional ensemble Kalman filter methods (4D-EnKF) [ZZ11] have been recently proposed. They naturally propagate flow dependent background covariance matrices via ensembles [HS00, Lor03, SHC$^+$13, CCFT13]. Numerical experiments show robust performance with a small number of ensemble members [YMW$^+$13, THL$^+$13]. Moreover, the solution (2.15) can be treated as the new background state in (2.14), which provides a better solution [CCF$^+$13].

4D-EnKF based methods are defined as follows. The initial ensemble

$$\mathbf{X}_0^b = \left[\mathbf{x}_0^{b[1]}, \mathbf{x}_0^{b[2]}, \ldots, \mathbf{x}_0^{b[N]}\right] \in \mathbb{R}^{n\times N}, \tag{2.16}$$

is propagated in time and $M+1$ snapshots of each background ensemble member state at time moments $t_0, t_1, \ldots, t_M$ along the trajectory are stored

$$\mathbf{X}^s = \begin{bmatrix} \mathbf{x}_0^{b[1]} & \mathbf{x}_0^{b[2]} & \ldots & \mathbf{x}_0^{b[N]} \\ \mathbf{x}_1^{b[1]} & \mathbf{x}_1^{b[2]} & \ldots & \mathbf{x}_1^{b[N]} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_M^{b[1]} & \mathbf{x}_M^{b[2]} & \ldots & \mathbf{x}_M^{b[N]} \end{bmatrix} \in \mathbb{R}^{(n\cdot(M+1))\times N}. \tag{2.17}$$

Each entry of the background ensemble matrix $\mathbf{X}^s$ is an n-dimensional vector $\mathbf{x}_k^{b[i]}$ which represents the state of ensemble member i at time t_k. The i-th column of $\mathbf{X}^s$ contains all the snapshots of the i-th ensemble member, and the k-th row of blocks corresponds to all ensemble member states at t_k.

Consider now a trajectory of the model. The state $\mathbf{x}_k$ at t_k is approximated by a linear combination of the anomalies (deviations from the mean)

$$\mathbf{x}_k = \overline{\mathbf{x}}_k^b + \sum_{i=1}^{N} \alpha_i \cdot \underbrace{\left(\mathbf{x}_k^{b[i]} - \overline{\mathbf{x}}_k\right)}_{\boldsymbol{\psi}_k^{(i)}} = \overline{\mathbf{x}}_k^b + \boldsymbol{\Psi}_k \cdot \boldsymbol{\alpha}, \tag{2.18}$$

where

$$\begin{aligned} \overline{\mathbf{x}}_k^b &= \frac{1}{N} \cdot \sum_{i=1}^{N} \mathbf{x}_k^{b[i]} \in \mathbb{R}^{n\times 1}\,, \\ \mathbf{\Psi}_k &= \left[\boldsymbol{\psi}_k^{[1]}, \boldsymbol{\psi}_k^{[2]}, \dots, \boldsymbol{\psi}_k^{[N]}\right] \in \mathbb{R}^{n\times N}\,, \end{aligned} \tag{2.19}$$

and the time-independent weight vector

$$\boldsymbol{\alpha} = \left[\alpha_1, \alpha_2, \dots, \alpha_N\right]^T \in \mathbb{R}^{N\times 1}\,,$$

contains the coordinates of $\mathbf{x}_k$ in the ensemble space.

By replacing (2.18) in (2.14) and linearizing the observation operator $\mathcal{H}_k' \approx \mathbf{H}_k$ about $\overline{\mathbf{x}}_k^b$, the 4D-Var cost function (2.14) can be written in the ensemble space as follows:

$$\mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) = \frac{1}{2} \left\|\mathbf{d}^b - \mathbf{\Psi}_0 \cdot \boldsymbol{\alpha}\right\|^2_{\mathbf{B}_0^{-1}} + \frac{1}{2} \sum_{k=0}^{M} \left\|\mathbf{d}_k^o - \mathbf{Q}_k \cdot \boldsymbol{\alpha}\right\|^2_{\mathbf{R}_k^{-1}} \tag{2.20}$$

where $\mathbf{d}^b = \mathbf{x}_0^b - \overline{\mathbf{x}}_0^b \in \mathbb{R}^{n\times 1}$ and $\mathbf{d}_k^o = \mathbf{y}_k - \mathbf{H}_k \cdot \overline{\mathbf{x}}_k^b \in \mathbb{R}^{m\times 1}$ are the innovation vectors on the background and observations, respectively, and $\mathbf{Q}_k = \mathbf{H}_k \cdot \mathbf{\Psi}_k \in \mathbb{R}^{m\times N}$.

The optimal solution in the ensemble space

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) \in \mathbb{R}^{N\times 1}\,, \tag{2.21}$$

provides an approximation of the analysis trajectory started from (2.15) through the relation

$$\overline{\mathbf{x}}_k^a = \overline{\mathbf{x}}_k^b + \mathbf{\Psi}_k \cdot \boldsymbol{\alpha}^* \in \mathbb{R}^{n\times 1}\,. \tag{2.22}$$

The derivatives of (2.20) are

$$\begin{aligned} \nabla_{\boldsymbol{\alpha}} \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) &= \left[\mathbf{\Psi}_0^T \cdot \mathbf{B}_0^{-1} \cdot \mathbf{\Psi}_0 + \sum_{k=0}^{M} \mathbf{Q}_k^T \cdot \mathbf{R}_k \cdot \mathbf{Q}_k\right] \cdot \boldsymbol{\alpha} \qquad \text{(2.23a)} \\ &- \left[\mathbf{\Psi}_0^T \cdot \mathbf{B}_0^{-1} \cdot \mathbf{d}^b + \sum_{k=0}^{M} \mathbf{Q}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{d}_k\right] \in \mathbb{R}^{N\times 1}\,, \\ \nabla^2_{\boldsymbol{\alpha},\boldsymbol{\alpha}} \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) &= \mathbf{\Psi}_0^T \cdot \mathbf{B}_0^{-1} \cdot \mathbf{\Psi}_0 + \sum_{k=0}^{M} \mathbf{Q}_k^T \cdot \mathbf{R}_k \cdot \mathbf{Q}_k \in \mathbb{R}^{N\times N}\,, \qquad \text{(2.23b)} \end{aligned}$$

and the solution of the quadratic minimization problem (2.21) is

$$\boldsymbol{\alpha}^* = \nabla^2_{\boldsymbol{\alpha},\boldsymbol{\alpha}} \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha})^{-1} \cdot \left[{\mathbf{\Psi}_0}^T \cdot \mathbf{B}_0^{-1} \cdot \mathbf{d}^{\mathrm{b}} + \sum_{k=0}^{M} \mathbf{Q}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{d}_k\right]\,.$$

Since $\overline{\mathbf{x}}_k^a$ in (2.22) represents an approximated solution rather than an exact solution, the initial analysis $\overline{\mathbf{x}}_0^a$ is only recovered and propagated in time in order to obtain an approximation of the optimal trajectory of (2.14).

Equivalent bases for the range of $\mathbf{\Psi}_k$ can be utilized to formulate the subspace approximation (2.18). For instance, the proper orthogonal decomposition (POD) [TXD08] is widely used to obtain a basis that captures most of the variance of the snapshot (2.17). Consider the matrix of snapshots deviations

$$\delta \mathbf{X}^s = \frac{1}{\sqrt{N}} \left[\mathbf{\Psi}_0{}^T, \mathbf{\Psi}_1{}^T, \dots, \mathbf{\Psi}_M{}^T \right]^T \in \mathbb{R}^{(n \cdot (M+1)) \times N} \,,$$

and its singular value decomposition (SVD)

$$\delta \mathbf{X}^s = \mathbf{U}_{\delta \mathbf{X}^s} \cdot \mathbf{\Sigma}_{\delta \mathbf{X}^s} \cdot \mathbf{V}_{\delta \mathbf{X}^s} \in \mathbb{R}^{(n \cdot (M+1)) \times N} \,,$$

where $\mathbf{U}_{\delta \mathbf{X}^s} \in \mathbb{R}^{(n \cdot (M+1)) \times (n \cdot (M+1))}$ and $\mathbf{V}_{\delta \mathbf{X}^s} \in \mathbb{R}^{N \times N}$ are the right and left singular vectors, respectively, and $\mathbf{\Sigma}_{\delta \mathbf{X}^s} = \operatorname{diag}(\sigma_1, \sigma_2, \dots, \sigma_N) \in \mathbb{R}^{(n \cdot (M+1)) \times N}$ is a diagonal matrix whose diagonal entries are the singular values with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_N$. Since

$$\delta \mathbf{X}^{sT} \cdot \delta \mathbf{X}^s = \mathbf{V}_{\delta \mathbf{X}^s} \cdot \mathbf{\Sigma}^2_{\delta \mathbf{X}^s} \cdot \mathbf{V}^T_{\delta \mathbf{X}^s} \in \mathbb{R}^{N \times N} \,,$$

the POD basis vectors can be computed as

$$\mathbf{\Phi}_k = \mathbf{\Psi}_k \cdot \mathbf{V}_{\delta \mathbf{X}^s} \cdot \mathbf{\Sigma}^{-1/2}_{\delta \mathbf{X}^s} \in \mathbb{R}^{n \times N} \,, \tag{2.24}$$

and therefore, equivalent to (2.21), $\mathbf{x}_k$ can be expressed as follows:

$$\mathbf{x}_k = \overline{\mathbf{x}}_k^b + \sum_{i=1}^{r} \beta_i \cdot \left(\frac{\mathbf{\Psi}_k \cdot \mathbf{v}_i}{\sqrt{\sigma_i}} \right) = \overline{\mathbf{x}}_k + \mathbf{\Phi}_k^r \cdot \boldsymbol{\beta} \,,$$

where we have chosen the columns of $\mathbf{\Sigma}$ to be orthonormal, $\mathbf{\Phi}_k^r$ holds the first r basis vectors, $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_r]^T \in \mathbb{R}^{r \times 1}$ is the vector of weights to be determined, and r can be computed as follows

$$r = \arg\min_p \left\{ p, I(p) : \frac{\sum_{i=1}^p \sigma_i}{\sum_{i=1}^N \sigma_i} > \gamma : \gamma \in (0,\, 1) \right\} . \tag{2.25}$$

Note that, the parameter γ provides how much variance (sometimes called kinetic energy) we want to retain in the POD bases, commonly the values of γ ranges in $(0.9\,, 0.95)$. It is well known that POD bases are the most efficient among all possible linear combinations in the sense, for a given number r of basis vectors, POD decomposition captures the most possible variance [HH01, HH02]. In addition, POD bases reduce the equation (2.20) to

$$\mathcal{J}_{\text{ens}}^{\text{POD}}(\boldsymbol{\beta}) = \frac{1}{2} \cdot N \cdot \|\boldsymbol{\beta}\|^2 + \frac{1}{2} \cdot \sum_{k=0}^{M} \|\mathbf{d}_k^o - \mathbf{Z}_k \cdot \boldsymbol{\beta}\|^2_{\mathbf{R}_k^{-1}} \,, \tag{2.26}$$

whose first and second derivatives are

$$\begin{aligned}
\nabla_{\boldsymbol{\beta}} \mathcal{J}^{\text{POD}}_{\text{ens}}\left(\boldsymbol{\beta}\right) &= \left[N \cdot \mathbf{I} + \sum_{k=0}^{M} \mathbf{Z}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{Z}_k \right] \cdot \boldsymbol{\beta} \\
&- \sum_{k=0}^{M} \mathbf{Z}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{d}_k^o \in \mathbb{R}^{r \times 1} \,, \\
\nabla^2_{\boldsymbol{\beta},\boldsymbol{\beta}} \mathcal{J}^{\text{POD}}_{\text{ens}}\left(\boldsymbol{\beta}\right) &= N \cdot \mathbf{I} + \sum_{k=0}^{M} \mathbf{Z}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{Z}_k \in \mathbb{R}^{r \times r} \,,
\end{aligned}$$

where $\mathbf{Z}_k = \mathbf{H}_k \cdot \boldsymbol{\Phi}_k$ and $\mathbf{I}$ is the identity matrix consistent with the dimension. Thus, an equivalent problem to (2.21) is

$$\boldsymbol{\beta}^* = \arg \min_{\boldsymbol{\beta}} \mathcal{J}^{\text{POD}}_{\text{ens}}(\boldsymbol{\beta}) \in \mathbb{R}^{r \times 1} \,, \tag{2.28}$$

whose solution reads:

$$\boldsymbol{\beta}^* = \nabla^2_{\boldsymbol{\beta},\boldsymbol{\beta}} \mathcal{J}^{\text{POD}}_{\text{ens}}\left(\boldsymbol{\beta}\right)^{-1} \cdot \left[\sum_{k=0}^{M} {\mathbf{Z}_k}^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{d}_k^o \right] \,. \tag{2.29}$$

Data assimilation methods that make use of the POD basis (such as, for example, POD-4D-EnKF [TXD08]) are defined as follows:

1. **Ensemble generation**. The initial ensemble (2.16) is built centered at $\mathbf{x}_0^{\text{b}}$ with covariance matrix $\mathbf{B}_0$. The ensemble members are propagated and $M+1$ snapshots of each member are saved.
2. **Basis computation**. The POD basis (2.24) are computed and r vectors are selected according to (2.25).
3. **Compute reduced-space solution**. Compute the optimal weights solution(2.29).
4. **Compute full-space initial condition**. Let $\overline{\mathbf{x}}_0^{\text{a}} = \overline{\mathbf{x}}_0^{b} + \boldsymbol{\Phi}_0^r \cdot \boldsymbol{\beta}^*$.
5. **Propagate analysis**. $\overline{\mathbf{x}}_k^{b} = \mathcal{M}_{t_{k-1} \rightarrow t_k}(\overline{\mathbf{x}}_{k-1}^{a})$ for $1 \leq k \leq M$.

According to Tian [TXD08] the POD bases capture not only the spatial structure of the state but also its temporal evolution.

The optimal solution of the POD-4D-EnKF provides an approximation of the analysis (2.15). The process can be continued in an iterative fashion in order to improve the analysis; the solution of one iteration becomes the new background state for the next iteration. The idea of using a sequence of minimizations of the surrogates (2.20) or (2.26) in order to approach the minimum of (2.14) has been explored in the derivative-free optimization literature [CSV09].

A rigorous implementation has been recently proposed by Gratton et al. [GLS14]. The method is called *Iterative Subspace Minimization* (ISM) and solves iteratively the problem (2.14) via the projection of the full space onto the space spanned by the POD bases. The ISM method is defined as follows:

1. **Initialization**. Let $\mathbf{x}_0^{(0)} \leftarrow \overline{\mathbf{x}}_0$ (the initial background) and $j \leftarrow 0$.
2. **Ensemble generation**. The initial ensemble (2.16) is built centered at $\mathbf{x}_0^{(j)}$ with covariance matrix $\mathbf{B}_0$. The ensemble members are propagated and $M+1$ snapshots are saved.
3. **Basis computation**. The POD basis (2.24) are computed and r vectors are selected according to (2.25).
4. **Suproblem solution**. The optimization problem (2.28) is partially solved making use of the *Coordinate Search Method (CSM)* [CV07, CRV10], from which we obtain $\boldsymbol{\beta}^*$.
5. **Solution update**. Set $\mathbf{x}_0^{(j+1)} \leftarrow \mathbf{x}_0^{(j)} + \Phi_0^r \cdot \boldsymbol{\beta}^*$, $j \leftarrow j+1$, and go to Step 2.

The ISM method solves the optimization subproblem (2.26) via the CSM approach which does not make use of derivative information, and therefore no optimality conditions are checked. Other methods can be used at this step. For instance, one can employ the analytical solution (2.28), which guarantees to obtain the local minimizer of each subproblem and reduce the total number of outer iterations and function evaluations.

Chapter 3

An Ensemble Kalman Filter Based on an Iterative Sherman Morrison Formula

In this chapter, we present a practical implementation of the ensemble Kalman (EnKF) filter based on an iterative Sherman-Morrison formula. The new direct method exploits the special structure of the ensemble-estimated error covariance matrices in order to efficiently solve the linear systems involved in the analysis step of the EnKF. The computational complexity of the proposed implementation is equivalent to that of the best EnKF implementations available in the literature when the number of observations is much larger than the number of ensemble members, as typically is case in practice. Moreover, the proposed method provides the best theoretical complexity when is compared to generic formulations of matrix inversion based on the Sherman Morrison formula. The stability analysis of the proposed method is carried out and a pivoting strategy is discussed in order to reduce the accumulation of round-off errors without increasing the computational effort. A parallel implementation is discussed as well. Computational experiments carried out using an oceanic quasi-geostrophic model reveal that the proposed algorithm yields the same accuracy as other EnKF implementations, but scales better with regard to the number of observations.

The chapter is structured as follows. Section 3.1 presents the novel implementation of the EnKF based on iterative Sherman-Morrison formula, in which the special structure of the measurements error covariance matrix is exploited. Computational cost and stability analyses are carried out for this approach, and pivoting and parallelization ideas are discussed. Section 3.2 reports numerical results of the proposed algorithm applied to the quasi-geostrophic model. Conclusions are drawn in Section 3.3.

3.1 Iterative Implementation of the EnKF Analysis Step

We make the assumptions [TAB$^+$03, Man06] that, in practice:

1. The data error covariance matrix $\mathbf{R}$ has a simple structure (e.g., is block diagonal).
2. The observation operator $\mathbf{H}$ is sparse or can be applied efficiently.
3. The variables m and n are very large.
4. In many real applications of the EnKF $m \gg N$, and the number of variables ranges between $\mathcal{O}\left(10^7\right)$ and $\mathcal{O}\left(10^9\right)$.

Taking into in account the previous assumptions, we now derive the implementation of the EnKF. Recall the matrix of member deviations $\mathbf{S}^b \in \mathbb{R}^{n \times N}$:

$$\mathbf{S}^b = \frac{1}{\sqrt{N-1}} \cdot \left[\mathbf{X}^b - \overline{\mathbf{x}}^b \cdot \mathbf{1}_N^T\right] \in \mathbb{R}^{n \times N} \,. \tag{3.1}$$

The linear system to solve during the assimilation step of the EnKF reads:

$$\left[\mathbf{R} + \mathbf{V}^b \cdot \mathbf{V}^{b^T}\right] \cdot \mathbf{Z}_{\mathbf{X}^a} = \mathbf{\Delta} \in \mathbb{R}^{m \times N} \tag{3.2}$$

where $\mathbf{V}^b = \mathbf{H} \cdot \mathbf{S}^b \in \mathbb{R}^{m \times N}$ and the innovation vector is given by $\mathbf{\Delta} = \mathbf{y} \cdot \mathbf{1}_N^T + \mathbf{E} - \mathbf{H} \cdot \mathbf{X}^b$, $\mathbf{E} = \left[\boldsymbol{\epsilon}^{[1]},\, \boldsymbol{\epsilon}^{[2]},\, \ldots,\, \boldsymbol{\epsilon}^{[N]}\right] \in \mathbb{R}^{m \times N}$ with

$$\boldsymbol{\epsilon}^{[i]} \sim \mathcal{N}\left(\mathbf{0},\, \mathbf{R}\right), \quad 1 \leq i \leq N \,.$$

Denote by $\mathbf{W} \in \mathbb{R}^{m \times m}$ the matrix of scaling weights:

$$\mathbf{W} = \mathbf{R} + \mathbf{V}^b \cdot \mathbf{V}^{b^T} = \mathbf{R} + \sum_{i=1}^{N} \mathbf{v}^{b[i]} \cdot \mathbf{v}^{b[i]^T}$$

where $\mathbf{v}^{b[i]} \in \mathbb{R}^{m \times 1}$ denotes the i-th column of matrix $\mathbf{V}^b$. Note that, $\mathbf{W}$ can be computed recursively via the sequence of matrices $\mathbf{W}^{(k)} \in \mathbb{R}^{m \times m}$ with $\mathbf{W}^{(0)} = \mathbf{R}$,

$$\mathbf{W}^{(k)} = \mathbf{W}^{(k-1)} + \mathbf{v}^{b[k]} \cdot \mathbf{v}^{b[k]^{\mathbf{T}}}, \;\; 1 \leq k \leq N. \tag{3.3}$$

By replacing equation (3.3) in (3.2) we obtain:

$$\left[\mathbf{W}^{(N-1)} + \mathbf{v}^{b[N]} \cdot \mathbf{v}^{b[N]^{\mathbf{T}}}\right] \cdot \mathbf{Z}_{\mathbf{X}^a} = \mathbf{\Delta} \,, \text{ for } 1 \leq j \leq N \,. \tag{3.4}$$

The linear system (3.4) can be solved making use of the Sherman Morrison formula [Fra08]

$$\left[\mathbf{A} + \mathbf{u} \cdot \mathbf{v}^T\right]^{-1} = \mathbf{A}^{-1} - \frac{1}{1 + \mathbf{v}^T \cdot \mathbf{A}^{-1} \cdot \mathbf{u}} \cdot \mathbf{A}^{-1} \cdot \mathbf{u} \cdot \mathbf{v}^T \cdot \mathbf{A}^{-1}$$

with $\mathbf{A} = \mathbf{W}^{(N)}$, $\mathbf{v} = \mathbf{u} = \mathbf{v}^{b[N]}$, the solution of (3.4) can be obtained for each right hand-side $\boldsymbol{\Delta}^{[j]}$, for $1 \leq j \leq N$, as follows:

$$\mathbf{W}_{\mathbf{Z}}^{(N)[j]} = \mathbf{W}_{\mathbf{Z}}^{(N-1)[j]} - \frac{1}{\gamma^{(N)}} \cdot \mathbf{U}_{\mathbf{Z}}^{(N-1)[N]} \cdot \left[{\mathbf{v}^{b[N]}}^T \cdot \mathbf{W}_{\mathbf{Z}}^{(N-1)[j]} \right] \tag{3.5}$$

where $\mathbf{W}_{\mathbf{Z}}^{(N)[j]} \in \mathbb{R}^{m \times 1}$ denotes the solution of the linear system for the j-th right-hand side (the j-th column of matrix $\mathbf{W}_{\mathbf{Z}}^{(N)}$). More general, we can refer to $\mathbf{W}_{\mathbf{Z}}^{(N)}$ as the solution of the linear system (3.4), for instance, $\mathbf{W}_{\mathbf{Z}}^{(N)} = \mathbf{Z}_{\boldsymbol{\delta}\mathbf{X}^a}$. Likewise, $\gamma^{(N)} = 1 + {\mathbf{v}^{b[N]}}^T \cdot \mathbf{U}^{(N-1)[N]}$ and

$$\begin{aligned} \left[\mathbf{W}^{(N-2)} + \mathbf{v}^{b[N-1]} \cdot {\mathbf{v}^{b[N-1]}}^T \right] \cdot \mathbf{W}_{\mathbf{Z}}^{(N-1)[j]} &= \boldsymbol{\Delta}^{[j]} && \text{(3.6a)} \\ \left[\mathbf{W}^{(N-2)} + \mathbf{v}^{b[N-1]} \cdot {\mathbf{v}^{b[N-1]}}^T \right] \cdot \mathbf{U}_{\mathbf{Z}}^{(N-1)[N]} &= \mathbf{v}^{b[N]} && \text{(3.6b)} \end{aligned}$$

the same idea applied to (3.4) can be used in order to solve the linear systems (3.6), for (3.6a) we have

$$\mathbf{W}_{\mathbf{Z}}^{(N-1)[j]} = \mathbf{W}_{\mathbf{Z}}^{(N-2)[j]} - \frac{1}{\gamma^{(N-1)}} \cdot \mathbf{U}_{\mathbf{Z}}^{(N-2)[N-1]} \cdot \left[{\mathbf{v}^{b[N-1]}}^T \cdot \mathbf{W}_{\mathbf{Z}}^{(N-2)[j]} \right]$$

while for (3.6b) we have

$$\mathbf{U}_{\mathbf{Z}}^{(N-1)[N]} = \mathbf{U}_{\mathbf{Z}}^{(N-2)[N]} - \frac{1}{\gamma^{(N-1)}} \cdot \mathbf{U}_{\mathbf{Z}}^{(N-2)[N-1]} \cdot \left[{\mathbf{v}^{b[N-1]}}^T \cdot \mathbf{U}_{\mathbf{Z}}^{(N-2)[N]} \right]$$

where $\gamma^{(N-1)} = 1 + {\mathbf{v}^{b[N-1]}}^T \cdot \mathbf{U}^{(N-2)[N-1]}$

$$\begin{aligned} \left[\mathbf{W}^{(N-3)} + \mathbf{v}^{b[N-2]} \cdot {\mathbf{v}^{b[N-2]}}^T \right] \cdot \mathbf{W}_{\mathbf{Z}}^{(N-2)[j]} &= \boldsymbol{\Delta}^{[j]} && \text{(3.7a)} \\ \left[\mathbf{W}^{(N-3)} + \mathbf{v}^{b[N-2]} \cdot {\mathbf{v}^{b[N-2]}}^T \right] \cdot \mathbf{U}_{\mathbf{Z}}^{(N-2)[N]} &= \mathbf{v}^{b[N]} && \text{(3.7b)} \\ \left[\mathbf{W}^{(N-3)} + \mathbf{v}^{b[N-2]} \cdot {\mathbf{v}^{b[N-2]}}^T \right] \cdot \mathbf{U}_{\mathbf{Z}}^{(N-2)[N-1]} &= \mathbf{v}^{b[N-1]} \,. && \text{(3.7c)} \end{aligned}$$

Again, the linear systems (3.7) can be solved as follows:

$$\begin{aligned} \mathbf{W}_{\mathbf{Z}}^{(N-2)[j]} &= \mathbf{W}_{\mathbf{Z}}^{(N-3)[j]} - \frac{1}{\gamma^{(N-2)}} \cdot \mathbf{U}_{\mathbf{Z}}^{(N-3)[N-2]} \cdot \left[{\mathbf{v}^{b[N-2]}}^T \cdot \mathbf{W}_{\mathbf{Z}}^{(N-3)[j]} \right], \\ \mathbf{U}_{\mathbf{Z}}^{(N-2)[N]} &= \mathbf{U}_{\mathbf{Z}}^{(N-3)[N]} - \frac{1}{\gamma^{(N-2)}} \cdot \mathbf{U}_{\mathbf{Z}}^{(N-3)[N-2]} \cdot \left[{\mathbf{v}^{b[N-2]}}^T \cdot \mathbf{U}_{\mathbf{Z}}^{(N-3)[N]} \right], \\ \mathbf{U}_{\mathbf{Z}}^{(N-2)[N-1]} &= \mathbf{U}_{\mathbf{Z}}^{(N-3)[N-1]} - \frac{1}{\gamma^{(N-2)}} \cdot \mathbf{U}_{\mathbf{Z}}^{(N-3)[N-2]} \cdot \left[{\mathbf{v}^{b[N-2]}}^T \cdot \mathbf{U}_{\mathbf{Z}}^{(N-3)[N-1]} \right]. \end{aligned}$$

Note that, the computation of $\mathbf{W}_{\mathbf{Z}}^{(k)[j]}$ depends on $\mathbf{W}_{\mathbf{Z}}^{(k-1)[j]}$ and $\mathbf{U}_{\mathbf{Z}}^{(k)[k-1]}$. This recursive dependencies can be seen as follows,

$$\mathbf{W}_{\mathbf{Z}}^{(k)[j]} = f\left(\mathbf{W}_{\mathbf{Z}}^{(k-1)[j]},\, \mathbf{U}_{\mathbf{Z}}^{(k)[k-1]} \right), \text{ for } 1 \leq k, j \leq N \tag{3.8a}$$

similarly,

$$\mathbf{U}_{\mathbf{Z}}^{(k)[i]} = f\left(\mathbf{U}_{\mathbf{Z}}^{(k-1)[i]},\, \mathbf{U}_{\mathbf{Z}}^{(k)[k-1]}\right), \text{ for } 1 \leq k < N \text{ and } 1 \leq i < k\,. \tag{3.8b}$$

In both cases, the recursions end with the solution of the trivial systems involving $\mathbf{W}^{(0)} = \mathbf{R}$:

$$\mathbf{W}^{(0)} \cdot \mathbf{W}_{\mathbf{Z}}^{(0)[j]} = \mathbf{\Delta}^{[j]}, \text{ for } 1 \leq j \leq N,$$

and

$$\mathbf{W}^{(0)} \cdot \mathbf{U}_{\mathbf{Z}}^{(0)[i]} = \mathbf{v}^{b[i]}, \text{ for } 1 \leq i \leq N\,.$$

From the previous analysis we derive a recursive Sherman-Morrison formula as follows. Define

$$f_{\mathcal{S}}\left(\mathbf{x},\, k\right) = \begin{cases} \mathbf{z} = \mathbf{R}^{-1} \cdot \mathbf{x}\,, & k = 0\,, \\ \mathbf{f} = f_{\mathcal{S}}\left(\mathbf{x},\, k-1\right)\,; & \\ \mathbf{g} = f_{\mathcal{S}}\left(\mathbf{v}^{b[k]},\, k-1\right)\,, & 1 \leq k \leq N\,, \\ \mathbf{z} = \mathbf{f} - \frac{1}{\gamma^{(k)}} \cdot \mathbf{g} \cdot \left[{\mathbf{v}^{b[k]}}^T \cdot \mathbf{f}\right]\,; & \end{cases} \tag{3.9}$$

where $\gamma^{(k)} = 1 + {\mathbf{v}^{b[k]}}^T \cdot \mathbf{g}$. The solution of the linear system (3.4) can be obtained as follows:

$$\mathbf{Z}_{\mathbf{X}^a}^{\mathrm{RSMF}} = \left[f_{\mathcal{S}}\left(\mathbf{\Delta}^{[1]},\, N\right),\, f_{\mathcal{S}}\left(\mathbf{\Delta}^{[2]},\, N\right),\, \ldots,\, f_{\mathcal{S}}\left(\mathbf{\Delta}^{[N]},\, N\right)\right] \tag{3.10}$$

The recursive the computations performed by (3.9) can be represented as a tree in which the solution $\mathbf{z} \in \mathbb{R}^{m \times 1}$ of each node depends on the computations of its left ($\mathbf{f} \in \mathbb{R}^{m \times 1}$) and right ($\mathbf{g} \in \mathbb{R}^{m \times 1}$) children (i.e., on the solutions of two linear systems). Figure 3.1 illustrates the derivation of linear systems in order to solve $\mathbf{W}^{(N)} \cdot \mathbf{Z}_{\mathbf{X}^{\mathbf{a}}} = \mathbf{\Delta}^{[j]}$ for $N = 3$ and $1 \leq j \leq N$.

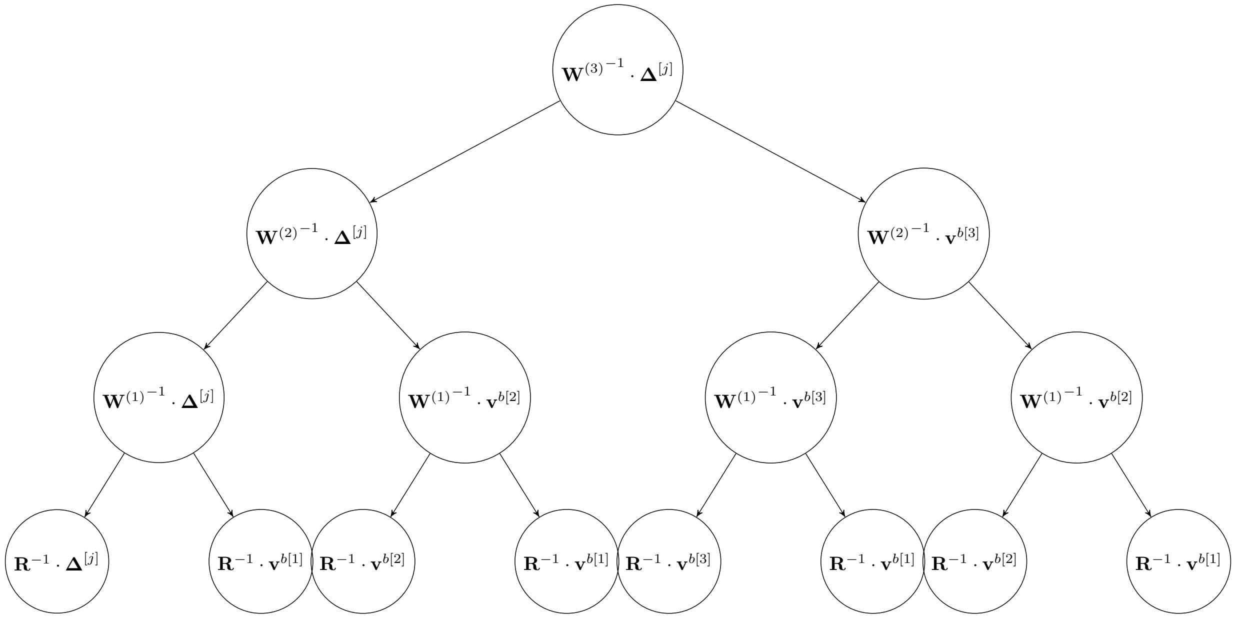


Figure 3.1: The recursive Sherman-Morrison formula (3.9) applied to solve the linear system $\mathbf{W}^{(N)} \cdot \mathbf{Z}_{\mathbf{X^a}} = \boldsymbol{\Delta}^{[j]}$ for $N = 3$ and $1 \leq j \leq N$.

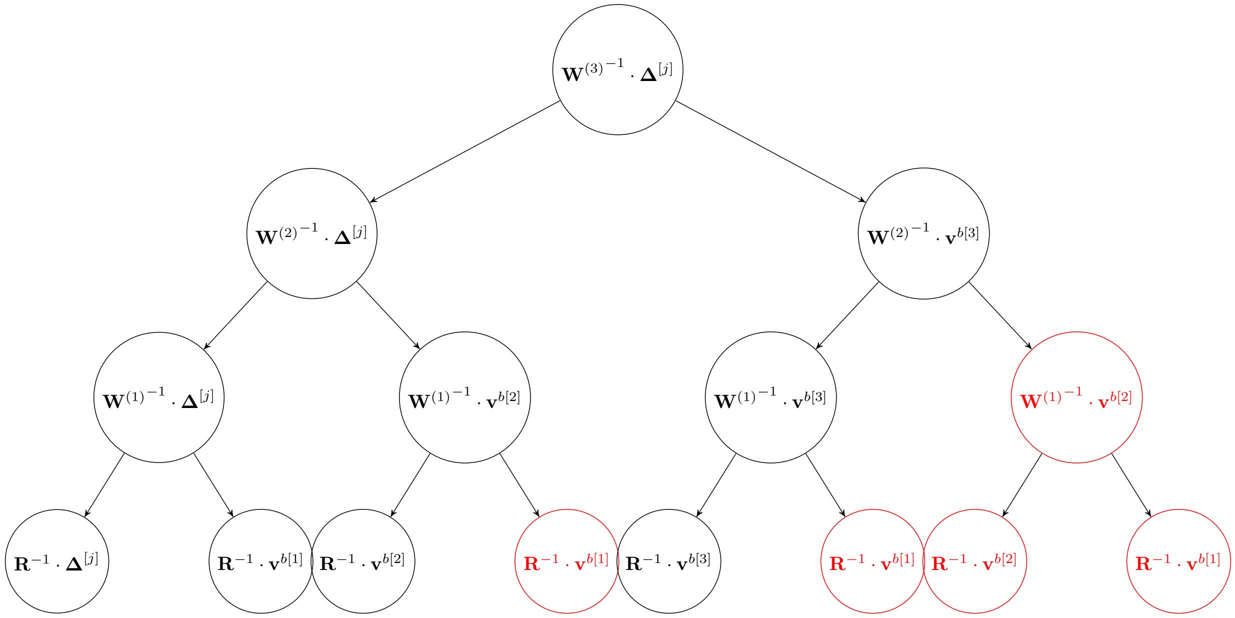


Figure 3.2: The recursive Sherman-Morrison formula (3.9) applied to solve the linear system $\mathbf{W}^{(N)} \cdot \mathbf{Z}_{\mathbf{X^a}} = \boldsymbol{\Delta}^{[j]}$ for $N = 3$ and $1 \leq j \leq N$. Red nodes represent repeated computations.

We see that (3.9) solves multiple times identical linear systems. For instance, the repeated computations performed in order to solve $\mathbf{W}^{(3)} \cdot \mathbf{Z}_{\mathbf{X}^a} = \boldsymbol{\Delta} \in \mathbb{R}^{m \times 3}$ are represented in Figure 3.2 as dashed nodes. There, for instance, the linear system $\mathbf{R} \cdot \mathbf{g} = \mathbf{v}^{b[1]}$ is solved four times in the last level. The total number of linear systems to solve is $\mathcal{O}\left(N \cdot 2^N\right)$, i.e., it increases exponentially with regard to the number of ensemble members if identical computations are not avoided. Next subsection discusses how to achieve this and obtain an efficient implementation of the recursive Sherman-Morrison formula.

3.1.1 An iterative Sherman-Morrison formula for matrix inversion

In order to avoid identical computations in Figure 3.1 we can solve the linear systems from the last level of the tree up to the root level. The key idea is to compute only once the common computations per level. From equations (3.8), at stage k, $\mathbf{U}_{\mathbf{Z}}^{(k,\,k-1)}$ is a common computation for (3.8b) and (3.8a).

We denote by $\mathbf{U} \in \mathbb{R}^{m \times N}$ and $\mathbf{Z} \in \mathbb{R}^{m \times N}$ the matrices holding partial results of the computations with regard to $\mathbf{V}^b$ and $\boldsymbol{\Delta}$, respectively. Then, level 0 can be computed as follows:

$$\mathbf{Z}^{(0)[j]} = \mathbf{R}^{-1} \cdot \boldsymbol{\Delta}^{[j]} \tag{3.11a}$$

$$\mathbf{U}^{(0)[j]} = \mathbf{R}^{-1} \cdot \mathbf{v}^{b[j]} \tag{3.11b}$$

and level $1 \leq k \leq N$

$$\boldsymbol{\theta}^{(k)} = \left[1 + \mathbf{v}^{b[k]^T} \cdot \mathbf{U}^{(k-1,k)}\right]^{-1} \cdot \mathbf{U}^{(k-1,k)}\,, \tag{3.12a}$$

$$\mathbf{Z}^{(k)[j]} = \mathbf{Z}^{(k-1)[j]} - \boldsymbol{\theta}^{(k)} \cdot \left[\mathbf{v}^{b[k]^T} \cdot \mathbf{Z}^{(k)[j]}\right], \text{ for } 1 \leq j \leq N\,, \tag{3.12b}$$

$$\mathbf{U}^{(k)[i]} = \mathbf{U}^{(k-1)[i]} - \boldsymbol{\theta}^{(k)} \cdot \left[\mathbf{v}^{b[k]^T} \cdot \mathbf{U}^{(k)[i]}\right], \text{ for } k+1 \leq i \leq N\,, \tag{3.12c}$$

where the solution of (3.4) for the j-th right-hand side is given by $\mathbf{Z}^{(N)[j]}$, for $1 \leq j \leq N$, or more general $\mathbf{Z}^{(N)} = \mathbf{Z}_{\boldsymbol{\delta}\mathbf{X}^a}$. Note that, at iteration k, $\boldsymbol{\theta}^{(k)}$ holds the pivot computation $\mathbf{U}_{\mathbf{Z}}^{(k-1)[k]}$ in (3.8). Moreover, the columns $1 \leq i \leq i$ of matrix $\mathbf{U}^{(k,i)}$ are not updated since they are not required for the next computations. Note that, the solution of linear system (3.4) is a function of three parameters:

$$\mathbf{Z}_{\mathbf{X}^a} = f^{*}_{\mathcal{S}}\left(\mathbf{R},\, \mathbf{V}^b,\, \boldsymbol{\Delta}\right)\,. \tag{3.13}$$

The computations performed by this iteration in order to solve $\mathbf{W}^{(N)} \cdot \mathbf{Z}_{\mathbf{X}^a} = \boldsymbol{\Delta}$ for $N = 3$ and $\mathbf{d} \in \left\{\boldsymbol{\Delta}^{[1]},\, \boldsymbol{\Delta}^{[2]},\, \boldsymbol{\Delta}^{[3]}\right\}$, $\mathbf{d} \in \mathbb{R}^{m \times 1}$ are shown in Figure 3.3.

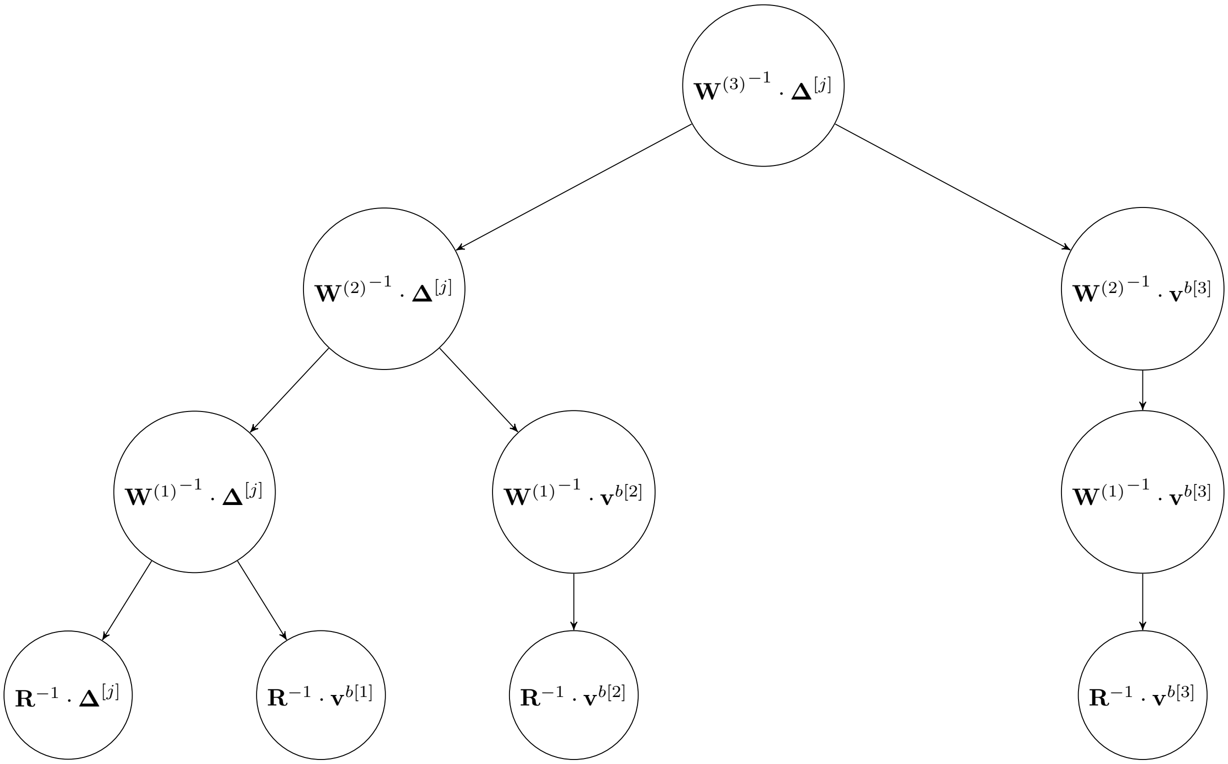


Figure 3.3: Necessary computations for the solution of $\mathbf{W}^{(3)} \cdot \mathbf{Z}_{\mathbf{X}^a} = \mathbf{\Delta}^{[j]} \in \mathbb{R}^{m \times 3}$ using the iterative Sherman-Morrison formula. This iterative version avoids all redundant computations.

Some key features of the iteration are highlighted next.

- The number of iterations is N.
- At level 0 matrices $\mathbf{Z}^{(0)[j]}$ and $\mathbf{U}^{(0)[j]}$ are computed according to (3.11). Notice, these computations does not imply a significant computational effort since they depend on the inverse of $\mathbf{R}$.
- The matrix $\boldsymbol{\alpha}^{(k)} \in \mathbb{R}^{m \times m}$ is never stored in memory. It can be represented implicitly by the matrices $\mathbf{R}$ and $\mathbf{V}^b$. This implicit representation realizes considerable memory savings, especially when $m \gg N$.
- At iteration k, only the vectors $\mathbf{U}^{(k)[i]}$ with $k < i \leq N$ are updated.

We now use the iterative Sherman-Morrison formula in the analysis step to obtain an efficient implementation of the Ensemble Kalman filter (SMEnKF). This filter is as follows:

$$\mathbf{X}^a \; = \; \mathbf{X}^b + \mathbf{S}^b \cdot \mathbf{V}^{b^T} \cdot \mathbf{Z}_{\mathbf{X}^a}^{\text{ISMF}} \in \mathbb{R}^{n \times N},$$

where

$$\mathbf{Z}_{\mathbf{X}^a}^{\text{ISMF}} = f_{\mathcal{S}}^{*}\left(\mathbf{R},\, \mathbf{V}^b,\, \boldsymbol{\Delta}\right) .$$

Since the matrix $\mathbf{R}$ has a simple structure its inverse is easy to obtain. Hence, under the assumptions done ($\mathbf{R}$ is easy to decompose), the computation of $\mathbf{Z}^{(0)[j]}$ and $\mathbf{U}^{(0)[j]}$, for $1 \leq j \leq N$, can be performed with no more than $\mathcal{O}\left(N^2 \cdot m\right)$ long operations.

3.1.2 Computational complexity

In the complexity analysis of the iterative Sherman-Morrison formula we count only the long operations (multiplications and divisions). Moreover, as discussed before, we make the assumptions presented in [TAB$^+$03, Man06], namely, the data error covariance matrix $\mathbf{R} \in^{m \times m}$ is inexpensive to decompose, and the observation operator $\mathbf{H}$ can be applied efficiently to any vector. We now analyze each each step of the iterative Sherman-Morrison formula when $\mathbf{R}$ is diagonal, the extension to nondiagonal data error covariance matrices is inmediate.

In the first step (3.11) each row i of matrices $\boldsymbol{\Delta} \in \mathbb{R}^{m \times N}$ and $\mathbf{V}^b \in \mathbb{R}^{m \times N}$ is divided by the corresponding component $\left\{\mathbf{R}\right\}_{i,i} \in \mathbb{R}^m$ in order to obtain $\mathbf{Z}^{(0)[j]}$ and $\mathbf{U}^{(0)[j]}$, for $1 \leq j \leq N$, respectively. This yields to $m \cdot N$ number of long operation for each matrix, therefore:

$$T_{\text{step}\,1}\left(N,\, m\right) = \mathcal{O}\left(2 \cdot m \cdot N\right) . \tag{3.14}$$

In the second step (3.12) we compute the vector $\boldsymbol{\theta}^{(k)} \in \mathbb{R}^m$, and the vectors $\mathbf{Z}^{(k)[j]}$ and $\mathbf{U}^{(k)[j]}$, for $1 \leq j \leq N$. The number of long operations for each of one are as follows:

$$\begin{aligned}
\boldsymbol{\theta}^{(k)} &= \overbrace{\mathbf{U}^{(k-1,k)} \cdot \frac{1}{1 + \underbrace{\mathbf{v}^{b[k]^T} \cdot \mathbf{U}^{(k-1)[k]}}_{m}}}^{m} , \\
\mathbf{Z}^{(k)[j]} &= \mathbf{Z}^{(k-1)[j]} - \overbrace{\boldsymbol{\theta}^{(k)} \cdot \left[\underbrace{\mathbf{v}^{b[k]^T} \cdot \mathbf{Z}^{(k-1)[j]}}_{N \cdot m}\right]}^{N \cdot m} ,\ \text{for } 1 \leq j \leq N, \\
\mathbf{U}^{(k)[i]} &= \mathbf{U}^{(k-1)[i]} - \overbrace{\boldsymbol{\theta}^{(k)} \cdot \left[\underbrace{\mathbf{v}^{b[k]^T} \cdot \mathbf{U}^{(k-1)[i]}}_{N \cdot m}\right]}^{N \cdot m} ,\ \text{for } k+1 \leq i \leq N,
\end{aligned}$$

Since the second step (3.12) is performed N times, the number of long operations can be expressed as:

$$\begin{aligned}
T_{\text{step}\,2}\left(N\,,m\right) &= \sum_{k=1}^{N}\left(\underbrace{2\cdot m}_{\boldsymbol{\theta}^{(k)}}+\underbrace{2\cdot m\cdot N}_{\mathbf{Z}^{(k)[j]}}+\overbrace{\sum_{\text{j}=1}^{k-1}\left(2\cdot m\right)}^{\mathbf{U}^{(k)[j]}}\right)\\
&= 2\cdot N\cdot m+2\cdot N^{2}\cdot m+\sum_{k=1}^{N}(k-1)\cdot 2\cdot m\\
&= 3\cdot N^{2}\cdot m+N\cdot m\,. \qquad (3.15)
\end{aligned}$$

Consequently, from (3.14)–(3.15), we have

$$\begin{aligned}
\mathbf{T}_{\text{ISMF}}\left(N,\,m\right) &= \underbrace{2\cdot m\cdot N}_{\text{step 1}}+\underbrace{3\cdot N^{2}\cdot m+N\cdot m}_{\text{step 2}}\\
&= 3\cdot\left(N^{2}\cdot m+N\cdot m\right),
\end{aligned}$$

which yields a complexity of

$$\mathcal{O}\left(N^{2}\cdot m\right)\,. \qquad (3.16)$$

Note that when $\mathbf{R}$ is not diagonal, under the assumptions done, the computations (3.11) of $\mathbf{Z}^{(0,j)}$ and $\mathbf{U}^{(0,j)}$, for $1\leq j\leq N$, can be efficiently performed in $\mathcal{O}\left(m\cdot N^{2}\right)$ long operations; the overall effort becomes $3\cdot\left(N^{2}\cdot m+N^{2}\cdot m\right)$. This leads to the same complexity (3.16) for $\mathbf{R}$ diagonal, block diagonal, or in general easy to decompose.

The overall complexity of the analysis step for the SMEnKF

$$\mathbf{X}^{a}=\mathbf{X}^{b}+\underbrace{\mathbf{S}^{b}\cdot\overbrace{\mathbf{V}^{b^{T}}\cdot\underbrace{\mathbf{Z}_{\mathbf{X}^{a}}^{\text{ISMF}}}_{\mathcal{O}(N^{2}\cdot m)}}^{\mathcal{O}\left(N^{2}\cdot m\right)}}_{\mathcal{O}(N^{2}\cdot n)}$$

is:

$$\mathcal{O}\left(N^{2}\cdot m+N^{2}\cdot n\right), \qquad (3.17)$$

The complexity of the SMEnKF is equivalent to the upper bounds os the methods described in [TAB^{+}03], as detailed in the Table 3.1. The term N^{3} does not appear in the upper-bound of the proposed method even when $\mathbf{R}$ is not diagonal. This implies that SMEnKF scales better than the other approaches when the size of the ensemble is increased.

Table 3.1: Summary of computational costs of the analysis steps for several ensemble filters. The costs are functions of the ensemble size N, number of observations m and state dimension n.

Analysis method	**Computational cost**
Direct [TAB⁺03]	$\mathcal{O}\left(N^2 \cdot m + N^3 + N^2 \cdot n\right)$
Serial [AC07]^1	$\mathcal{O}\left(N \cdot m + N \cdot m \cdot n\right)$
ETKF [And01]	$\mathcal{O}\left(N^2 \cdot m + N^3 + N^2 \cdot n\right)$
EAKF [And01]	$\mathcal{O}\left(N^2 \cdot m + N^3 + N^2 \cdot n\right)$
SMEnKF	$\mathcal{O}\left(N^2 \cdot m + N^2 \cdot n\right)$

Maponi [Map07] proposed a general approach based on the Sherman Morrison formula to solve linear systems. The application of this generic algorithm to (3.4) leads to an increased computational cost as the special structure of the system (and special structure of $\mathbf{R}$) are not exploited. The generic algorithm applied to EnKF analysis

$$\mathbf{W}^{(N)} \cdot \mathbf{Z}_{\mathbf{X}^a} = \mathbf{\Delta} \in \mathbb{R}^m \,, \text{ for } 1 \leq \mathrm{i} \leq \mathrm{N} \,, \tag{3.18}$$

uses the decomposition [Map07, Remark 1]:

$$\mathbf{W}^{(N)} = \mathbf{W}^{(0)} + \sum_{i=1}^{m} \mathbf{u}^{[i]} \cdot \mathbf{u}^{[i]^T} \,,$$

where $\mathbf{W}^{(0)} = \operatorname{diag}\left(\mathbf{R} + \mathbf{V}^b \cdot \mathbf{V}^{b^T}\right)$ and $\mathbf{W}^{(N)} - \mathbf{W}^{(0)} = \sum_{i=1}^{m} \mathbf{u}^{[i]} \cdot \mathbf{u}^{[i]^T}$. Thus, according to [Map07, Corollary 4], the linear system (3.4) can be solved with

$$\mathcal{O}\left(N \cdot m^3\right) .$$

long computations. Therefore the computational cost of the analysis step is:

$$\mathcal{O}\left(N \cdot m^3 + N \cdot n\right) \,, \tag{3.19}$$

which is larger than the computational cost of the SMEnKF when $m \gg N$. Moreover, according to [Map07, Theorem 3], when $m \gg N$, the solution of linear system (3.18) can be computed with no more than $\mathcal{O}\left(N^2 \cdot m + N^2\right)$ long operations. The resulting computational cost of the analysis step is:

$$\mathcal{O}\left(N^3 \cdot m + N^3 + N \cdot n\right) \,,$$

which is similar to the computational costs of the ETKF and EAKF methods when $m \gg N$. In addition, Maponi's algorithm requires the explicit representation in memory of the matrix $\mathbf{W}^{(N)}$, which, in practice, can be large. In contradistinction, $\mathbf{W}^{(N)}$ is not required explicitly in memory by our iterative Sherman Morrison formula.

Lastly, in [Map07, Section 2] the author shows that the intermediate matrices $\mathbf{W}^{(k)}$ in his method can be singular. In order to deal with this, Maponi proposes in [Map07, Algorithm 4] a method which makes use of partial pivoting in order to avoid singular matrices in $\mathbf{W}^{(k)}$. The stability conditions are discussed in [Map07, Theorem 12] where the author shows that [Map07, Algorithm 4] can be carried out since (in our notation)

$$\underbrace{\mathbf{I} + {\mathbf{V}^b}^T \mathbf{R}^{-1} \mathbf{V}^b}_{\mathbb{R}^{N \times N}} = \underbrace{\mathbf{R}^{-1} \cdot \mathbf{W}^{(N)}}_{\mathbb{R}^{m \times m}}, \tag{3.20}$$

is always positive definite, where the identity matrix $\mathbf{I}$ is consistent with the dimension. Note that (3.20) holds only for $N = m$ in [Map07]. Recent EnKF implementations based of Maponi's method [GM12] do not consider any stability conditions. In the next section we perform a stability analysis of the proposed iterative Sherman Morrison formula.

3.1.3 Stability Analysis

The solution of the linear system (3.4) by the iterative Sherman Morrison formula yields the next sequence of matrices during the computation of ${\mathbf{W}^{(N)}}^{-1}$:

$$\begin{array}{rcl}
{\mathbf{W}^{(0)}}^{-1} & = & \mathbf{R}^{-1} \\
{\mathbf{W}^{(1)}}^{-1} & = & {\mathbf{W}^{(0)}}^{-1} - \dfrac{1}{\gamma^{(1)}} \cdot \mathbf{U}^{(0)[1]} \cdot \left[{\mathbf{v}^{b[1]}}^T \cdot {\mathbf{W}^{(0)}}^{-1} \right] \\
{\mathbf{W}^{(2)}}^{-1} & = & {\mathbf{W}^{(1)}}^{-1} - \dfrac{1}{\gamma^{(2)}} \cdot \mathbf{U}^{(1)[2]} \cdot \left[{\mathbf{v}^{b[2]}}^T \cdot {\mathbf{W}^{(1)}}^{-1} \right] \\
\vdots & = & \vdots \\
{\mathbf{W}^{(k)}}^{-1} & = & {\mathbf{W}^{(k-1)}}^{-1} - \dfrac{1}{\gamma^{(k)}} \cdot \mathbf{U}^{(k-1)[k]} \cdot \left[{\mathbf{v}^{b[k]}}^T \cdot {\mathbf{W}^{(k-1)}}^{-1} \right]
\end{array}$$

where $\gamma^{(k)} = 1 + {\mathbf{v}^{b[k]}}^T \cdot \mathbf{U}^{(k-1)[k]}$, for $1 \le k \le N$. The following situations may affect the proposed method:

1. If any step produces $\gamma^{(k)} = 0$, then subsequent steps cannot proceed.
2. Round-off errors can be considerably amplified if $\gamma^{(k)} \approx 0$ (numerical instability).
3. If any matrix $\mathbf{W}^{(k)}$ in the sequence:

   $$\left\{ \mathbf{W}^{(0)}, \mathbf{W}^{(1)}, \dots, \mathbf{W}^{(N)} \right\},$$

   is singular, the algorithm cannot proceed.

We now show that the positive definiteness of the covariance matrix $\mathbf{R}$ is a sufficient in order to guarantee the stability of the iterative Sherman Morrison formula.

Theorem 1. *Assume that* $\mathbf{R}$ *is positive definite with* $\boldsymbol{\varrho}^T \cdot \mathbf{R} \cdot \boldsymbol{\varrho} \geq \alpha \cdot \|\boldsymbol{\varrho}\|^2$ *for any* $\boldsymbol{\varrho} \in \mathbb{R}^m$. *Then all matrices* $\mathbf{W}^{(k)}$ *are positive definite with* $\boldsymbol{\varrho}^T \cdot \mathbf{W}^{(k)} \cdot \boldsymbol{\varrho} \geq \alpha \cdot \|\boldsymbol{\varrho}\|^2$ *for any* $\xi \in \mathbb{R}^m$.

Proof. First, $\mathbf{W}^{(0)} = \mathbf{R}$ is positive definite. Next, we proceed by finite induction and assume that $\mathbf{W}^{(k-1)}$ is positive definite with $\boldsymbol{\varrho}^T \cdot \mathbf{W}^{(k-1)} \cdot \boldsymbol{\varrho} \geq \alpha \cdot \|\boldsymbol{\varrho}\|^2$. From (3.3) we have that:

$$\mathbf{W}^{(k)} \;=\; \mathbf{W}^{(k-1)} + \mathbf{v}^{b[k]} \cdot \mathbf{v}^{b[k]^T},$$

and therefore $\mathbf{W}^{(k)}$ is also positive definite:

$$\boldsymbol{\varrho}^T \cdot \mathbf{W}^{(k)} \cdot \boldsymbol{\varrho} \;=\; \underbrace{\boldsymbol{\varrho}^T \cdot \mathbf{W}^{(k-1)} \cdot \boldsymbol{\varrho}}_{\geq \alpha \, \|\boldsymbol{\varrho}\|^2} + \underbrace{\left[\boldsymbol{\varrho}^T \cdot \mathbf{v}^{b[k]}\right]^2}_{\geq 0} \geq \alpha \cdot \|\boldsymbol{\varrho}\|^2 \quad \forall \, \boldsymbol{\varrho} \in \mathbb{R}^m \,.$$

□

Theorem 2. *Assume that* $\mathbf{R}$ *is positive definite. The sequence of values* $\gamma^{(k)}$ *generated by the algorithm are strictly greater than one for all* $1 \leq k \leq N$.

Proof. By the iterative Sherman Morrison formula, the common computations $\mathbf{U}^{(k-1,k)}$ are given by:

$$\begin{aligned}
\mathbf{U}^{(0)[1]} &= \mathbf{R}^{-1} \cdot \mathbf{v}^{b[1]} = \boldsymbol{\alpha}^{(0)^{-1}} \cdot \mathbf{v}^{b[1]} \\
\mathbf{U}^{(1)[2]} &= \underbrace{\left[\mathbf{I} - \frac{1}{\gamma^{(1)}} \cdot \mathbf{U}^{(0)[1]} \cdot \mathbf{v}^{b[1]}\right] \cdot \mathbf{W}^{(0)^{-1}} \cdot \mathbf{v}^{b[2]}}_{\mathbf{W}^{(1)}} \\
\vdots &= \vdots \\
\mathbf{U}^{(k)[k+1]} &= \underbrace{\left[\mathbf{I} - \frac{1}{\gamma^{(k)}} \cdot \mathbf{U}^{(k-1)[k]} \cdot \mathbf{v}^{b[k]}\right] \cdot \mathbf{W}^{(k)^{-1}} \cdot \mathbf{v}^{b[k+1]}}_{\mathbf{W}^{(k)}}
\end{aligned}$$

Since $\mathbf{W}^{(k-1)}$ is positive definite we have:

$$\gamma^{(k)} = 1 + \mathbf{v}^{b[k]^T} \cdot \mathbf{U}^{(k-1)[k]} = 1 + \underbrace{\mathbf{v}^{b[k]^T} \cdot \mathbf{W}^{(k-1)} \cdot \mathbf{v}^{b[k]}}_{>0} > 1 \,,$$

consequently $\gamma^{(k)} > 1$ for all $1 \leq k \leq N-1$. □

Theorem 1 leads directly to the following

Theorem 3. *Assume that* $\mathbf{R}$ *is positive definite. At iteration* k, *the linear system:*

$$\mathbf{W}^{(k)} \cdot \mathbf{Z}^{(k)[j]} = \boldsymbol{\Delta}^{[j]} \,, \; \textit{for } 1 \leq j \leq Nens,$$

has a unique solution, for $1 \leq k \leq N$.

3.1.4 Pivoting

Theorem 2 shows that $\gamma^{(k)}$ values cannot be near zero. Due to this, we expect that the round-off errors will not increase considerably during an iteration of the iterative Sherman Morrison formula since:

$$\frac{1}{\gamma^{(k)}} \in (0\,,\,1)\,.$$

The following pivoting strategy can be (optionally) applied in order to further decrease round-off error accumulation. Formally, at iteration k, prior the matrix computations (3.12), we look for a column index i_k such that:

$$i_k = \arg\max_{i} \left\{ \left| 1 + \mathbf{v}^{b[i]^T} \cdot \mathbf{U}^{(k)[i]} \right| , k \leq i \leq N \right\} \,, \tag{3.21}$$

and then, the columns k and i_k are interchanged in matrices $\mathbf{V}^b$ and $\mathbf{U}^{(k)}$.

The iterative Sherman Morrison formula with pivoting gives the next computational cost:

$$\begin{aligned} \mathrm{T}_{\mathrm{SMF}}^{\mathrm{PIV}}(N,m) &= \underbrace{3\cdot\left(N^2\cdot m + N\cdot m\right)}_{\mathbf{T}NmSMF} + \underbrace{\sum_{k=1}^{N} \underbrace{\sum_{\mathrm{i}=1}^{k} m}_{(3.21)}}_{\text{Pivoting}}\,, \\ &= \frac{7}{2}\cdot\left(N^2\cdot m + N\cdot m\right) \end{aligned}$$

which yields to:

$$\mathrm{T}_{\mathrm{SMF}}^{\mathrm{PIV}}(N,m) \in \mathcal{O}\left(N^2\cdot m\right)\,,$$

from which we can conclude that seeking the maximum value of $\gamma^{(k)}$ according to (3.21) does not increase the computational cost of the iterative Sherman Morrison formula. Consequently, the overall complexity in the analysis step of SMEnKF remains bounded by (3.17).

3.2 Experimental Results

In this section several computation tests are conducted in order to assess the accuracy and running time of the SMEnKF. The results are compared against the efficient EnKF implementations presented in section 2.2.1.

3.2.1 Experimental setting

The SMEnKF as well as the EnKF implementations based on Cholesky (EnKFCH) and SVD (EnKFSVD) are coded in Fortran 90. The LAPACK and BLAS libraries [ABD^{+}90] are utilized in order to obtain efficient implementations of the Cholesky and SVD decompositions, as well as of other computations in SMEnKF as follows:

- The matrix $\mathbf{W}^{(N)} \in \mathbb{R}^{m \times m}$ is built using DSYRK function as follows:
  $$\mathbf{W}^{(N)} = \alpha \cdot \mathbf{V}^{b} \cdot {\mathbf{V}^{b}}^{T} + \beta \cdot \mathbf{R} \in \mathbb{R}^{m \times m}\,,$$
  with $\alpha = (N-1)^{-1}$ and $\beta = 1$.
- The functions DPOTRF and DPOTRI are used to compute the Cholesky decomposition of matrix $\mathbf{W}$. Only the upper triangular of $\mathbf{W}^{(N)}$ is stored.
- The SVD decomposition is performed by the DGESVD function. Only the first N pair of eigenvectors/values are computed. The right eigenvectors are not computed since they are not required.
- In the SMEnKF, the update
  $$\mathbf{Z}^{(k)[j]} = \mathbf{Z}^{(k-1)[j]} - \alpha \cdot \mathbf{x} \cdot \mathbf{y}^{\mathrm{T}}\,,\ \text{for } 1 \leq j \leq N,$$
  is performed making use of the DGER function with $\alpha = {\gamma^{(k)}}^{-1}$, $\mathbf{x} = \mathbf{U}^{(k-1)[k]}$, and $\mathbf{y} = {\mathbf{v}^{b[k]}}^{\mathrm{T}} \cdot \mathbf{Z}^{(k-1)[j]}$. $\mathbf{U}^{(k)}$ is updated in a similar way.

In order to measure the quality of the solutions we employ the following performance metrics. The Elapsed Time (ET) measures the overall simulation time for a method $*$. This metric is defined as

$$\mathrm{ET}(*) = \mathrm{Forecast}_{*} + \mathrm{Analysis}_{*} \tag{3.22}$$

where $\mathrm{Forecast}_{*}$ and $\mathrm{Analysis}_{*}$ are the overall running times for the forecast and analysis steps respectively.

The Root Mean Square Error (RMSE) is defined as

$$\varepsilon\,(*) = \mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{N_{steps}}} \cdot \left(\sum_{\mathrm{t}=1}^{\mathrm{N_{steps}}} \mathrm{RSE}_{\mathrm{t}}^{2} \right)}$$

where $\mathrm{N_{steps}}$ is the number of time steps and $\mathrm{RSE_t}$ is the Root Square Error at time t defined as follows:

$$\mathrm{RSE_t} = \|\mathbf{x}_t^{*} - \overline{\mathbf{x}}_t^{a}\|_2$$

where $\mathbf{x}^*$ is the reference vector state at time t, and $\overline{\mathbf{x}}_t^a$ is the analysis ensemble mean at time t. As can be seen the RMSE measures in average the distance between a reference solution ($\mathbf{x}^*$) and the given solution ($\overline{\mathbf{x}}^a$).

The EnKF implementations are tested using a quasi-geostrophic (QG) model [CB94, SO08c] which defines the model operator ($\mathcal{M}$) in the EnKF experiments. To compare the performance of different EnKF implementations we measure the elapsed times and the accuracy of analyses for different values of m and N.

3.2.2 Quasi-geostrophic model

The Earth's ocean has a complex flow system influenced by the rotation of the Earth, the density stratification due to temperature and salinity, as well as other factors. QG is a simple model which mimics the behavior of the ocean. It is defined by the following partial differential equations [CB94, SO08c]:

$$\begin{aligned} \frac{\partial \omega}{\partial t} + r_n \cdot J(\psi, \omega) + \beta \cdot \frac{\partial \psi}{\partial x} &= -r_b \cdot \psi + r_h \cdot \omega - r_g \cdot \Delta^2 \psi \\ &+ \underbrace{\sin(2\pi y)}_{\text{External Force}} \end{aligned} \tag{3.23}$$

where ψ is the stream function, ω is the vorticity, and Δ is the Laplacian operator

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \,,$$

F is the Froud number, r_n represents the Rossby number, r_b is the bottom friction, r_h is the horizontal friction and r_g is the biharmonic horizontal friction and x and y represent the horizontal and vertical spatial coordinates. Note that the stream function and the potential vorticity are related via the Laplacian operator $\omega = \Delta\psi$. This elliptic property reflects the assumption that the flow is geostrophically balanced in the horizontal direction, and hydrostatically balanced in the vertical direction.

The QG experiment studies the behavior of EnKF implementations when $m \gg N$ as is usually the case in practice. We consider three different grids, denoted $Q_{D_1 \times D_2}$, where the number of horizontal and vertical grid points in space are D_1 and D_2, respectively. Specifically, we employ in experiments $Q_{33\times 33}$ (small instance), $Q_{65\times 65}$ (medium instance) and $Q_{129\times 129}$ (large instance). The horizontal and vertical dimensions of the grid are denoted by $\mathrm{L_x}$ and $\mathrm{L_y}$ respectively. For all the instances the parameters $r_b = 10^{-6}$, $r_h = 10^{-7}$, $r_g = 2^{-12}$, $\beta = 1.0$ and $r = 10^{-5}$ are fixed. The other parameters values are summarized in Table 3.2.

Table 3.2: Parameter values for the QG model instances considered. $\mathrm{L_x}$ and $\mathrm{L_y}$ represent the horizontal and vertical grid sizes, and N and M are the number of horizontal and vertical grid points, respectively.

Instance	$\mathrm{L_x}$	$\mathrm{L_y}$	D_1	D_2
$Q_{33\times 33}$	0.4	0.4	33	33
$Q_{65\times 65}$	1.0	1.0	65	65
$Q_{129\times 129}$	1.0	1.0	129	129

The experimental settings are described below.

- There are 1200 time steps, each of one representing 1.27 days in the ocean.
- The vorticity of the ocean at each grid point provides a component of the vector state.
- The computation of the stream function is done through the solution of the Helmholtz equation [OL99].
- Homogeneous Dirichlet boundary conditions are assumed. Due to this, the boundaries of the grid are not mapped into the state vector, and $n = (\mathrm{D_1} - 2) \cdot (\mathrm{D_2} - 2)$.
- The initial ensemble members are

  $$\mathbf{x}^{\mathrm{b}}i = \mathbf{x}^* + \varepsilon_{\mathrm{i}}^{\mathrm{B}} \cdot \underbrace{\left(\frac{1}{n} \cdot \sum_{\mathrm{k=1}}^{n} |\mathbf{x}^*{}_{\mathrm{k}}|\right)}_{\mathbf{C}} \in \mathbb{R}^{n\times 1}\,,$$

  for $1 \leq \mathrm{i} \leq \mathrm{N}$, where ε^{B} is drawn from a normal distribution with zero mean and covariance matrix $\mathbf{B}_0 = \sigma_B^2 \cdot \mathbf{I} \in \mathbb{R}^{n\times n}$. For testing purposes, three values are assumed for the standard deviation of background errors $\sigma_B \in \{0.05, 0.1, 0.15\}$.
- The number of observation per simulation, for each size (n) of the model state , is defined as follows:

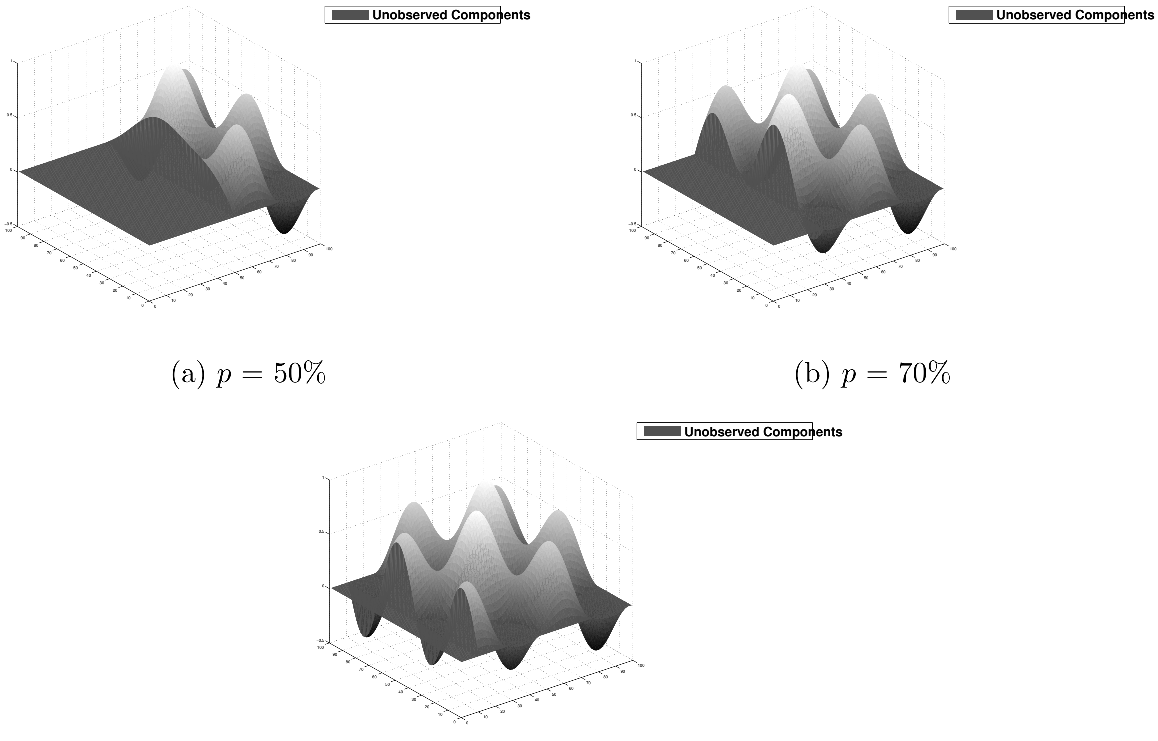
  $$m = p \cdot n,$$

  where p is the percentage of components observed from the model state. We consider $p \in \{50\%, 70\%, 90\%\}$. The observed areas for each value of p are shown in Figure 3.4.
- Measurements are taken every 10 time units and they are constructed as follows:

  $$\mathbf{y}_i = \mathbf{H} \cdot \mathbf{x}^* + \boldsymbol{\epsilon}_i \in \mathbb{R}^{m\times 1}$$

  where $\boldsymbol{\epsilon}_i \in \mathcal{N}(\mathbf{0}, \mathbf{R})$ with $\mathbf{R} = \sigma_O^2 \cdot \mathbf{I} \in \mathbb{R}^{m\times m}$, and $\sigma_O = 0.01$.

- For the time evolution of the model, zero boundary conditions are assumed and the boundaries are not included onto the ensemble representation. Due to this, the dimension of the vector state $n = (\mathrm{N}-2)\cdot(\mathrm{M}-2)$.
- For each instance we consider simulations with $N \in \{20, 60, 100\}$ ensemble members. The number of ensemble members is one to two orders of magnitude smaller than the total number of observations.



(a) $p = 50\%$

(b) $p = 70\%$

(c) $p = 90\%$

Figure 3.4: Observed components in the model state for different values of p

The RSME values for analysis errors for the $Q_{33\times 33}$, $Q_{65\times 65}$ and $Q_{129\times 129}$ instances are shown in Tables 3.6, 3.7, and 3.8, respectively. The results depend on the number of ensemble members (N), the number of observations (m), and the deviation of the initial ensemble mean (σ_B). The RSME quantifies errors in the stream function ψ. In terms of accuracy there is no significant difference between different EnKF implementations. As expected, when the error in the initial ensemble is increased, the accuracy in the analysis decreases. The error does not show an exponential growth, even when the number of components in the model state (n) is much larger than the number of ensemble members (e.g., for the $Q_{129\times 129}$ instance). When the number of ensemble members is increased, the analysis error is decreased. This is illustrated by the snapshots of the $Q_{33\times33}$ simulation over 1200 time steps presented in Figure 3.5. There, we can clearly see that the ensemble of size 100 provides a better estimation ($\overline{\mathbf{x}}^a$) to the true state of the model ($\mathbf{x}^*$) than the ensembles of sizes 20 and 60. Additionally, the number of observations plays an important role in the estimation of the true model state when the size of the vector state is much larger than the number of ensemble members.

The ET values for the $Q_{33\times33}$, $Q_{65\times65}$ and $Q_{129\times129}$ instances are shown in Tables 3.6, 3.7 and 3.8, respectively. The time is expressed in seconds (s) if it is below 30 minutes, and otherwise is expressed in minutes (min) and hours (h). The EnKCH shows good performance when the number of observations is small. From Table 3.3 (the blocks where the number of observations are 480, 672 and 864) we see that the EnKFCH performance is more sensitive to the number of observations than to the number of ensemble members. Due to this, small increments in the number of observed components increases considerably the elapsed time of this method. This EnKF implementation is not suitable for a large number of observations. For instance, the elapsed time for the $Q_{129\times129}$ instance is not presented since each simulation takes more than 4 days in order to be completed. Thus, in practice, this method is prohibitive.

The EnKFSVD shows a good performance for the $Q_{33\times33}$ and $Q_{65\times65}$ instances. We note that, for those instances, when the number of ensemble members is the largest tested ($N = 100$), the EnKFSVD performs relatively better than the SMEnKF implementation. We say relatively because the difference is just given by seconds. This is expected since both methods provide similar computational complexities. However, it is possible that these two methods scale different with regard to the dimension of the model, the number of observations and the size of the ensemble.

The SMEnKF is sensible to the number of ensemble members when the condition $m \gg N$ is not fully satisfied. For instance, consider the instances $Q_{33\times33}$ and $Q_{65\times65}$. There, the SMEnKF performs well when the number of observations and ensemble members are relatively close. This obeys to the computational efforts described in the Table 3.1. We note that the term N^3 does not appear in the computational cost of the proposed implementation but it appears on the EnKFSVD bound (2.8).

$$\mathcal{O}\left(n \cdot N^2 + m \cdot N^2 + m \cdot N + m\right) .$$

Thus, when the number of observations is not much greater than the number of ensemble members, the SMEnKF provides better performance than the EnKFSVD. However, for those scenarios, when the size of the ensemble is increased and the number of observations is hold, the SMEnKF performance is altered considerably. Recall that, the computational effort of the iterative Sherman Morrison formula is given by $\mathcal{O}\left(N^2 \cdot m\right)$, therefore we can expect low performances when m is not much greater than N. On the other hand, when the number of observations is several order of magnitude larger than the number of ensemble members (i.e. in $QG_{129\times129}$ is approximately 2), the SMEnKF provides the best performance. For instance, in some cases, the SMEnKF is 2.12 min faster than the EnKFSVD. This magnitude is large enough considering the size of the model and the similar theoretical complexity for both methods. Then, even when both methods provide the same theoretical computational cost, they do not scale similar, the SMEnKF is less sensitive to increments in the number of observations than the EnKFSVD when $m \gg N$, which is the relevant case in practice. Thus, the results of this test leads to conclude that the SMEnKF is not sensitive to the increase in the number of observations when this is considerably larger than the number of ensemble members, making it attractive for implementation with large-scale observational systems.

Table 3.5: Computational times for several EnKF implementations applied to the $Q_{129\times 129}$ instance. Different numbers of ensemble members and numbers of observations are considered.

N	m	σ^B	SMEnKF	EnKFSVD
20	8064	0.05	239.4 s	286.4 s
		0.10	244.4 s	293.1 s
		0.15	241.8 s	287.9 s
	11290	0.05	247.0 s	289.6 s
		0.10	248.7 s	287.7 s
		0.15	248.0 s	301.0 s
	14516	0.05	251.9 s	295.0 s
		0.10	254.3 s	291.0 s
		0.15	254.8 s	290.8 s
60	8064	0.05	669.9 s	713.0 s
		0.10	692.4 s	718.5 s
		0.15	694.3 s	725.7 s
	11290	0.05	724.3 s	775.0 s
		0.10	749.2 s	777.9 s
		0.15	745.1 s	776.7 s
	14516	0.05	797.1 s	843.6 s
		0.10	792.1 s	845.4 s
		0.15	776.1 s	836.5 s
100	8064	0.05	1252.2 s	1338.7 s
		0.10	1223.5 s	1300.7 s
		0.15	1238.7 s	1301.5 s
	11290	0.05	1410.7 s	1441.7 s
		0.10	1285.9 s	1418.0 s
		0.15	1387.5 s	1419.9 s
	14516	0.05	1508.4 s	1595.5 s
		0.10	1500.8 s	1563.9 s
		0.15	1501.3 s	1556.2 s

Table 3.3: Computational times for several EnKF implementations applied to the $Q_{33\times 33}$ instance. Different numbers of ensemble members and numbers of observations are considered.

N	m	σ_B	SMEnKF	EnKFCH	EnKFSVD
20	480	0.05	15.2 s	33.4 s	17.2 s
		0.10	15.4 s	32.8 s	17.4 s
		0.15	15.3 s	33.1 s	17.2 s
	672	0.05	15.6 s	61.9 s	17.0 s
		0.10	15.4 s	62.9 s	17.6 s
		0.15	15.5 s	62.1 s	17.2 s
	864	0.05	15.9 s	113.7 s	17.7 s
		0.10	15.8 s	116.3 s	17.4 s
		0.15	16.0 s	118.4 s	17.7 s
60	480	0.05	40.4 s	57.8 s	40.9 s
		0.10	40.5 s	57.5 s	41.5 s
		0.15	40.2 s	57.8 s	41.3 s
	672	0.05	42.3 s	90.3 s	43.4 s
		0.10	42.6 s	89.8 s	43.1 s
		0.15	42.7 s	90.3 s	43.1 s
	864	0.05	44.3 s	150.6 s	44.4 s
		0.10	44.4 s	156.7 s	44.2 s
		0.15	44.5 s	154.3 s	44.3 s
100	480	0.05	70.0 s	83.2 s	69.1 s
		0.10	70.1 s	83.2 s	68.7 s
		0.15	70.1 s	83.4 s	68.6 s
	672	0.05	75.5 s	118.9 s	72.5 s
		0.10	75.5 s	119.5 s	72.7 s
		0.15	75.5 s	120.1 s	72.5 s
	864	0.05	80.9 s	209.1 s	76.7 s
		0.10	80.9 s	212.7 s	77.1 s
		0.15	81.0 s	202.6 s	76.6 s

Table 3.4: Computational times for several EnKF implementations applied to the $Q_{65\times 65}$ instance. Different numbers of ensemble members and numbers of observations are considered.

N	m	σ^B	SMEnKF	EnKFCH	EnKFSVD
20	1984	0.05	60.0 s	44.9 min	68.9 s
		0.10	60.2 s	33.4 min	67.9 s
		0.15	60.7 s	45.4 min	68.5 s
	2778	0.05	61.5 s	1.8 h	67.8 s
		0.10	61.3 s	1.3 h	70.2 s
		0.15	61.0 s	1.4 h	68.5 s
	3572	0.05	62.4 s	3.8 h	70.3 s
		0.10	62.1 s	3.3 h	71.0 s
		0.15	62.1 s	3.0 h	70.4 s
60	1984	0.05	161.4 s	45.0 min	168.3 s
		0.10	161.2 s	55.3 min	166.5 s
		0.15	162.3 s	51.9 min	167.8 s
	2778	0.05	170.4 s	2.4 h	172.8 s
		0.10	170.2 s	1.9 h	175.4 s
		0.15	169.9 s	2.5 h	174.6 s
	3572	0.05	179.2 s	4.1 h	180.7 s
		0.10	178.2 s	2.9 h	181.5 s
		0.15	177.5 s	4.3 h	179.9 s
100	1984	0.05	281.8 s	52.4 min	279.3 s
		0.10	281.5 s	40.1 min	280.4 s
		0.15	280.8 s	58.4 min	280.5 s
	2778	0.05	303.2 s	1.7 h	295.6 s
		0.10	303.4 s	1.7 h	294.8 s
		0.15	303.3 s	2.9 h	295.5 s
	3572	0.05	327.9 s	4.8 h	313.4 s
		0.10	327.2 s	4.0 h	313.4 s
		0.15	327.3 s	5.1 h	315.7 s

Table 3.6: Analysis RMSE for different EnKF implementations applied to the $Q_{33\times 33}$ instance. All methods give similar results. When the number of ensemble and/or observations is increased, the analysis accuracy is improved.

N	m	σ^B	SMEnKF	EnKFCH	EnKFSVD
20	480	0.05	1.71×10^{-4}	1.71×10^{-4}	1.71×10^{-4}
		0.10	3.42×10^{-4}	3.42×10^{-4}	3.42×10^{-4}
		0.15	5.13×10^{-4}	5.13×10^{-4}	5.13×10^{-4}
	672	0.05	1.68×10^{-4}	1.68×10^{-4}	1.68×10^{-4}
		0.10	3.37×10^{-4}	3.37×10^{-4}	3.37×10^{-4}
		0.15	0.106×10^{-4}	0.11×10^{-4}	0.11×10^{-4}
	864	0.05	1.68×10^{-4}	1.68×10^{-4}	1.68×10^{-4}
		0.10	3.37×10^{-4}	3.37×10^{-4}	3.37×10^{-4}
		0.15	0.106×10^{-4}	0.11×10^{-4}	0.11×10^{-4}
60	480	0.05	1.71×10^{-4}	1.71×10^{-4}	1.71×10^{-4}
		0.10	3.43×10^{-4}	3.43×10^{-4}	3.43×10^{-4}
		0.15	5.14×10^{-4}	5.14×10^{-4}	5.14×10^{-4}
	672	0.05	1.64×10^{-4}	1.64×10^{-4}	1.64×10^{-4}
		0.10	3.29×10^{-4}	3.29×10^{-4}	3.29×10^{-4}
		0.15	4.94×10^{-4}	4.94×10^{-4}	4.94×10^{-4}
	864	0.05	1.64×10^{-4}	1.64×10^{-4}	1.64×10^{-4}
		0.10	3.29×10^{-4}	3.29×10^{-4}	3.29×10^{-4}
		0.15	4.94×10^{-4}	4.94×10^{-4}	4.94×10^{-4}
100	480	0.05	1.62×10^{-4}	1.62×10^{-4}	1.62×10^{-4}
		0.10	3.23×10^{-4}	3.23×10^{-4}	3.23×10^{-4}
		0.15	4.84×10^{-4}	4.84×10^{-4}	4.84×10^{-4}
	672	0.05	1.54×10^{-4}	1.54×10^{-4}	1.54×10^{-4}
		0.10	3.10×10^{-4}	3.10×10^{-4}	3.10×10^{-4}
		0.15	4.66×10^{-4}	4.66×10^{-4}	4.66×10^{-4}
	864	0.05	1.44×10^{-4}	1.44×10^{-4}	1.44×10^{-4}
		0.10	2.90×10^{-4}	2.90×10^{-4}	2.90×10^{-4}
		0.15	4.35×10^{-4}	4.35×10^{-4}	4.35×10^{-4}

Table 3.7: Analysis RMSE for different EnKF implementations applied to the $Q_{65\times 65}$ instance. All methods give similar results.

N	m	σ_B	SMEnKF	EnKFCH	EnKFSVD
20	1984	0.05	2.38×10^{-4}	2.38×10^{-4}	2.38×10^{-4}
		0.10	4.77×10^{-4}	4.77×10^{-4}	4.77×10^{-4}
		0.15	7.16×10^{-4}	7.16×10^{-4}	7.16×10^{-4}
	2778	0.05	2.38×10^{-4}	2.38×10^{-4}	2.38×10^{-4}
		0.10	4.777×10^{-4}	4.77×10^{-4}	4.77×10^{-4}
		0.15	7.15×10^{-4}	7.15×10^{-4}	7.15×10^{-4}
	3572	0.05	2.38×10^{-4}	2.38×10^{-4}	2.38×10^{-4}
		0.10	4.77×10^{-4}	4.77×10^{-4}	4.77×10^{-4}
		0.15	7.15×10^{-4}	7.15×10^{-4}	7.15×10^{-4}
60	1984	0.05	2.34×10^{-4}	2.34×10^{-4}	2.34×10^{-4}
		0.10	4.69×10^{-4}	4.69×10^{-4}	4.69×10^{-4}
		0.15	7.04×10^{-4}	7.04×10^{-4}	7.04×10^{-4}
	2778	0.05	2.34×10^{-4}	2.34×10^{-4}	2.34×10^{-4}
		0.10	4.68×10^{-4}	4.68×10^{-4}	4.68×10^{-4}
		0.15	7.03×10^{-4}	7.03×10^{-4}	7.03×10^{-4}
	3572	0.05	2.34×10^{-4}	2.34×10^{-4}	2.34×10^{-4}
		0.10	4.68×10^{-4}	4.68×10^{-4}	4.68×10^{-4}
		0.15	7.03×10^{-4}	7.03×10^{-4}	7.03×10^{-4}
100	1984	0.05	2.37×10^{-4}	2.37×10^{-4}	2.37×10^{-4}
		0.10	4.74×10^{-4}	4.74×10^{-4}	4.74×10^{-4}
		0.15	7.10×10^{-4}	7.10×10^{-4}	7.10×10^{-4}
	2778	0.05	2.34×10^{-4}	2.34×10^{-4}	2.34×10^{-4}
		0.10	4.68×10^{-4}	4.68×10^{-4}	4.68×10^{-4}
		0.15	7.02×10^{-4}	7.02×10^{-4}	7.02×10^{-4}
	3572	0.05	2.32×10^{-4}	2.32×10^{-4}	2.32×10^{-4}
		0.10	4.64×10^{-4}	4.64×10^{-4}	4.64×10^{-4}
		0.15	6.97×10^{-4}	6.97×10^{-4}	6.97×10^{-4}

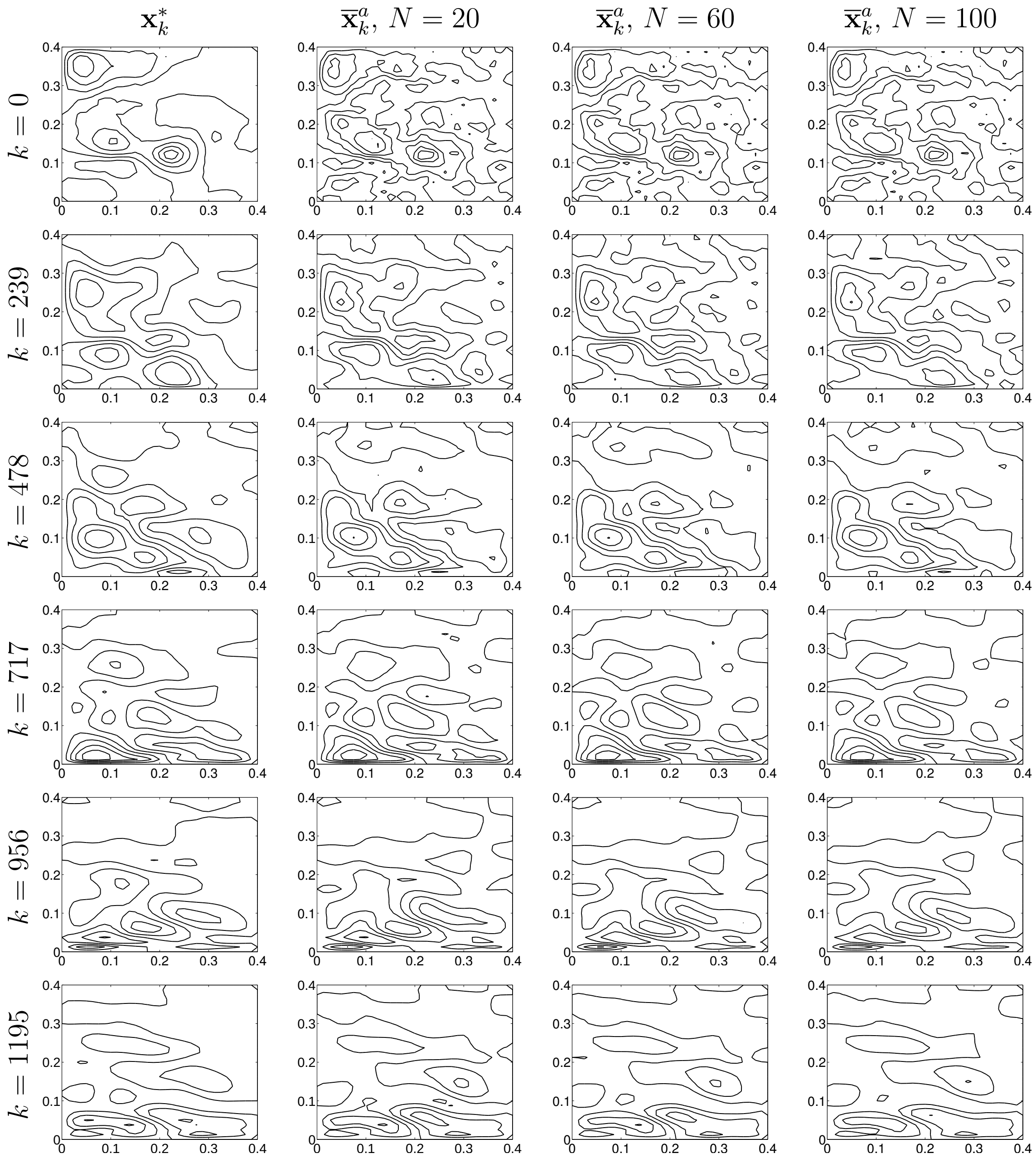


Figure 3.5: Snapshots of the $Q_{33\times33}$ simulation for $N = 20$,60 and 100 members, at the time steps $t = 0, 239, 478, 717, 956$ and 1195 (out of 1200). As expected, when the number of ensemble members is increased the estimation of the true state ($\mathbf{x}^*$) is improved (the RMSE is decreased).

Table 3.8: Analysis RMSE for different EnKF implementations applied to the $Q_{129\times129}$ instance. All methods give similar results.

N	m	σ_B	SMEnFK	EnKFSVD
20	8064	0.05	9.92×10^{-5}	9.92×10^{-5}
		0.10	1.98×10^{-4}	1.98×10^{-4}
		0.15	2.97×10^{-4}	2.97×10^{-4}
	11290	0.05	9.90×10^{-5}	9.90×10^{-5}
		0.10	1.97×10^{-4}	1.97×10^{-4}
		0.15	2.96×10^{-4}	2.96×10^{-4}
	14516	0.05	9.87×10^{-5}	9.87×10^{-5}
		0.10	1.97×10^{-4}	1.97×10^{-4}
		0.15	2.96×10^{-4}	2.96×10^{-4}
60	8064	0.05	9.74×10^{-5}	9.74×10^{-5}
		0.10	1.94×10^{-4}	1.94×10^{-4}
		0.15	2.92×10^{-4}	2.92×10^{-4}
	11290	0.05	9.63×10^{-5}	9.63×10^{-5}
		0.10	1.92×10^{-4}	1.92×10^{-4}
		0.15	2.89×10^{-4}	2.89×10^{-4}
	14516	0.05	9.67×10^{-5}	9.67×10^{-5}
		0.10	1.93×10^{-4}	1.93×10^{-4}
		0.15	2.90×10^{-4}	2.90×10^{-4}
100	8064	0.05	9.56×10^{-5}	9.56×10^{-5}
		0.10	1.91×10^{-4}	1.91×10^{-4}
		0.15	2.87×10^{-4}	2.87×10^{-4}
	11290	0.05	9.49×10^{-5}	9.49×10^{-5}
		0.10	1.89×10^{-4}	1.89×10^{-4}
		0.15	2.84×10^{-4}	2.84×10^{-4}
	14516	0.05	9.47×10^{-5}	9.47×10^{-5}
		0.10	1.89×10^{-4}	1.89×10^{-4}
		0.15	2.84×10^{-4}	2.84×10^{-4}

3.3 Conclusions

This chapter discusses an implementation of the EnKF based on an iterative application of the Sherman-Morrison formula. The algorithm exploits the special structure of the background error covariance matrix projected onto the observation space. The computational complexity of the new approach is equivalent to that of the best EnKF formulations available in the literature. A sufficient condition for the stability of the proposed method is the non-singularity of the data error covariance matrix, which is typically the case in practicee. In addition, a pivoting strategy is developed in order to reduce round-off error propagation without increasing the computational effort of the proposed method. The computational cost of this algorithm provides a better theoretical performance than other generic formulations of matrix inversion based on the Sherman Morrison formula available in the literature. To assess the accuracy and performance of the proposed implementation a tests have been carried out using a quasi-geostrophic model. All EnKF implementations tested (EnKFCH, EnKFSVD, SMEnKF) provide virtually identical analyses. However, the proposed Sherman-Morrison approach scales better than the others with regard to the number of observations. The parallel version of the new algorithm has a theoretical complexity that grows only linearly with the number of observations, and is therefore well suited for implementation in large scale data assimilation systems.

Chapter 4

An Ensemble Kalman Filter Implementation Based on Shrinkage Covariance Matrix Estimation

This chapter develops efficient ensemble Kalman filter (EnKF) implementations based on shrinkage covariance estimation. The forecast ensemble members at each step are used to estimate the background error covariance matrix via the Rao-Blackwell Ledoit and Wolf estimator, which has been specifically developed to approximate high-dimensional covariance matrices using a small number of samples. Two implementations are considered: in the EnKF Full-Space (EnKF-FS) approach the assimilation process is performed in the model space, while the EnKF Reduce-Space (EnKF-RS) formulation performs the analysis in the subspace spanned by the ensemble members. In the context of EnKF-RS, additional samples are taken from the normal distribution described by the background ensemble mean and the estimated background covariance matrix, in order to increase the size of the ensemble and reduce the sampling error of the filter. This increase in the size of the ensemble is obtained without running the forward model. After the assimilation step, the additional samples are discarded and only the model-based ensemble members are propagated further. Methodologies to reduce the impact of spurious correlations and under-estimation of sample variances in the context of the EnKF-FS and EnKF-RS implementations are discussed. An adjoint-free four dimensional extension of EnKF-RS is also discussed. Numerical experiments carried out with the Lorenz-96 model and a quasi-geostrophic model show that the use of shrinkage covariance matrix estimation can mitigate the impact of spurious correlations during the assimilation process.

This chapter is organized as follows. Section 4.1 reviews shrinkage covariance estimation methods. In section 4.2 the two novel implementations of the ensemble Kalman filter based on shrinkage covariance estimation are proposed. Experimental results making use of a quasi-geostrophic model are given in section 4.4. Section 6.4 summarizes the conclusions of this work.

4.1 Shrinkage covariance matrix estimation

Many problems in science and engineering require an estimate of a covariance matrix and/or its inverse, where the matrix dimension n is large compared to the sample size N. Different applications ranging from variational [CJAS10b, HEE01] to sequential [CEK$^+$13, Zup09] data assimilation rely on accurately estimated covariance matrices.

Let

$$\mathbf{S} = \left[\mathbf{s}^{[1]},\, \mathbf{s}^{[2]},\, \dots,\, \mathbf{s}^{[N]}\right] \in \mathbb{R}^{n\times N}, \tag{4.1}$$

be a sample of independent identical distributed n-dimensional Gaussian vectors

$$\mathbf{s}^{[i]} \sim \mathcal{N}(\mathbf{0},\, \mathbf{Q}) \in \mathbb{R}^{n\times 1}.$$

A common approach is to estimate $\mathbf{Q} \in \mathbb{R}^{n\times n}$ by the sample covariance matrix $\mathbf{C}_s$

$$\mathbf{C}_s \;=\; \frac{1}{N-1} \cdot \sum_{i=1}^{N} \mathbf{s}^{[i]} \cdot \mathbf{s}^{[i]^T} \in \mathbb{R}^{n\times n}\,. \tag{4.2}$$

$\mathbf{C}_s$ is the maximum likelihood estimator when it is invertible [LW04]. However, under the condition $n \gg N$, this is not the case. The simpler thing to do in order to deal with the rank-deficiency of $\mathbf{C}_s$ is to impose some structure (i.e., localization in ensemble-based methods). However, in the absence of prior information about the true structure of $\mathbf{Q}$, $\mathbf{C}_s$ will poorly describe the correlations between different components of the samples (4.1). In order to improve estimation of covariance matrices many methods have been proposed in the literature based on tapering procedures [CZZ10, CWM12], minimizing the log-determinant divergence [RWRY11], and greedy methods [JJR12]. Another class of well-conditioned estimators is based on shrinkage approximations [Far78, DMUN13, CM14, Par14, BP14, FS11]. These approximations express the estimated covariance matrix as a weighted average of some target matrix $\mathbf{T} \in \mathbb{R}^{n\times n}$ and the empirical covariance matrix (4.2). To better understand this assume that the components of $\mathbf{s}^{[i]}$ are uncorrelated for $1 \le i \le N$. A simple estimate of $\mathbf{Q}$ is given by

$$\mathbf{T} = \frac{\mathrm{tr}\left(\mathbf{C}_s\right)}{n} \cdot \mathbf{I}\,,$$

where $\mathbf{I}$ is the identity matrix in the n-dimensional space. Note that this structure will reduce the variance but will increase the bias when the diagonal assumption is not fulfilled. A reasonable trade-off is achieved by the shrinkage of $\mathbf{C}_s$ towards $\mathbf{T}$ and provides the followng class of estimators

$$\widehat{\mathbf{C}} = \lambda \cdot \mathbf{T} + (1-\lambda) \cdot \mathbf{C}_s \in \mathbb{R}^{n\times n}\,, \tag{4.3}$$

where $\lambda \in [0,\, 1]$. The problem is then reduced to find an optimal value for λ in which the squared loss

$$\lambda^* = \arg\min_{\lambda} \mathbb{E}\left[\left\|\widehat{\mathbf{C}} - \mathbf{Q}\right\|_F^2\right] \tag{4.4}$$

is minimized, where $\|\bullet\|_F$ denotes the Frobenius norm. There are many shrinkage based estimators derived from the minimization of (4.4) subject to (4.3). We restrict our exploration to three well-accepted methods: the Ledoit and Wolf estimator [LW04], the Rao-Blackwell Ledoit and Wolf estimator [CWEH10] and the oracle approximating shrinkage estimator [CWEH10, CWH11].

The distribution-free Ledoit and Wolf (LW) estimator [LW04] has been proven more accurate than the sample covariance matrix and some estimators proposed in finite sample decision theory. Moreover, it is better conditioned than the true covariance matrix [LW04]. The optimal λ value proposed by this estimator is

$$\lambda_{\text{LW}} = \min\left(\frac{\sum_{i=1}^{N}\left\|\mathbf{C}_s - s_i \cdot s_i^T\right\|_F^2}{N^2 \cdot \left[\operatorname{tr}\left(\mathbf{C}_s^2\right) - \frac{\operatorname{tr}^2(\mathbf{C}_s)}{n}\right]},\, 1\right) \tag{4.5}$$

and the LW estimator $\widehat{\mathbf{C}}_{\text{LW}}$ is obtained by using λ_{LW} in (4.3).

The Rao-Blackwell Ledoit and Wolf (RBLW) estimator [CWH11, CWEH10] provably improves the LW method in a square loss sense under Gaussian assumptions. The motivation of this estimator is that, under Gaussian assumptions, all the information required in order to get a well-conditioned estimate of $\mathbf{Q}$ is contained in $\mathbf{C}_s$. The proposed value for λ is

$$\lambda_{\text{RBLW}} = \min\left(\frac{\frac{N-2}{n} \cdot \operatorname{tr}\left(\mathbf{C}_s^2\right) + \operatorname{tr}^2\left(\mathbf{C}_s\right)}{(N+2) \cdot \left[\operatorname{tr}\left(\mathbf{C}_s^2\right) - \frac{\operatorname{tr}^2(\mathbf{C}_s)}{n}\right]},\, 1\right) \tag{4.6}$$

and the corresponding estimator $\widehat{\mathbf{C}}_{\text{RBLW}}$ is obtained by replacing (4.6) in the equation (4.3). In addition, in [CWH11, Theorem 2], it is proven that

$$\mathbb{E}\left[\left\|\widehat{\mathbf{C}}_{\text{RBLW}} - \mathbf{Q}\right\|_F^2\right] \leq \mathbb{E}\left[\left\|\widehat{\mathbf{C}}_{\text{LW}} - \mathbf{Q}\right\|_F^2\right],$$

which rigorously shows the RBLW estimator to be a better approximation of $\mathbf{Q}$ than the LW estimator under the Gaussian assumption.

All the estimators presented in this section provide well-conditioned approximations to the unknown covariance matrix $\mathbf{Q}$. We center our attention on the RBLW estimator since in high dimensional problems, such those found in data assimilation, this estimator can be implemented easily, and under the Gaussian assumption it provides better approximations than the LW estimator.

4.2 Ensemble Filters Based on Shrinkage Covariance Estimators

In this section, we propose two efficient implementations of the EnKF based on the RBLW estimator (4.6). Note that, the information contained in the sample covariance matrix (2.5b)

$$\mathbf{P}^b = \mathbf{S}^b \cdot {\mathbf{S}^b}^T \,, \tag{4.7a}$$

with

$$\mathbf{S}^b = \frac{1}{\sqrt{N-1}} \left[\cdot \mathbf{X}^b - \overline{\mathbf{x}}^b \cdot \mathbf{1}_N^T \right] \in \mathbb{R}^{n \times N}, \tag{4.7b}$$

is more than only background errors

$$\mathbf{P}^b = \mathbf{B} + \mathbf{M} \in \mathbb{R}^{n \times n} \,,$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the covariance of model errors. Errors coming from different sources are assumed to be uncorrelated. We seek to exploit the information brought by ensemble members and use the RBLW covariance estimator (4.6) to build a covariance matrix that better represent error correlations. The standard form of this estimator depends on the explicit representation of $\mathbf{P}^b$. The efficient implementation for high-dimensional covariance matrices presented in section 4.2.1 avoids the explicit computation of $\mathbf{P}^b$. Section 4.2.2 discusses two EnKF implementations based on the RBLW estimator. Section 4.2.3 develops an efficient sampling method in high dimensions for drawing samples from the prior error distribution based on the RBLW estimate. Finally, section 4.2.5 discusses the similarities and differences between the two proposed implementations.

4.2.1 RBLW estimator for covariance matrices in high-dimensions

Consider the sample covariance matrix (4.7a). In the context of data assimilation the RBLW estimator (4.3),(4.6) reads

$$\widehat{\mathbf{B}} = \lambda_{\widehat{\mathbf{B}}} \cdot \mu_{\mathbf{B}} \cdot \mathbf{I} + (1 - \lambda_{\widehat{\mathbf{B}}}) \cdot \mathbf{P}^b \in \mathbb{R}^{n \times n} \,, \tag{4.8a}$$

where

$$\mu_{\widehat{\mathbf{B}}} \;=\; \frac{\operatorname{tr}\left(\mathbf{P}^b\right)}{n} \,, \tag{4.8b}$$

$$\lambda_{\mathbf{B}} \;=\; \min\left(\frac{\frac{N-2}{n} \cdot \operatorname{tr}\left(\left[\mathbf{P}^b\right]^2 \right) + \operatorname{tr}^2\left(\mathbf{P}^b\right)}{(N+2) \cdot \left[\operatorname{tr}\left([\mathbf{P}^b]^2 \right) - \frac{\operatorname{tr}^2\left(\mathbf{P}^b\right)}{n} \right]} ,\, 1 \right) \,. \tag{4.8c}$$

Since the dimension of the model state is high ($n \sim \mathcal{O}\left(10^7\right)$), the direct computation of (4.8a) is impractical as it requires the explicit representation of the sample covariance matrix $\mathbf{P}^b$. An alternative manner to compute $\operatorname{tr}\left(\mathbf{P}^b\right)$ and $\operatorname{tr}\left([\mathbf{P}^b]^2\right)$ is proposed. Consider the eigenvalue decomposition of $\mathbf{P}^b$

$$\mathbf{P}^b = \mathbf{U}_{\mathbf{P}^b} \cdot \mathbf{\Sigma}_{\mathbf{P}^b} \cdot \mathbf{U}_{\mathbf{P}^b}^T \in \mathbb{R}^{n \times n}\,, \tag{4.9}$$

where $\mathbf{\Sigma}_{\mathbf{P}^b} \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose diagonal components σ_i, for $1 \le i \le n$, are the eigenvalues of $\mathbf{P}^b$ and $\mathbf{U}_{\mathbf{P}^b} \in \mathbb{R}^{n \times n}$ is a set of orthogonal basis vectors spanning the ensemble space (since $\mathbf{P}^b$ is rank deficient). By definition $\operatorname{tr}\left(\mathbf{P}^b\right) = \sum_{i=1}^{n} \sigma_i$ and $\operatorname{tr}\left(\left[\mathbf{P}^b\right]^2\right) = \sum_{i=1}^{n} \sigma_i^2$. Since there are only $N-1$ eigenvalues different from zero we obtain:

$$\operatorname{tr}\left(\mathbf{P}^b\right) = \sum_{i=1}^{N-1} \sigma_i\,, \qquad \operatorname{tr}\left(\left[\mathbf{P}^b\right]^2\right) = \sum_{i=1}^{N-1} \sigma_i^2\,,$$

and the computations in the set of equations (4.8) can be efficiently performed whenever the first $N-1$ eigenvalues of $\mathbf{P}^b$ can be easily obtained. Consider the singular value decomposition (SVD) of the matrix of member deviations (4.7b)

$$\mathbf{S}^b = \mathbf{U}_{\mathbf{S}^b} \cdot \widehat{\mathbf{\Sigma}}_{\mathbf{S}^b} \cdot \mathbf{V}_{\mathbf{S}^b}^T \in \mathbb{R}^{n \times N}\,, \tag{4.10}$$

where $\widehat{\mathbf{\Sigma}}_{\mathbf{S}^b} \in \mathbb{R}^{n \times N}$ is a diagonal matrix holding the singular values $\widehat{\sigma_i}$ of $\mathbf{S}$, for $1 \le i \le N$. Likewise, $\mathbf{U}_{\mathbf{S}} \in \mathbb{R}^{n \times n}$ and $\mathbf{V}_{\mathbf{S}^b} \in \mathbb{R}^{N \times N}$ are the left and right singular vectors, respectively. Since $\mathbf{P}^b = \mathbf{S}^b \cdot {\mathbf{S}^b}^T$ we have $\mathbf{\Sigma}_{\mathbf{P}^b} = \widehat{\mathbf{\Sigma}}_{\mathbf{S}} \cdot \widehat{\mathbf{\Sigma}}_{\mathbf{S}}^T$ and

$$\begin{aligned}
\operatorname{tr}\left(\mathbf{P}^b\right) &= \sum_{i=1}^{N-1} \sigma_i = \sum_{i=1}^{N-1} \widehat{\sigma_i}^2,\\
\operatorname{tr}\left(\left[\mathbf{P}^b\right]^2\right) &= \sum_{i=1}^{N-1} \sigma_i^2 = \sum_{i=1}^{N-1} \widehat{\sigma_i}^4.
\end{aligned}$$

The computational effort of the SVD decomposition (4.10) is $\mathcal{O}\left(N^2 \cdot n\right)$. The traces in (4.8) can be computing without calculating the sample covariance matrix $\mathbf{P}^b$ by making use of the inexpensive SVD decomposition of $\mathbf{S}$. None of the singular vector of $\mathbf{S}$ are required, but only the singular values $\widehat{\sigma_i}$, for $1 \le i \le N-1$. The parameter values in (4.8) are computed as follows:

$$\mu_{\widehat{\mathbf{B}}} = \frac{\sum_{i=1}^{N-1} \widehat{\sigma_i}^2}{n}\,, \tag{4.11a}$$

$$\lambda_{\mathbf{B}} = \min\left(\frac{\frac{N-2}{n} \cdot \sum_{i=1}^{N-1} \widehat{\sigma_i}^4 + \left[\sum_{i=1}^{N-1} \widehat{\sigma_i}^2\right]^2}{(N+2) \cdot \left[\sum_{i=1}^{N-1} \widehat{\sigma_i}^4 - \frac{\left[\sum_{i=1}^{N-1} \widehat{\sigma_i}^2\right]^2}{n}\right]},\, 1\right)\,. \tag{4.11b}$$

With $\varphi = \mu_{\widehat{\mathbf{B}}} \cdot \lambda_{\widehat{\mathbf{B}}}$ and $\delta = 1 - \lambda_{\widehat{\mathbf{B}}}$ the estimated covariance matrix (4.8) is

$$\widehat{\mathbf{B}} = \varphi \cdot \mathbf{I} + \delta \cdot \mathbf{S} \cdot \mathbf{S}^T \in \mathbb{R}^{n \times n} \,. \tag{4.12}$$

Clearly, $\widehat{\mathbf{B}}$ is a full rank estimator:

$$\begin{aligned} \widehat{\mathbf{B}} = \varphi \cdot \mathbf{I} + \delta \cdot \mathbf{S} \cdot \mathbf{S}^T &= \varphi \cdot \mathbf{I} + \mathbf{U}_{\mathbf{S}} \cdot \left[\delta \cdot \Sigma_{\mathbf{P}^b}\right] \cdot \mathbf{U}_{\mathbf{S}}^T \\ &= \mathbf{U}_{\mathbf{S}} \cdot \left[\varphi \cdot \mathbf{I} + \delta \cdot \mathbf{\Sigma}_{\mathbf{P}^b}\right] \cdot \mathbf{U}_{\mathbf{S}}^T, \end{aligned}$$

since $\varphi \cdot \mathbf{I} + \delta \cdot \mathbf{\Sigma}_{\mathbf{P}^b} \in \mathbb{R}^{n \times n}$ is always positive definite:

$$\begin{aligned} \det\left(\widehat{\mathbf{B}}\right) &= \det\left(\varphi \cdot \mathbf{I} + \delta \cdot \mathbf{\Sigma}_{\mathbf{P}^b} \in \mathbb{R}^{n \times n}\right) \\ &= \varphi^{n-N+1} \cdot \left[\prod_{i=1}^{N-1} \left(\delta \cdot \sigma_i^2 + \varphi\right)\right] > 0 \,. \end{aligned}$$

When the model is described in terms of multiple variables with (possibly) different magnitudes, the target matrix can be calculated making use of traces of sub-matrices. For instance, consider a model with two variables v_1 and v_2, grid resolution $[\sqrt{d} \times \sqrt{d}]$, and number of samples (ensemble members) equal to s. The estimated sample covariance matrix of background errors is:

$$\mathbf{P}^b_{u_1,u_2} = \begin{bmatrix} \mathbf{P}^b_{u_1} & \mathbf{S}_{u_1} \cdot \mathbf{S}^T_{u_2} \\ \mathbf{S}_{u_1} \cdot \mathbf{S}^T_{u_2} & \mathbf{P}^b_{u_2} \end{bmatrix} \in \mathbb{R}^{2d \times 2d} \tag{4.13}$$

where $\mathbf{S}_{u_1} \in \mathbb{R}^{d \times s}$ and $\mathbf{S}_{u_2} \in \mathbb{R}^{d \times s}$ are the matrix of member deviations for the variable u_1 and u_2, respectively. Likewise, $\mathbf{P}^b_{u_1} \in \mathbb{R}^{d \times d}$ and $\mathbf{P}^b_{u_2} \in \mathbb{R}^{d \times d}$ are the sample covariance matrices of background errors for the variables u_1 and u_2, respectively. The target matrix reads:

$$\mathbf{T}_{u_1,u_2} = \begin{bmatrix} \frac{\mathbf{tr}\left(\mathbf{P}^b_{u_1}\right)}{d} \cdot \mathbf{I}_{d \times d} & \mathbf{0}_{d \times d} \\ \mathbf{0}_{d \times d} & \frac{\mathbf{tr}\left(\mathbf{P}^b_{u_2}\right)}{d} \cdot \mathbf{I}_{d \times d} \end{bmatrix} \in \mathbb{R}^{2d \times 2d} \,.$$

4.2.2 EnKF implementations based on the RBLW estimator

By replacing the estimated error covariance matrix (4.12) by the ensemble covariance matrix in the EnKF formulation (2.6a), the EnKF analysis in matrix form becomes

$$\mathbf{X}^a = \mathbf{X}^b + \widehat{\mathbf{B}} \cdot \mathbf{H}^T \cdot \left[\mathbf{R} + \mathbf{H} \cdot \widehat{\mathbf{B}} \cdot \mathbf{H}^T\right]^{-1} \cdot \mathbf{\Delta} \in \mathbb{R}^{n \times N} \,, \tag{4.14}$$

where the matrix of innovations $\mathbf{\Delta} \in \mathbb{R}^{m \times N}$ reads:

$$\mathbf{\Delta} = \mathbf{y} \cdot \mathbf{1}_N^T + \mathbf{G} - \mathbf{H} \cdot \mathbf{X}^b \in \mathbb{R}^{m \times N}, \tag{4.15}$$

with the i-th column of $\mathbf{G} \in \mathbb{R}^{m \times N}$ being a sample from the data error distribution

$$\boldsymbol{\epsilon}^{[i]} \sim \mathcal{N}\left(\mathbf{0},\, \mathbf{R}\right), \text{ for } 1 \leq i \leq N.$$

We have

$$\begin{aligned} \mathbf{X}^a &= \mathbf{X}^b + \left(\varphi \cdot \mathbf{I} + \delta \cdot \mathbf{S} \cdot \mathbf{S}^T\right) \cdot \mathbf{H}^T \\ &\cdot \left(\mathbf{R} + \mathbf{H} \cdot \left(\varphi \cdot \mathbf{I} + \delta \cdot \mathbf{S} \cdot \mathbf{S}^T\right) \cdot \mathbf{H}^T\right)^{-1} \cdot \mathbf{D} \\ \mathbf{X}^a &= \mathbf{X}^b + \mathbf{E} \cdot \boldsymbol{\Pi} \cdot \mathbf{Z}_{\widehat{\mathbf{B}}} + \varphi \cdot \mathbf{H}^T \cdot \mathbf{Z}_{\widehat{\mathbf{B}}}, \end{aligned} \tag{4.16}$$

where

$$\mathbf{E} = \sqrt{\delta} \cdot \mathbf{S} \in \mathbb{R}^{n \times N}, \tag{4.17a}$$

$$\boldsymbol{\Pi} = \mathbf{H} \cdot \mathbf{E} \in \mathbb{R}^{m \times N}, \tag{4.17b}$$

and $\mathbf{Z}_{\widehat{\mathbf{B}}} \in \mathbb{R}^{m \times N}$ is given by the solution of the linear system

$$\begin{aligned} \left(\boldsymbol{\Gamma} + \boldsymbol{\Pi} \cdot \boldsymbol{\Pi}^T\right) \cdot \mathbf{Z}_{\mathbf{B}} &= \mathbf{D}, \\ \boldsymbol{\Gamma} &= \mathbf{R} + \varphi \cdot \mathbf{H} \cdot \mathbf{H}^T \in \mathbb{R}^{m \times m}. \end{aligned} \tag{4.17c}$$

When $\mathbf{H}$ possesses a simple structure (e.g., indexes to observed components from vector states) the matrix $\boldsymbol{\Gamma}$ also has a simple structure (since in practice $\mathbf{R}$ is block diagonal). The linear system (4.17c) can be efficiently solved via the iterative Sherman Morrison formula (ISFM) [NRSA14] with no more than $\mathcal{O}\left(N^2 \cdot m\right)$ computations. To summarize the above discussion, the implementation of the ensemble Kalman filter based on the RBWL estimator with the assimilation performed on the model space (EnKF-FS) consists of the following steps:

1. Estimate the matrix of member deviations $\mathbf{S}^b$ (4.7b) based on the samples $\mathbf{x}^{b[i]}$, for $1 \leq i \leq N$.
2. Compute the singular values of $\mathbf{S}^b$.
3. Calculate $\mu_{\widehat{\mathbf{B}}}$ and $\lambda_{\widehat{\mathbf{B}}}$ according to (4.11a) and (4.11b), respectively. Set $\varphi = \mu_{\widehat{\mathbf{B}}} \cdot \lambda_{\widehat{\mathbf{B}}}$ and $\delta = 1 - \lambda_{\widehat{\mathbf{B}}}$.
4. Compute the innovation matrix $\boldsymbol{\Delta}$ according to (4.15).
5. Compute the set of matrices (4.17).
6. Perform the assimilation (4.16).
7. Propagate the ensemble members

   $$\mathbf{x}^{b[i]}_{\text{next}} = \mathcal{M}_{t_{\text{current}} \rightarrow t_{\text{next}}}\left(\mathbf{x}^{a[i]}_{\text{current}}\right),$$

   until the next assimilation step, for $1 \leq i \leq N$.

Another efficient implementation of the filter based on the RBLW estimator can be obtained via the three-dimensional variational (3D-Var) cost function in the ensemble space. Intuitively, the analysis state $\overline{\mathbf{x}}^a \in \mathbb{R}^{n \times 1}$ lives in the space spanned by the ensemble members:

$$\overline{\mathbf{x}}^a \in \mathbf{span}\left\{\mathbf{x}^{b[1]},\, \mathbf{x}^{b[2]},\, \dots,\, \mathbf{x}^{b[N]}\right\}. \tag{4.18}$$

Any vector $\mathbf{x} \in \mathbb{R}^{n \times 1}$ in this space can be written as:

$$\mathbf{x} = \overline{\mathbf{x}}^b + \mathbf{U}^b \cdot \boldsymbol{\alpha}\,, \tag{4.19}$$

where the set of basis vectors $\mathbf{U}^b \in \mathbb{R}^{n \times N}$ is given by:

$$\mathbf{U}^b = \mathbf{X}^b - \overline{\mathbf{x}}^b \cdot \mathbf{1}_N^T\,,$$

and $\boldsymbol{\alpha} \in \mathbb{R}^{N \times 1}$ is the vector of weights in redundant coordinates. Recall the 3D-Var cost function:

$$\mathcal{J}^{\text{3D}}\left(\mathbf{x}\right) = \frac{1}{2} \cdot \left\|\mathbf{x} - \mathbf{x}^b\right\|^2_{\mathbf{B}^{-1}} + \frac{1}{2} \cdot \left\|\mathbf{y} - \mathbf{H} \cdot \mathbf{x}\right\|^2_{\mathbf{R}^{-1}}. \tag{4.20}$$

Replacing (4.19) in (4.20) leads to the 3D-Var formulation in the ensemble space:

$$\mathcal{J}^{\text{3D}}_{\text{ens}}\left(\boldsymbol{\alpha}\right) = \frac{1}{2} \cdot \left\|\mathbf{U}^b \cdot \boldsymbol{\alpha}\right\|^2_{\mathbf{B}^{-1}} + \frac{1}{2} \cdot \left\|\mathbf{d} - \mathbf{Q} \cdot \boldsymbol{\alpha}\right\|^2_{\mathbf{R}^{-1}}\,, \tag{4.21}$$

where $\mathbf{d} = \mathbf{y} - \mathbf{H} \cdot \mathbf{x} \in \mathbb{R}^{m \times N}$ is the innovation vector and $\mathbf{Q} = \mathbf{H} \cdot \mathbf{U}^b \in \mathbb{R}^{m \times N}$. This formula can be used to obtain the optimal value of $\boldsymbol{\alpha}$ for each ensemble member by maximizing its posterior probability for a given perturbed observation $\mathbf{y}^{s[i]} = \mathbf{y} + \boldsymbol{\epsilon}^{[i]}$, for $1 \le i \le N$. The 3D-Var cost function with the RBLW estimator reads:

$$\mathcal{J}^{\text{3D}}_{\text{RBLW}}\left(\boldsymbol{\lambda}\right) = \frac{1}{2} \cdot \left\|\mathbf{U}^b \cdot \boldsymbol{\lambda}\right\|^2_{\widehat{\mathbf{B}}^{-1}} + \frac{1}{2} \cdot \left\|\boldsymbol{\Delta} - \mathbf{Q} \cdot \boldsymbol{\lambda}\right\|^2_{\mathbf{R}^{-1}} \tag{4.22}$$

and the optimal set of weights $\boldsymbol{\lambda}^* \in \mathbb{R}^{N \times N}$ is obtained by minimizing the cost function (4.22):

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \mathcal{J}^{\text{3D}}_{\text{RBLW}}\left(\boldsymbol{\lambda}\right). \tag{4.23}$$

The resulting analysis ensemble reads:

$$\mathbf{X}^a = \mathbf{X}^b + \mathbf{U}^b \cdot \boldsymbol{\lambda}^* \in \mathbb{R}^{n \times N}.$$

When $\mathbf{H}$ is linear, the close form solution for $\boldsymbol{\lambda}^*$ can be written as follows:

$$\boldsymbol{\lambda}^* = \left[\mathbf{U}^{b^T} \cdot \mathbf{Z}_{\widehat{\mathbf{B}}\mathbf{U}} + \mathbf{Q}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{Q}\right]^{-1} \cdot \left[\mathbf{Q}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{D}\right]\,, \tag{4.24}$$

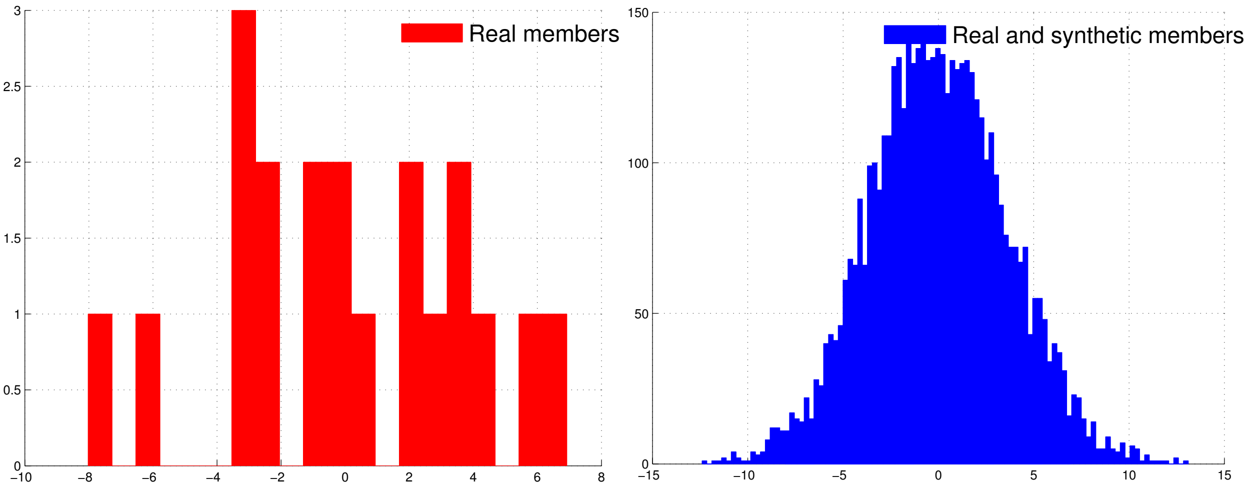
where $\mathbf{Z}_{\widehat{\mathbf{B}}\mathbf{U}} \in \mathbb{R}^{n \times N}$ is the solution of the linear system $\widehat{\mathbf{B}} \cdot \mathbf{Z}_{\widehat{\mathbf{B}}\mathbf{U}} = \mathbf{U}^b$ which can be efficiently solved, again, by making use of the ISMF in no more than $\mathcal{O}\left(n \cdot N^2\right)$ long computations.

Typically, in reduced space approximations, the quality of the analysis depends on the number of directions (degrees of freedom) in which the analysis state is sought. From a statistical point of view, a large number of degrees of freedom provides a better representation of the background error distribution, and therefore the impact of sampling errors is reduced. Each degree of freedom added to (4.18) depends on a full model propagation. In practice, the number of model realizations is small given the computational effort to propagate them in time. Since we have a well-conditioned estimator of the background error covariance matrix, a better representation of the background uncertainty can be obtained by taking K additional samples from the estimated normal distribution:

$$\widetilde{\mathbf{x}}_i^b \sim \mathcal{N}\left(\overline{\mathbf{x}}^b,\, \widehat{\mathbf{B}}\right) \in \mathbb{R}^{n\times 1}\,, \quad 1 \le i \le K. \tag{4.25}$$

This yields a new ensemble formed of two kinds of members, real and synthetic. The *real members* $\{\mathbf{x}_i^b\}_{i=1}^N$ are obtained by model propagation of the previous analysis ensemble. The *synthetic members* $\{\widetilde{\mathbf{x}}_i^b\}_{i=1}^K$ are syntheticly built by taking samples from the distribution (4.25) and do not require additional model runs.

The synthetic increase in the size of the ensemble is therefore a relatively inexpensive modality to bring in additional degrees of freedom in the solution of the optimization problem (4.22). Figure 4.1 exemplifies the effect of additional members using two-dimensional projections of ensemble member states from the Lorenz-96 model. Figure 4.1a shows the approximated background error distribution for the 5-th model component based on 20 real members. Figure 4.1b shows the distribution when synthetic members are added to the background ensemble, resulting in a better representation of the background error and therefore a decrease in the sampling error.



(a) $K = 0$

(b) $K = 120$

Figure 4.1: Error distribution for different values of K using histograms for the 5-th model component of ensemble members from the Lorenz-96 model.

The analysis state is now computed in the subspace spanned by both real and synthetic members:

$$\overline{\mathbf{x}}^a \in \textbf{range}\left\{\widetilde{\mathbf{X}}^b\right\}, \tag{4.26}$$

where the *extended ensemble* reads:

$$\widetilde{\mathbf{X}}^b = \left[\mathbf{x}^{b[1]},\, \dots,\, \mathbf{x}^{b[N]},\, \widetilde{\mathbf{x}}^{b[1]},\, \dots,\, \widetilde{\mathbf{x}}^{b[K]}\right] \in \mathbb{R}^{n \times K}\,.$$

with $K = N + K$. A new set of basis vectors is built as follows:

$$\widetilde{\mathbf{U}} = \left[\mathbf{x}^{b[1]} - \overline{\mathbf{x}}^b,\, \dots,\, \mathbf{x}^{b[N]} - \overline{\mathbf{x}}^b, \widetilde{\mathbf{x}}^{b[1]} - \overline{\mathbf{x}}^b,\, \dots,\, \widetilde{\mathbf{x}}^{b[K]} - \overline{\mathbf{x}}^b\right] \in \mathbb{R}^{n \times K}\,,$$

where $\overline{\mathbf{x}}^b$ is the background ensemble mean. Similar to (4.22), the 3D-Var cost function based on the RBLW estimator with real and synthetic members reads:

$$\widetilde{\mathcal{J}}^{\text{3D}}_{\text{ens}}\left(\widetilde{\boldsymbol{\lambda}}\right) = \frac{1}{2} \cdot \left\|\widetilde{\mathbf{U}} \cdot \widetilde{\boldsymbol{\lambda}}\right\|^2_{\widehat{\mathbf{B}}^{-1}} + \frac{1}{2} \cdot \left\|\mathbf{D} - \widetilde{\mathbf{Q}} \cdot \widetilde{\boldsymbol{\lambda}}\right\|^2_{\mathbf{R}^{-1}}\,, \tag{4.27}$$

where $\widetilde{\boldsymbol{\lambda}} \in \mathbb{R}^{K \times N}$ is the matrix of weights whose i-th column represents the coordinates of the i-th ensemble member in the space (4.26), for $1 \leq i \leq K$, and $\widetilde{\mathbf{Q}} = \mathbf{H} \cdot \widetilde{\mathbf{U}} \in \mathbb{R}^{m \times K}$. The resulting 3D-Var optimization problem

$$\widetilde{\boldsymbol{\lambda}}^* = \underset{\widetilde{\boldsymbol{\lambda}}}{\arg\min}\ \mathcal{J}_{\text{ens}}\left(\widetilde{\boldsymbol{\lambda}}\right)\,, \tag{4.28}$$

has the solution

$$\widetilde{\boldsymbol{\lambda}}^* = \left[\widetilde{\mathbf{U}}^T \cdot \mathbf{Z}_{\widehat{\mathbf{B}}\widetilde{\mathbf{U}}} + \widetilde{\mathbf{Q}}^T \cdot \mathbf{R}^{-1} \cdot \widetilde{\mathbf{Q}}\right]^{-1} \cdot \widetilde{\mathbf{Q}}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{D} \in \mathbb{R}^{K \times N}\,, \tag{4.29}$$

where $\mathbf{Z}_{\widehat{\mathbf{B}}\mathbf{U}} = \widehat{\mathbf{B}}^{-1} \cdot \widetilde{\mathbf{U}} \in \mathbb{R}^{n \times K}$. The resulting analysis ensemble is

$$\mathbf{X}^a = \mathbf{X}^b + \widetilde{\mathbf{U}} \cdot \widetilde{\boldsymbol{\lambda}}^* \in \mathbb{R}^{n \times N}\,. \tag{4.30}$$

To summarize, the implementation of the EnKF based on the RBLW estimator with the assimilation performed in the reduced space (4.26) (EnKF-RS) consists of the following steps:

1. Estimate the matrix of member deviations $\mathbf{S}^b$ (4.7b) based on the samples $\mathbf{x}^{b[i]}$, for $1 \leq i \leq N$.
2. Compute the singular values of $\mathbf{S}^b$.
3. Calculate $\mu_{\widehat{\mathbf{B}}}$ and $\lambda_{\widehat{\mathbf{B}}}$ according to (4.11a) and (4.11b), respectively. Set $\varphi = \mu_{\widehat{\mathbf{B}}} \cdot \lambda_{\widehat{\mathbf{B}}}$ and $\delta = 1 - \lambda_{\widehat{\mathbf{B}}}$.

4. Compute the innovation matrix $\mathbf{\Delta}$ according to (4.15).
5. Draw K synthetic members according to (4.25).
6. Compute the matrix of optimal weights (4.29).
7. Perform the assimilation (4.30).
8. Remove the synthetic members.
9. Propagate the physical ensemble members to the next assimilation step:

$$\mathbf{x}^{b[i]}_{\text{next}} = \mathcal{M}_{t_{\text{current}} \rightarrow t_{\text{next}}} \left(\mathbf{x}^{a[i]}_{\text{current}} \right) , \quad 1 \le i \le N.$$

Even if the synthetic members (4.25) are used in the covariance approximation for the analysis step, according to (4.29) only the states of real members are adjusted. After the assimilation step the synthetic members are discarded and only the real members form the analysis ensemble. This strategy does not increase the number of ensemble members to be propagated by model runs (and therefore, the computational effort). When more computational resources become available, or when some real members are lost due to hardware failures, selected synthetic members can be updated and propagated as well. In this case they become real members during the next assimilation step. Moreover, some real members can be replaced by synthetic members in order to refresh the ensemble directions, e.g., such as to prevent filter divergence. This can be done as well by replacing analysis members by samples from the distribution:

$$\mathbf{x} \sim \mathcal{N} \left(\overline{\mathbf{x}}^a,\, \widehat{\mathbf{A}} \right) , \tag{4.31}$$

where $\overline{\mathbf{x}}^a \in \mathbb{R}^{n \times 1}$ is the empirical mean of the analysis ensemble and $\widehat{\mathbf{A}} \in \mathbb{R}^{n \times n}$ is the RBLW estimator of the posterior ensemble. Other methodologies based on adding synthetic members in order to mitigate the impact of sampling errors can be found in the literature. One of them is the Adaptative ensemble Kalman filter (AEnKF) [SHCS10] in which, after each analysis step, a new member is added to the ensemble:

$$\mathbf{x}^e = \overline{\mathbf{x}}^a + \zeta \cdot \boldsymbol{\delta}\mathbf{x}^* \in \mathbb{R}^n,$$

where $\boldsymbol{\delta}\mathbf{x}^* \in \mathbb{R}^{n \times 1}$ is the optimal increment for $\mathbf{x} = \overline{\mathbf{x}}^b + \boldsymbol{\delta}\mathbf{x}$ in (4.20) and can be written as follows [Lor86]:

$$\boldsymbol{\delta}\mathbf{x}^* = \mathbf{C}^{\mathbf{B}} \cdot \mathbf{H}^T \cdot \left[\mathbf{R} \cdot \mathbf{H} \cdot \mathbf{C}^{\mathbf{B}} \cdot \mathbf{H}^T \right] \cdot \mathbf{z},$$

where $\mathbf{z} = \mathbf{y} - \mathbf{H} \cdot \overline{\mathbf{x}}^a \in \mathbb{R}^{m \times 1}$, $\mathbf{C}^{\mathbf{B}} \in \mathbb{R}^{n \times n}$ is an approximation of the background error covariance matrix $\mathbf{B}$, and $\zeta \in \mathbb{R}$ is a tunable parameter: larger values of ζ shift the mean of the new ensemble toward a state that fits the observations. Alternatively, a set of new members can be added from partial solutions of the 3D-Var optimization problem, i.e., first conjugate directions. Samples can be taken from the posterior distribution as well by making use of a pseudo-square root of the posterior sample covariance matrix. Then, a second order sampling procedure is used in order to build the synthetic members. The new directions are propagated making use of the fully non-linear model and therefore, in order to avoid increments in the computational effort of AEnKF, some real members must be replaced prior the model propagation. There are some clear differences between the EnKF-RS and AEnKF formulations: AEnKF samples the analysis distribution in order to refresh the set of directions of the ensemble and converge to directions where the states best fit observations; EnKF-RS samples the background distribution in order to increase the degrees of freedom of the ensemble space and therefore, to mitigate the effects of under-sampling during the assimilation process. EnKF-RS can also sample the analysis in order to refresh the set of directions, however, this is not the main purpose of sampling in the EnKF-RS formulation. Moreover, the sampling procedure in EnKF-RS is based on the well-conditioned background error covariance matrix estimator [LW04], while the approximation used in the AEnKF context is a pseudo-square root rank-deficient approximation. Unlike AEnKF, there is no need to tune parameters in the EnKF-RS formulation, a highly desirable feature for real applications.

4.2.3 Sampling in high-dimensions based on the RBLW estimator

Both implementations discussed in Section 4.2.2 use samples from the distribution (4.25). Such samples can be generated as follows:

$$\widetilde{\mathbf{x}}_i^b = \overline{\mathbf{x}}^b + \widehat{\mathbf{B}}^{1/2} \cdot \boldsymbol{\nu}_i = \overline{\mathbf{x}}^b + \left(\varphi \cdot \mathbf{I} + \delta \cdot \mathbf{S}^b \cdot {\mathbf{S}^b}^T\right)^{1/2} \cdot \boldsymbol{\nu}_i \tag{4.32}$$

for $1 \le i \le K$, where $\boldsymbol{\nu}_i \sim \mathcal{N}(\mathbf{0},\, \mathbf{I})$. However, this computation requires the explicit representation in memory of the estimated error covariance matrix $\widehat{\mathbf{B}}$, which is prohibitive for high-resolution models. Moreover, the square root matrix $\widehat{\mathbf{B}}^{1/2}$ is required making the use of (4.32) impractical.

We need an equivalent strategy to obtain the samples (4.25) that requires a reasonable computational effort and does not use a full representation of the covariance matrix $\widehat{\mathbf{B}}$. Toward this end consider the random vectors

$$\begin{aligned} \boldsymbol{\nu}_i^1 &\sim \mathcal{N}(\mathbf{0},\, \mathbf{I}) \in \mathbb{R}^{n \times 1}\,, \\ \boldsymbol{\nu}_i^2 &\sim \mathcal{N}(\mathbf{0},\, \mathbf{I}) \in \mathbb{R}^{N \times 1}\,, \end{aligned}$$

and let

$$\begin{aligned} \mathrm{Cov}\left(\boldsymbol{\nu}_i^1,\, \boldsymbol{\nu}_i^2\right) &= \boldsymbol{\nu}_i^1 \cdot {\boldsymbol{\nu}_i^2}^T = \mathbf{0}\,, \\ \mathrm{Cov}\left(\boldsymbol{\nu}_i^2,\, \boldsymbol{\nu}_i^1\right) &= \boldsymbol{\nu}_i^2 \cdot {\boldsymbol{\nu}_i^1}^T = \mathbf{0}\,. \end{aligned}$$

We make the following substitution in (4.32)

$$\widehat{\mathbf{B}}^{1/2} \cdot \boldsymbol{\nu}_i \sim \sqrt{\varphi} \cdot \boldsymbol{\nu}_i^1 + \sqrt{\delta} \cdot \mathbf{S} \cdot \boldsymbol{\nu}_i^2 .$$

This does not change the statistics since

$$\begin{aligned}
&\mathbb{E}\left[\left(\sqrt{\varphi} \cdot \mathbf{I} \cdot \boldsymbol{\nu}_i^1 + \sqrt{\delta} \cdot \mathbf{S}^b \cdot \boldsymbol{\nu}_i^2\right) \cdot \left(\sqrt{\varphi} \cdot \mathbf{I} \cdot \boldsymbol{\nu}_i^1 + \sqrt{\delta} \cdot \mathbf{S}^b \cdot \boldsymbol{\nu}_i^2\right)^T\right] \\
&= \varphi \cdot \underbrace{\boldsymbol{\nu}_i^1 \cdot \boldsymbol{\nu}_i^{1T}}_{\mathrm{Cov}\left(\boldsymbol{\nu}_i^1, \boldsymbol{\nu}_i^1\right) = \mathbf{I}} + \sqrt{\varphi \cdot \delta} \cdot \underbrace{\boldsymbol{\nu}_i^1 \cdot \boldsymbol{\nu}_i^{2T}}_{\mathrm{Cov}\left(\boldsymbol{\nu}_i^1, \boldsymbol{\nu}_i^2\right) = \mathbf{0}} + \sqrt{\varphi \cdot \delta} \cdot \underbrace{\boldsymbol{\nu}_i^2 \cdot \boldsymbol{\nu}_i^{1T}}_{\mathrm{Cov}\left(\boldsymbol{\nu}_i^2, \boldsymbol{\nu}_i^1\right) = \mathbf{0}} \\
&\quad + \delta \cdot \mathbf{S}^b \cdot \underbrace{\boldsymbol{\nu}_i^2 \cdot \boldsymbol{\nu}_i^{2T}}_{\mathrm{Cov}\left(\boldsymbol{\nu}_i^2, \boldsymbol{\nu}_i^2\right) = \mathbf{I}} \cdot \mathbf{S}^{bT} = \varphi \cdot \mathbf{I} + \delta \cdot \mathbf{S}^b \cdot \mathbf{S}^{bT} \\
&= \widehat{\mathbf{B}} .
\end{aligned}$$

The synthetic ensemble members are obtained as follows:

$$\widetilde{\mathbf{x}}_i^b = \overline{\mathbf{x}}^b + \sqrt{\varphi} \cdot \mathbf{I} \cdot \boldsymbol{\nu}_i^1 + \sqrt{\delta} \cdot \mathbf{S} \cdot \boldsymbol{\nu}_i^2, \quad i = 1, \ldots, K. \tag{4.33}$$

The components of the random variables $\boldsymbol{\nu}_i^1$ and $\boldsymbol{\nu}_i^2$ are drawn independently from the standard normal distribution $\mathcal{N}(0,\, 1)$. For large model resolutions the components of the random vectors can be prepared independently taking advantage of parallel computations. Moreover, the random vectors $\boldsymbol{\nu}_i^1$ and $\boldsymbol{\nu}_i^2$ can be sampled prior the assimilation process in an off-line computation.

The estimated error covariance matrix is never represented explicitly in memory. Instead, the estimator $\widehat{\mathbf{B}}$ is represented via the triplet

$$\widehat{\mathbf{B}} \equiv [\varphi,\, \mu,\, \mathbf{S}] \, .$$

which contains two scalars and one matrix of dimension $n \times N$. In addition, the scalars φ and μ are computed making use only of the matrix $\mathbf{S}$. This data is sufficient for sampling from the distribution (4.25).

4.2.4 Covariance inflation, localization and four dimensional formulation

There are techniques from the current literature that can be easily incorporated into our proposed implementations of the EnKF in order to provide better analysis corrections during the assimilation process. In this section we address covariance inflation, localization, and extension to four-dimensional variational (4D-Var) framework.

Localization

Localization methods can be used in the context of the EnKF-FS and EnKF-RS implementations in order to mitigate the impact of spurious correlations. Direct covariance matrix localization is not possible in our context since it require the explicit computation of $\widehat{\mathbf{B}}$, but one can still performing local analyses. Similar to the EnKF formulations [OHS$^+$04b, Kep00], we choose a sub-domain of radius ζ around each model component and then, making use of the surrounding components, local background errors covariance matrices $\widehat{\mathbf{B}}_\zeta \in \mathbb{R}^{n_\zeta \times n_\zeta}$ are estimated, where n_ζ is the number of components in the local domain. The assimilation process is carried out using only the observed components within the radius of influence ζ. An attractive feature of local analysis is that, typically, the sub-domain dimension is smaller than the ensemble size, and the local filter effectively uses $N > n$. This situation allows for new target matrices $\mathbf{T}_\zeta \in \mathbb{R}^{n_r \times n_\zeta}$ in the formulation of $\widehat{\mathbf{B}}_\zeta$. The structure of $\mathbf{T}_\zeta$ can be based on our knowledge about how the dynamics of the numerical model drive the background error correlations. Note that matrix computations in the sub-domain are inexpensive, and therefore other shrinkage covariance estimators such as those presented in Section (4.1) can be used as well.

Covariance inflation

While localization methods reduce the impact of spurious correlations, covariance inflation mitigates the impact of under-estimation of sample variances [AA99]. The ensemble members are inflated about the ensemble mean $\overline{\mathbf{x}}^b$ by a factor of $\rho \in \mathbb{R}$, e.g., before the assimilation process:

$$\mathbf{X}^{b[inf]} = \overline{\mathbf{x}}^b \cdot \mathbf{1}_N^T + \rho \cdot \left[\mathbf{X}^b - \overline{\mathbf{x}}^b \cdot \mathbf{1}_N^T\right] \in \mathbb{R}^{n \times N} \,.$$

It can be seen that inflating each deviation by a factor of ρ has the following effect on $\widehat{\mathbf{B}}$:

$$\widehat{\mathbf{B}}^{[\mathrm{inf}]} = \varphi \cdot \mathbf{I} + \left[\delta \cdot \rho^2\right] \cdot \mathbf{S} \cdot \mathbf{S}^T \,.$$

When the inflation is performed after the assimilation process the current $\widehat{\mathbf{B}}$ does not change.

Four dimensional EnKF-RS

When observations are available at multiple times, $\{\mathbf{y}_k\}_{k=0}^{M}$ where $M+1$ is the total number of observation times, the EnKF-RS can be easily extended to incorporate all of them at once into the assimilation process. This is done via the four dimensional variational (4D-Var) cost function [Lor86] in the ensemble space:

$$\mathcal{J}_{\mathrm{ens}}^{\mathrm{4D}}(\boldsymbol{\beta}) = \frac{1}{2} \cdot \left\|\mathbf{U}_0 \cdot \boldsymbol{\beta}\right\|^2_{\widehat{\mathbf{B}}_0^{-1}} + \frac{1}{2} \cdot \sum_{k=0}^{M} \left\|\boldsymbol{\Delta}_k - \mathbf{Q}_k \cdot \boldsymbol{\beta}\right\|^2_{\mathbf{R}_k^{-1}} \,, \tag{4.34}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{N \times N}$ is the matrix of weights in redundant coordinates that need to be determined. At time t_k, $0 \le k \le M$, $\mathbf{U}_k = \mathbf{X}_k^b - \overline{\mathbf{x}}_k^b \cdot \mathbf{1}_N^T \in \mathbb{R}^{n \times N}$ is the set of basis vectors, $\mathbf{X}_k^b \in \mathbb{R}^{n \times N}$ is the ensemble of model runs, $\overline{\mathbf{x}}_k^b \in \mathbb{R}^{n \times 1}$ is the ensemble mean, $\mathbf{Q}_k = \mathbf{H}_k \cdot \mathbf{U}_k \in \mathbb{R}^{m \times N}$, $\mathbf{H}_k$ is the observational operator, $\boldsymbol{\Delta}_k \in \mathbb{R}^{m \times N}$ is the innovation matrix:

$$\mathbf{D}_k = \mathbf{Y}_k^s - \mathbf{H}_k \cdot \mathbf{X}_k^b \in \mathbb{R}^{m \times N}\,,$$

whose columns include perturbed observations:

$$\mathbf{y}_k^{s[i]} \sim \mathcal{N}\left(\mathbf{y}_k,\, \mathbf{R}_k\right)\,, \quad 1 \le i \le N,$$

and $\mathbf{R}_k \in \mathbb{R}^{m \times m}$ is the data error covariance matrix. The optimal matrix of weights $\boldsymbol{\beta}^* \in \mathbb{R}^{N \times N}$ is obtained via the minimization of the cost function (4.34):

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \mathcal{J}_{\text{ens}}^{\text{4D}}\left(\boldsymbol{\beta}\right)\,. \tag{4.35}$$

When $\{\mathbf{H}_k\}_{k=0}^M$ are linear operators, the solution of (4.35) reads:

$$\boldsymbol{\beta}^* = \left[\mathbf{U}_0^T \cdot \widehat{\mathbf{B}}_0^{-1} \cdot \mathbf{U}_0 + \sum_{k=0}^{M} \mathbf{Q}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{Q}_k\right]^{-1} \cdot \left[\sum_{k=0}^{M} \mathbf{Q}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{D}_k\right]\,. \tag{4.36}$$

The 4D-EnKF-RS analysis state at time t_k reads:

$$\mathbf{X}_k^a = \mathbf{X}_k^b + \mathbf{U}_k \cdot \boldsymbol{\beta}^*\,, \quad 0 \le k \le M. \tag{4.37}$$

Since (4.37) represents an approximated solution, after the initial ensemble mean $\overline{\mathbf{x}}_0^a$ is recovered it is propagated forward in time to obtain an approximation of the optimal trajectory:

$$\overline{\mathbf{x}}_k^a = \mathcal{M}_{t_{k-1} \to t_k}\left(\overline{\mathbf{x}}_{k-1}^a\right)\,, \text{ for } 1 \le k \le M\,.$$

A very attractive property of this extension is that it is adjoint-free. Moreover, the computational effort of the assimilation process is $\mathcal{O}\left(n \cdot N^2\right)$ computations which is equivalent to the computational cost of EnKF for a single assimilation step. Note that, the ideas of local analyses can be easily extended to this 4D-EnKF-RS formulation. Local observations belonging to different observation times are assimilated via (4.36). Similar to the sequential case, all the local analysis components are mapped to the global domain in order to build the global analysis solution. Similarly, EnKF-FS can be extended to incorporate observations at different times, however, the adjoint model is needed.

4.2.5 Comparison of EnKF-FS and EnKF-RS versions of the filter

Although both EnKF-FS and EnKF-RS methods are based on the EnKF equations and RBLW estimator, their underlying theoretical properties are slightly different. To facilitate the comparison of the two proposed implementations we bring the EnKF-FS analysis equation (4.14) to the form (4.30):

$$\mathbf{X}^a = \mathbf{X}^b + \widehat{\mathbf{B}}^{1/2} \cdot \boldsymbol{\alpha}_{\widehat{\mathbf{B}}} \in \mathbb{R}^{n \times N}\,, \tag{4.38}$$

where the weights $\boldsymbol{\alpha}_{\widehat{\mathbf{B}}} \in \mathbb{R}^{n \times N}$ are given by

$$\boldsymbol{\alpha}_{\widehat{\mathbf{B}}} = \widehat{\mathbf{B}}^{1/2} \cdot \mathbf{H}^T \left[\mathbf{R} + \mathbf{H} \cdot \widehat{\mathbf{B}} \cdot \mathbf{H}^T\right]^{-1} \cdot \boldsymbol{\Delta} \in \mathbb{R}^{n \times N}\,.$$

It is readily apparent from equations (4.38) and (4.30) that EnKF-FS and EnKF-RS implementations differ in the number of degrees of freedom used in the assimilation process. In the EnKF-FS approach the columns of $\widehat{\mathbf{B}}^{1/2}$ serve as the basis set for generating an ensemble of background deviations. Since the estimated background error covariance matrix is full-rank the optimal solution is searched in the full space:

$$\overline{\mathbf{x}}^a \in \mathbf{span}\left\{\mathbf{b}^{[1]}, \mathbf{b}^{[2]}, \ldots, \mathbf{b}^{[n]}\right\} \tag{4.39}$$

where $\mathbf{b}^{[i]} \in \mathbb{R}^{n \times 1}$, for $1 \leq i \leq n$, is the i-th column of $\widehat{\mathbf{B}}^{1/2}$. For this reason, the use of synthetic members in the EnFK-FS formulation can be avoided.

The matrix identity

$$\left[\widehat{\mathbf{B}}^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H}\right]^{-1} \cdot \mathbf{H}^T \cdot \mathbf{R}^{-1} \equiv \widehat{\mathbf{B}} \cdot \mathbf{H}^T \cdot \left[\mathbf{R} + \mathbf{H} \cdot \widehat{\mathbf{B}} \cdot \mathbf{H}^T\right]^{-1}$$

together with (4.38) reveal that the weighted covariance matrix of the EnKF-FS implementation

$$\mathbf{W} = \widehat{\mathbf{B}}^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H} \in \mathbb{R}^{n \times n}\,, \tag{4.40}$$

is related to the weighted covariance matrix of the EnKF-RS method by the relation

$$\mathbf{W}_{\text{ens}} = \widetilde{\mathbf{U}}^T \cdot \mathbf{W} \cdot \widetilde{\mathbf{U}} \in \mathbb{R}^{K \times K}\,. \tag{4.41}$$

Therefore when the size of the ensemble K is increased (by adding real or synthetic members) more information from the matrix $\mathbf{W}$ is captured by its projection onto the K-dimensional space. Note that when $K \to n$ and $\widetilde{\mathbf{U}}$ is orthonormal we have that $\mathbf{W}_{\text{ens}} \to \mathbf{W}$. Consequently, the number of synthetic members will play an important role in the performance of the EnKF-RS implementation.

4.3 Parallel Implementation

Following the idea of the LETKF, we can make use of local assimilation methods in order to perform the assimilation process in parallel. Basically, we rely on the assumption that the observational operator can be locally approximated, of course when this is not the case, other alternatives should be sought. The parallel computations can be performed as follows:

1. Denote the number of local model components by
$$n_{\text{local}} = \frac{n}{z} \tag{4.42}$$
where z is the number of available processors. Boundary information for each sub-domain is collected for the assimilation process.

2. For all sub-domains, compute the local innovations $\mathbf{\Delta}_{\text{local}} \in \mathbb{R}^{m_{\text{local}} \times N}$, where m_{local} is the number of local observed components.

3. For all sub-domains, estimate the local background error covariance matrix $\widehat{\mathbf{B}}_{\text{local}} \in \mathbb{R}^{n_{\text{local}} \times n_{\text{local}}}$.

4. Perform the local assimilations:
$$\mathbf{X}^a{}_{\text{local}} = \mathbf{X}^b{}_{\text{local}} + \widehat{\mathbf{B}}_{\text{local}} \cdot \left[\mathbf{R}_{\text{local}} + \mathbf{H}_{\text{local}}^T \cdot \widehat{\mathbf{B}}_{\text{local}} \cdot \mathbf{H}_{\text{local}}\right]^{-1} \cdot \mathbf{\Delta}_{\text{local}}$$

5. Map the local analysis sub-domains to the global domain.

Note that, this approximation can be used when, for instance, block of correlations in **R** are of small size. We name this parallel approximation the parallel ensemble Kalman filter based on shrinkage covariance matrix estimation (EnKF-SC) Note that, the general idea is based on the EnKF-FS formulation. A similar approach can be proposed for the EnKF-RS.

4.4 Experimental Results

In this section, we study the performance of the proposed EnKF implementations. Two set of experiments are performed in order to assess the accuracy of the EnKF-FS and EnKF-RS methods. The first set of experiments address the impact of synthetic members on the accuracy of the EnKF-RS. In this initial set, the Lorenz-96 model is used as the forward model operator and a full observational network is considered with perfect observational operator. The last set of experiments address the impact of $\widehat{\mathbf{B}}$ in the accuracy of the proposed methods. In this case, the experiments are performed making use of a quasi-geostrophic model and, three sparse observational networks are used with, again, perfect observational operators. In general, the misestimation of background error correlations under sparse observational networks yields to improper innovations on (distant) unobserved model components. Thus, since no representativeness errors have place under both experimental settings, the quality of the analysis corrections will depend on how well **B** is estimated.

The root mean square error (RMSE) is used
$$\text{RMSE} = \left[\frac{1}{M} \cdot \sum_{k=1}^{M} \left\|\mathbf{x}_k^a - \mathbf{x}_k^{\text{ref}}\right\|^2\right]^{1/2},$$

as a metric of performance in terms of accuracy, where $\mathbf{x}_k^a$ and $\mathbf{x}_k^{\text{ref}}$ are the analysis and reference solution at time t_k, respectively. The accuracy of the proposed methods is contrasted against that for the LETK implementation. In all the experiments, the covariance inflation factor is 1.04.

4.4.1 The impact of synthetic members in the EnKF-RS formulation

The Lorenz-96 model [Lor05] is described by the following set of ordinary differential equations:

$$\frac{dx_j}{dt} = \begin{cases} (x_2 - x_{n-1}) \cdot x_n - x_1 + F & \text{for } j = 1 \\ (x_{j+1} - x_{j-2}) \cdot x_{j-1} - x_j + F & \text{for } 2 \leq j \leq n-1 \\ (x_1 - x_{n-2}) \cdot x_{n-1} - x_n + F & \text{for } j = n \end{cases} \tag{4.43}$$

where x_j, for $1 \leq j \leq n$, represent the spatial coordinates. This model has been heuristically formulated in order to account for properties of global atmospheric models such as the advection, dissipation and forcing. The model (4.43) exhibits extended chaos when the external force F is 8. For this reason, the model is adequate to perform basic studies of predictability. We follow the experimental settings of Hunt et al in [OHS$^+$04b]. One time unit (0.01) in the Lorenz-96 model represent one hour in the atmosphere. The number of components in the vector state n is 40. All model components are observed $m = n$. Observational errors are uncorrelated with equal variances of 0.01^2. Observations are taken every 200 time units ($\sim$ 8 days) during a time period of 200 days. No representativeness errors are present in the preparation of the observations and therefore, we expect the misspecification of background error correlations to be the main source of error in the computation of the analysis innovations (and therefore, analysis states) Details regarding the construction of the reference solution, background state and initial background ensemble member are provided next:

1. The reference solution is computed numerically, this is, at some time t_{-3} a random initial condition is built and then propagated forward in time from t_{-3} to t_{-2} making use of the ode45 method. The final state in the numerical integration provides the initial reference solution $\mathbf{x}_{-2}^{ref}$.

2. The background state is computed as follows: at time t_{-2}, the perturbed background solution $\widehat{\mathbf{x}^{ref}}_{-2} \in \mathbb{R}^{n\times 1}$ is built:

$$\widehat{\mathbf{x}^{ref}}_{-2} = \mathbf{x}_{-2}^{ref} + \pi^b \circ \mathbf{x}_{-2}^{ref} \in \mathbb{R}^{n\times 1}, \text{ with } \pi^b \sim \mathcal{N}\left(\mathbf{0}_n,\, 0.15 \cdot \mathbf{I}\right),$$

   and then, it is propagated forward in time from t_{-2} to t_{-1} from which the background state $\mathbf{x}_{-1}^b \in \mathbb{R}^{n\times 1}$ is obtained. This model propagation attenuates the random noise added to $\widehat{\mathbf{x}^{ref}}_{-2}$ and makes the background state $\mathbf{x}^b_{-1}$ consistent with the physics of the model (4.43).

3. The initial ensemble at time t_0 is built by forward propagations of perturbed background states at time t_{-1}: for each ensemble member a perturbed background state is built:

   $$\widehat{\mathbf{x}^{b[i]}}_{-1} = \mathbf{x}^b_{-1} + \pi^b \circ \mathbf{x}^b_{-1} \in \mathbb{R}^{n\times 1}, \text{ with } \pi^b \sim \mathcal{N}\left(\mathbf{0}_n,\, 0.15 \cdot \mathbf{I}\right) ,$$

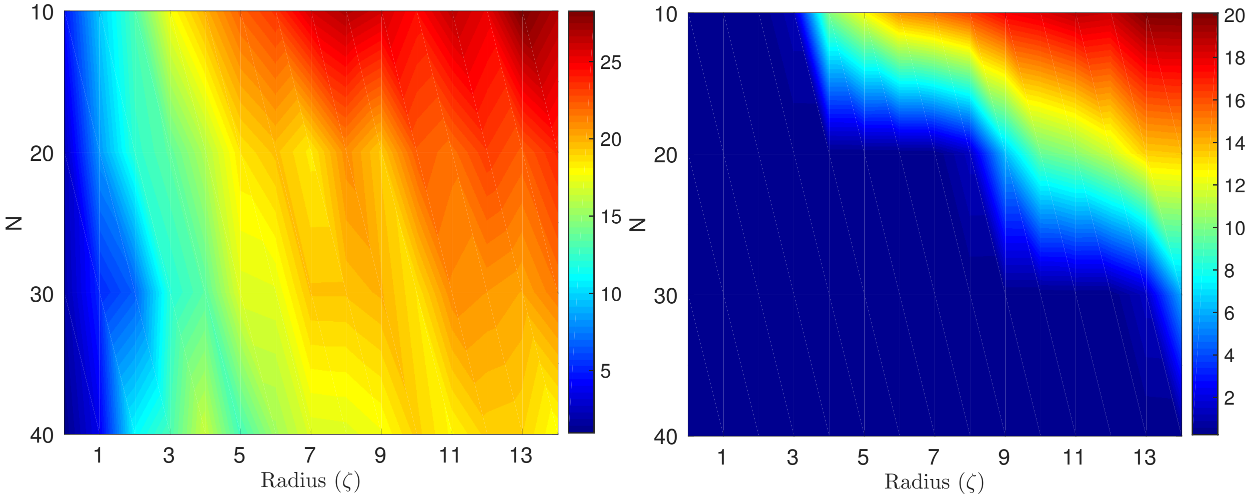
   for $1 \le i \le N$, and then it is propagated forward in time until time t_0:

   $$\mathbf{x}_0^{b[i]} = \mathcal{M}_{t_{-1}\rightarrow t_0}\left(\widehat{\mathbf{x}^{b[i]}}_{-1}\right) ,$$

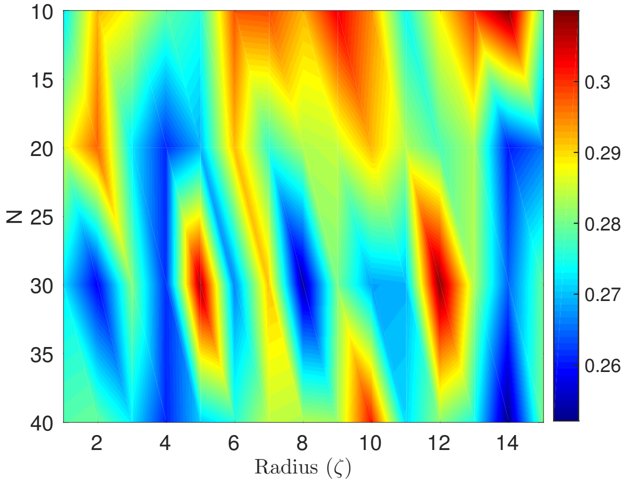
   again, this model propagation ensures that the ensemble members are consistent with the physics of the model (4.43).

4. The reference and background solutions $\mathbf{x}^{ref}_{-2}$ and $\mathbf{x}^b_{-1}$, respectively, are propagated forward in time until t_0 from which the initial reference solution $\mathbf{x}_0^{ref}$ and initial background state $\mathbf{x}_0^b$ are obtained.

Initially, the EnKF-FS, the EnKF-RS and the LETKF implementations are tested under different ensemble sizes N and radii of influence ζ. The ensemble sizes ranges in [10, 40] while r does that in [1, 15]. For each pair, $(N,\, r)$ the same initial ensemble and observations are used across all the EnKF implementations. The number of synthetic members for EnKF-RS is zero ($K = 0$). The results are shown in the figure 4.2. Figure 4.2a shows that, for the LETKF implementation, the quality of the analysis innovations are strictly related to the radius sizes. Even for large ensemble sizes, the analysis means poorly estimates the reference solutions. Mainly, this occurs because the analysis innovations are considerably affected by spurious correlations when the radius of influence is large (i.e., 13). This is not surprising since the background error covariance matrix estimator in the context of LETKF is the ensemble covariance matrix $\mathbf{P}^b$. In figure 4.2b, the RMSE values for the EnKF-RS increases toward zones where N is small and r is large. However, modest ensemble sizes (i.e., $N = 20$) allows the use of large r with no significant impact in the quality of the analysis innovations. It can be easily seen from figures 4.2a and 4.2b that a better estimation of the background error correlations provides analysis innovations which are not highly affected by sampling errors (spurious correlations) For instance, the EnKF-RS analysis approximations are better than those obtained by the LETKF wherein the sample covariance matrix is used. On the other hand, the quality of the analysis innovations of the EnKF-FS implementation are not subject to the value of ζ, as can be seen in figure 4.2c. Again, this reveals the importance of $\widehat{\mathbf{B}}$ as the estimator of $\mathbf{B}$: spurious correlations does not affect the quality of the analysis even for large radius sizes.



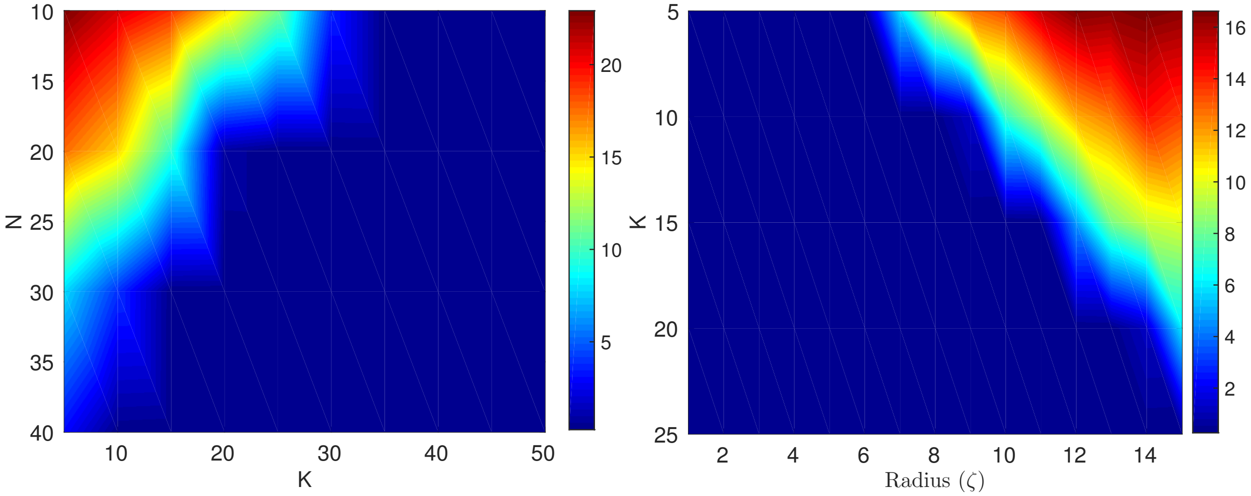
(a) LETKF (b) EnKF-RS



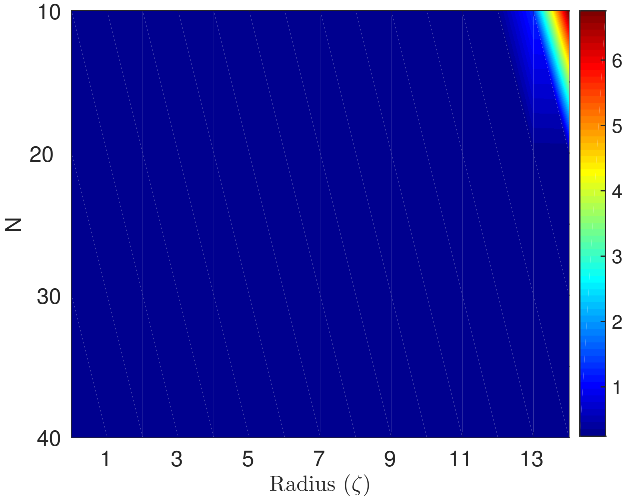
(c) EnKF-FS

Figure 4.2: Error distribution for different values of K using histograms for the 5-th model component of ensemble members from the Lorenz-96 model.

Notice, the EnKF-FS implementation outperforms the LETKF and the EnKF-RS methods in terms of accuracy. For instance, the RMSE values for the EnKF-FS ranges in $\sim [0.28,\, 0.30]$ while the RMSE values for the EnKF-RS and the LETKF implementations range in $\sim [2,\, 20]$ and $\sim [5,\, 25]$, respectively. The small variations in the context of EnKF-FS depends on the numerical computation of the singular values required in (4.11) during the assimilation step. As we mentioned before, LETKF is outperformed since it uses the ensemble covariance matrix in order to approximate $\mathbf{B}$. On the other hand, even though the EnKF-RS and the EnKF-FS implementations make use of the same $\widehat{\mathbf{B}}$ estimator, when no synthetic members are used, the EnKF-RS implementation projects $\widehat{\mathbf{B}}$ onto the ensemble space in order to compute the optimal analysis increments (perturbations) Thus, while the assimilation step of the EnKF-FS is performed onto the model space, the EnKF-RS does that in a reduce space whose dimension is much lower than the model space dimension. This explains why the EnKF-FS implementation outperforms to the EnKF-RS method. As we discussed in section 4.2.2, for the EnKF-RS method, it is possible to increase the sub-space size in which the assimilation is performed by the use of synthetic members. Notice, by increasing the sub-space size more information of $\widehat{\mathbf{B}}$ can be projected and therefore, more information about the background error correlations can be captured in such sub-space. To assess the importance of synthetic members in the context of EnKF-RS, three more sets of experiments are performed. In the first set of experiments, we avoid the use of localization in order to show how the use of synthetic members can mitigate the impact of sampling errors and therefore, the impact of spurious correlations. We vary the number of ensemble members (real members) in [10, 40] and we let the number of synthetic members K to range in [5, 50]. Figure 4.3a shows the RMSE values for different combinations of N and K. Note that, the quality of the results when the ensemble size is $N = 40$ (40 model propagations) and $K = 5$ synthetic members are used is lower than those obtained by making use of $N = 10$ real members (only 10 model propagations) and $K = 50$ synthetic ones (which do not require model propagations) The importance of synthetic members can be explained as follows: synthetic members provide useful directions consistent with the background error statistics. Thus, the set of relevant directions in which the analysis state is sought can be increased without the use of the forward model. However, in practice, domain localization (or decomposition) is needed in order to perform the assimilation owing to current operational model resolutions. In our second set of experiments, we explore the behaviour of the EnKF-RS for $N = 10$ model propagations (real members) and different localizations radii r in [1, 15]. We vary the number of synthetic members from 5 to 25. As can be seen in figure 4.3b, for 25 synthetic members, the size of the radius of influence does not affect the analysis innovations. Even more, the RMSE values in figure 4.3b are comparable to those shown in figure 4.2b wherein real members are used in order to increase the degrees of freedom of the sub-space. Thus, we can mitigate the impact of large radius of influences (spurious correlations) by making use of small number of real members and large number of synthetic ones. In the last set of experiments, we fix the number of synthetic members to 20 and the radii of influences are ranged in [1, 15]. The number of real members is varied from 10 to 40. The RMSE values for this test case are shown in figure 4.3c. The results reveal that a fixed number of synthetic members improve considerably the analysis innovations in the context of EnKF-RS. Notice, the RMSE values range in $\sim$ [1, 6] while when no synthetic members are used, the RMSE values range in $\sim$ [2, 10] (figure 4.2b)



(a) Global analysis (b) Fixed $N = 10$



(c) Fixed $K = 20$

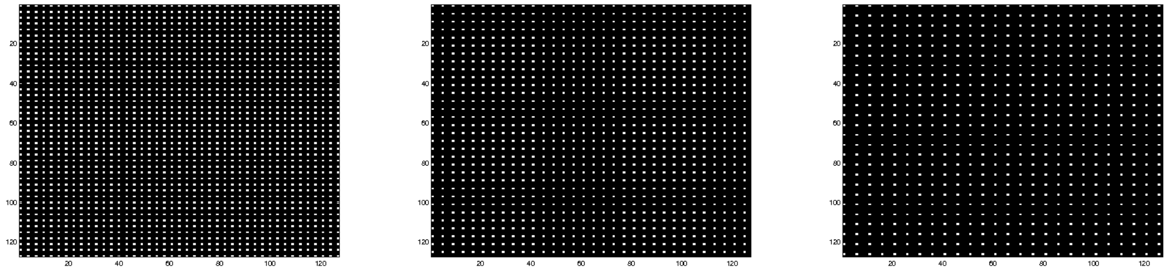
Figure 4.3: Contour of RMSE values of the EnKF-RS formulation for different configurations: in figure 4.3a, the assimilation is performed in the global domain for different values of N and K, likewise, figures 4.3b and 4.3c correspond to local analyses for fixed values of N and K, respectively.

4.4.2 The importance of $\widehat{\mathbf{B}}$ in the assimilation process

The oceans form a complex flow system influenced by the rotation of the Earth, the density stratification due to temperature and salinity, as well as other factors. The quasi-geostrophic (QG) model is a simple approximation of the real behaviour of the ocean. It is defined by the following partial differential equation [SO08a]:

$$\omega_t + k \cdot J\left(\omega,\, \psi\right) + \psi_x = -\mu \cdot \Delta^3 \psi + 2 \cdot \pi \cdot \sin\left(2 \cdot \pi \cdot y\right) \tag{4.44}$$

in $\Omega \in [0,\, 1] \times [0,\, 1]$, where x and y represent the zonal and meridional space components, ω is the vorticity, ψ is the stream function, $J\left(\psi, \omega\right) = \psi_x \cdot \omega_y - \psi_y \cdot \omega_x$ is the Jacobian of two fields and Δ is the Laplacian operator. The coefficients $k = 10^{-5}$ and $\mu = 2 \times 10^{-11}$ are associated with the advection term and the biharmonic zonal friction, respectively. Moreover, the vorticity is related to the stream function by the elliptical equation: $\Delta\psi = \omega$. The interior of the domain Ω is covered by a computational grid of resolution $[129 \times 129]$. Since homogeneous boundary conditions are assumed, the total number of components in the model state is $n = 16129$. The initial reference solution, the initial background state and the initial background ensemble are built similarly to those in the Lorenz 96 experiments. We just changed the initial perturbation to 5% the reference vorticity for the perturbed background state and 5% the background state for the perturbed initial ensemble. The number of ensemble members is $N = 80$ for all the configurations. Observational errors are uncorrelated with variances $[0.001]^2$. Observations are taken every 10 time units. The time discretization of the model (4.44) uses of a fourth order Runge-Kutta method. The time step size is 1.27 (units) which represent one hour in the ocean. The integration is performed for 250 hours. Three observational networks are used for the experiments: $\mathbf{H}^{[1]}$, $\mathbf{H}^{[2]}$ and $\mathbf{H}^{[3]}$ are sparse observational errors in which 12%, 6% and 4% of the global model components are observed, respectively. Again, we make use of perfect observational operators and therefore, we expect the main source of error to come from the misspecification of background error correlations. We denote by p the percentage of components observed from the model state. In the context of EnKF-RS, we let the number of synthetic members to be $K = 10$.



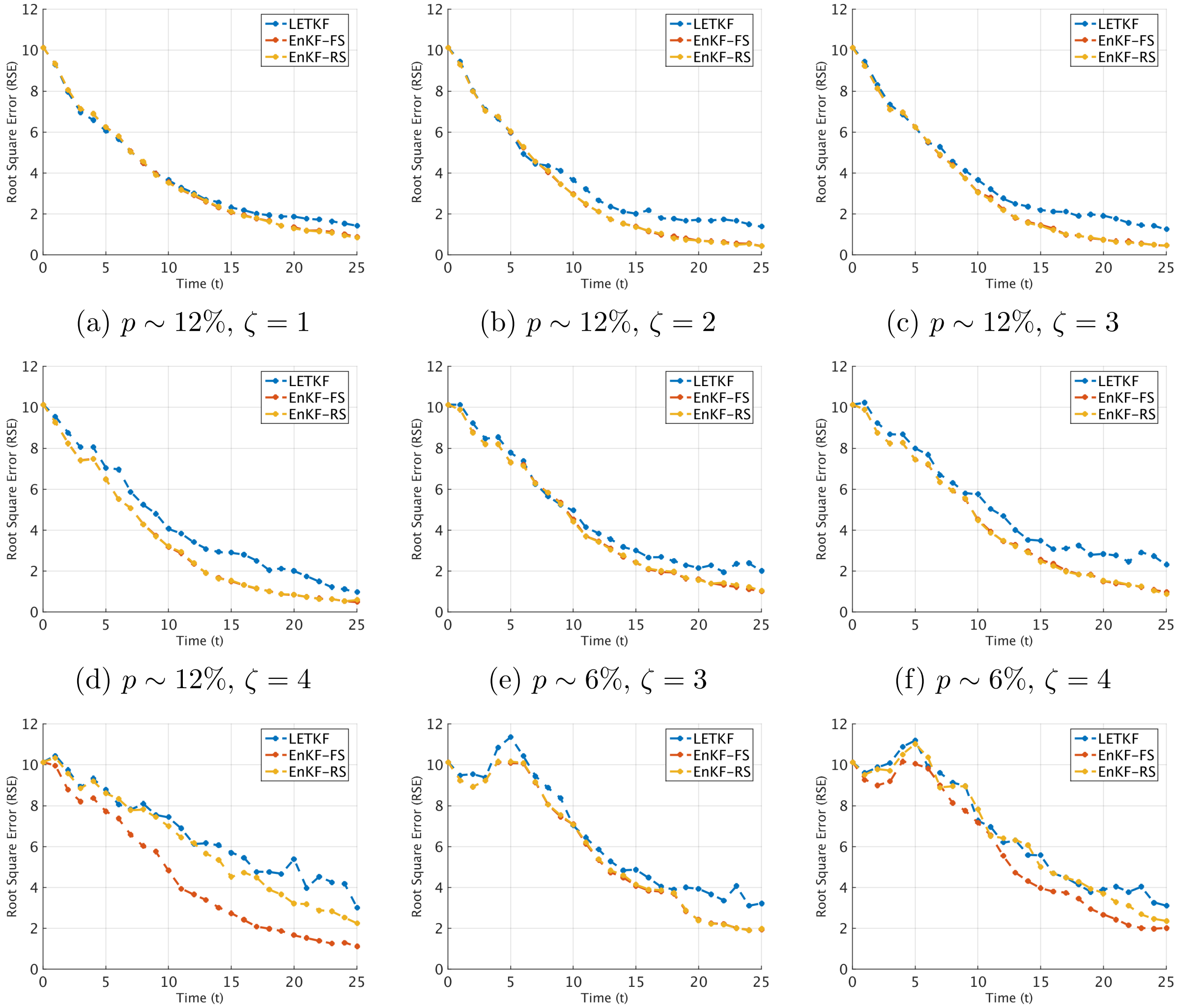
(a) $p \sim 12\%$ (b) $p \sim 6\%$ (c) $p \sim 4\%$

Figure 4.4: Sparse observational networks. Dark regions represent unobserved components. p denotes the approximated number of observed components from the global domain.

The RMSE values and CPU-times in seconds (s) are reported in the Table 4.1 for different values of p, radius of influence ζ and EnKF implementations. In general, for small radius of influences and dense observational networks the solutions of the EnKF implementations are comparable in most of the assimilations steps. However, for sparse observational networks and large radii of influences, the solutions provided by the EnKF-FS approximation outperforms those obtained by the EnKF-RS and the LETKF implementations. As we mentioned before, the background error correlations are well-represented by $\widehat{\mathbf{B}}$ and therefore, when sparse observational networks are considered, the unobserved components are properly adjusted with regard to the observed ones via the background error correlations in $\widehat{\mathbf{B}}$. Figure 4.5 shows the RMSE values across different observational times. In general, the EnKF-RS and the EnKF-FS provide analysis solutions which are more accurate than those obtained via the LETKF. Recall that, since the ensemble covariance matrix is sensitive to spurious correlations mainly owing to its rank-deficiency, the impact of observed components over unobserved ones is misspecified in $\mathbf{P}^b$. Thus, when the radius of influence is sufficiently large, spurious correlations affect the analysis innovation in the LETKF context. On the other hand, based on the results, we can infer that the impact of spurious correlations in the EnKF context can be mitigated by making use of the shrinkage covariance matrix estimator of background errors $\widehat{\mathbf{B}}$ as is proposed in the EnKF-FS and the EnKF-FS implementations.

Table 4.1: RMSE and CPU-time (s) for the EnKF-FS, EnKF-RS and LETKF implementations applied to the quasi-geostrophic model.

p	ζ	Method	RMSE	CPU-time (s)
12 %	1	LETKF	4.279	28.094
		EnKF-FS	4.248	61.410
		EnKF-RS	4.246	26.657
	2	LETKF	4.187	29.063
		EnKF-FS	3.951	54.823
		EnKF-RS	3.954	35.178
	3	LETKF	4.348	35.951
		EnKF-FS	4.066	76.593
		EnKF-RS	4.062	56.038
	4	LETKF	4.841	39.613
		EnKF-FS	4.173	104.796
		EnKF-RS	4.172	91.290
6 %	3	LETKF	5.248	37.189
		EnKF-FS	4.939	76.427
		EnKF-RS	4.934	55.019
	4	LETKF	5.600	40.469
		EnKF-FS	4.995	101.034
		EnKF-RS	4.986	89.788
	5	LETKF	6.783	41.850
		EnKF-FS	5.096	111.560
		EnKF-RS	6.359	81.814
4 %	3	LETKF	6.526	37.207
		EnKF-FS	6.453	80.791
		EnKF-RS	6.479	55.045
	4	LETKF	6.948	42.981
		EnKF-FS	6.370	100.360
		EnKF-RS	6.396	90.959
	5	LETKF	7.176	43.344
		EnKF-FS	6.386	108.724
		EnKF-RS	7.032	81.077



(a) $p \sim 12\%$, $\zeta = 1$ (b) $p \sim 12\%$, $\zeta = 2$ (c) $p \sim 12\%$, $\zeta = 3$

(d) $p \sim 12\%$, $\zeta = 4$ (e) $p \sim 6\%$, $\zeta = 3$ (f) $p \sim 6\%$, $\zeta = 4$

(g) $p \sim 6\%$, $\zeta = 5$ (h) $p \sim 4\%$, $\zeta = 4$ (i) $p \sim 4\%$, $\zeta = 5$

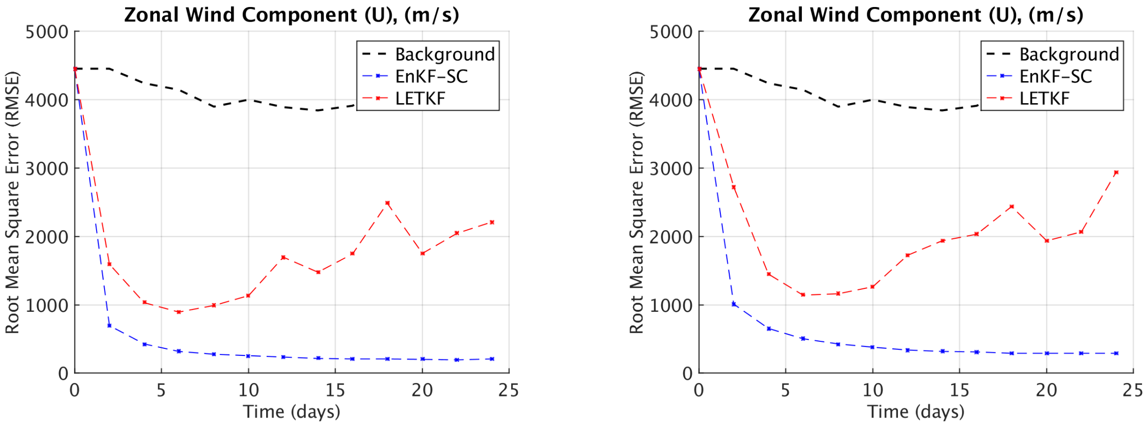
Figure 4.5: Plots of RMSE values of the EnKF-FS, the EnKF-RS and the LETKF implementations for different values of p and radius sizes ζ.

4.4.3 Parallel EnKF implementation

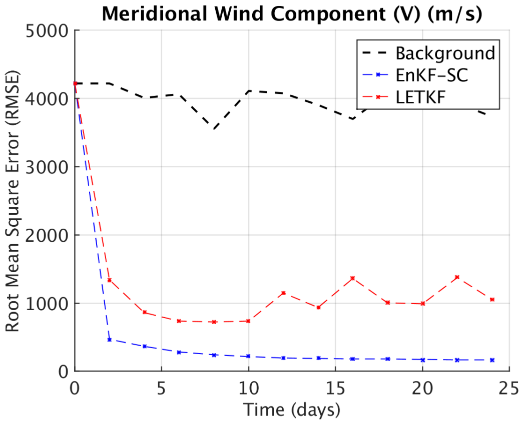
The proposed parallel implementation of the EnKF based on shrinkage covariance matrix estimation (EnKF-SC) is compared against the well-known LETKF formulation [OHS$^+$08, OHS$^+$04b] in terms of accuracy and parallel performance. The tests are performed on the super computer Blueridge cluster at the university of Virginia Tech. BlueRidge is a 408-node Cray CS-300 cluster. Each node is outfitted with two octa-core Intel Sandy Bridge CPUs and 64 GB of memory, for a total of 6528 cores and 27.3 TB of memory systemwide. The Atmospheric Global Circulation Model better known as SPEEDY is used for the simulation

[Mol03a]. The variables to be assimilated are the wind components u and v, the temperature T and the specific humidity sh. Thus, the vector state is formed by 4 model variables $\mathbf{x} = [u,\, v,\, T,\, sh]^T$. There are eight layers for each model variable with resolution T-63 (192×96) for a total number of model components $n = 589,824$. The reference solution is built after a model propagation over one year. From here, an initial perturbation of 5% on the reference solution and three months of model propagation provide the background state. Similarly, the initial ensemble is built over perturbations on the background state and long model propagations. The total number of ensemble members for all the experiments is $N = 94$. The methods are coded making use of FORTRAN 90 and 77, the linear algebra computations are performed making use of BLAS and LAPACK functions. The parallel computations are performed making use of MPI. Observational errors are assumed to be uncorrelated and Normal distributed with variances 0.01^2. No representativeness errors are considered in the simulated data therefore, we expect the missestimation of background errors to be the main source of errors in the analysis corrections. The simulation is performed over a period of 24 days. Observations are taken every two days (48 hours) We consider the Root Mean Square Error (RMSE) in order to measure the accuracy of the analysis solutions provided by the EnKF-SC and the LETKF.

The results in the Tables in the Tables 4.2 and 4.3 reveal that, the use of shrinkage covariance matrix estimation can mitigate the impact of spurious correlations. Note that, the accuracy of the EnKF-SC method in some cases is up to one order of magnitude with regard to the LETKF accuracy. In figure 4.6 the impact of spurious correlations affect the performance of the LETKF while, the background error correlations estimated via the RBLW estimator provide good analysis increments in the context of EnKF-SC. Snapshots of the analysis states after the first assimilation are shown in figures 4.7 and 4.8 for the meridional and the zonal wind components. Furthermore, we show in figure 4.9 the RMSE values for the compared implementations by varying the number of processors from 96 (6 computing nodes) to 2,048 (128 computing nodes) Note that, the results of the EnKF-SC slightly vary for different number of processors. This is expected since the EnKF-SC is a stochastic filter which makes use of synthetic data in orer to perform the assimilation. The local synthetic data depends on the processors id and therefore, for different number of processors (sub-domains) the results will present small variations. However, as can be seen, across different number of processors, the results of the EnKF-SC implementation tend to look similar. On the other hand, the results of LETKF does not change when the number of sub-domains is varied since that is a deterministic filter. Figure 4.10 shows the computational time of the compared implementations for different number of processors. Clearly, the use of domain decomposition reduces the elapsed time of the EnKF-SC implementation.



(a) $r = 3$ and $p = 12\%$ (b) $r = 5$ and $p = 6\%$



(c) $r = 4$ and $p = 12\%$

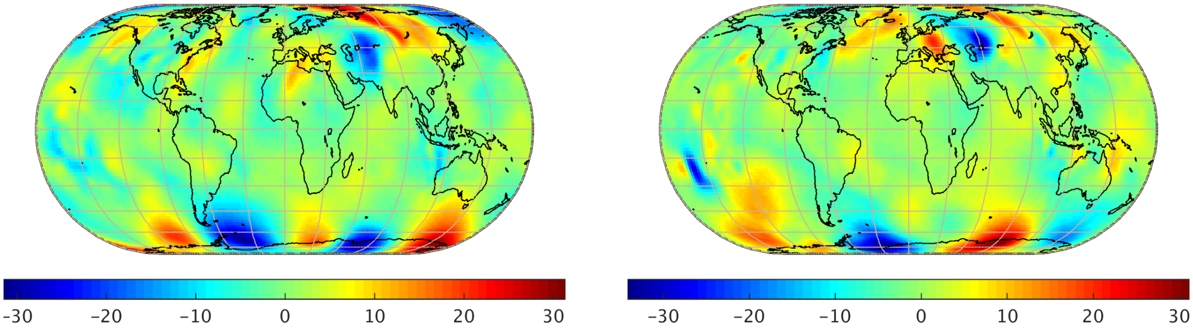
Figure 4.6: RMSE of the LETKF and EnKF-SC implementations for different model variables, radii of influence and observational networks.

Table 4.2: RMSE values of the EnKF-SC and the LETKF analyses for the wind-components from the SPEEDY model.

Variable (units)	ζ	p	EnKF-SC	LETKF
Zonal Wind Component (u), (m/s)	1	12%	4.872×10^2	5.471×10^2
		6%	1.228×10^3	1.168×10^3
		4%	1.766×10^3	1.737×10^3
	2	12%	3.330×10^2	1.275×10^3
		6%	5.705×10^2	7.591×10^2
		4%	8.444×10^2	8.569×10^2
	3	12%	3.186×10^2	1.661×10^3
		6%	4.854×10^2	1.237×10^3
		4%	6.750×10^2	9.997×10^2
	4	12%	3.161×10^2	1.752×10^3
		6%	4.729×10^2	1.608×10^3
		4%	6.261×10^2	1.258×10^3
	5	12%	3.334×10^2	1.862×10^3
		6%	4.730×10^2	1.983×10^3
		4%	6.148×10^2	1.602×10^3
Meridional Wind Component (v) (m/s)	1	12%	4.214×10^2	4.496×10^2
		6%	1.218×10^3	1.158×10^3
		4%	1.775×10^3	1.749×10^3
	2	12%	2.559×10^2	7.285×10^2
		6%	4.727×10^2	5.520×10^2
		4%	7.416×10^2	7.420×10^2
	3	12%	2.464×10^2	9.510×10^2
		6%	3.845×10^2	8.334×10^2
		4%	5.616×10^2	7.455×10^2
	4	12%	2.513×10^2	1.048×10^3
		6%	3.787×10^2	1.146×10^3
		4%	5.189×10^2	9.026×10^2
	5	12%	2.729×10^2	1.101×10^3
		6%	3.871×10^2	1.575×10^3
		4%	5.139×10^2	1.102×10^3

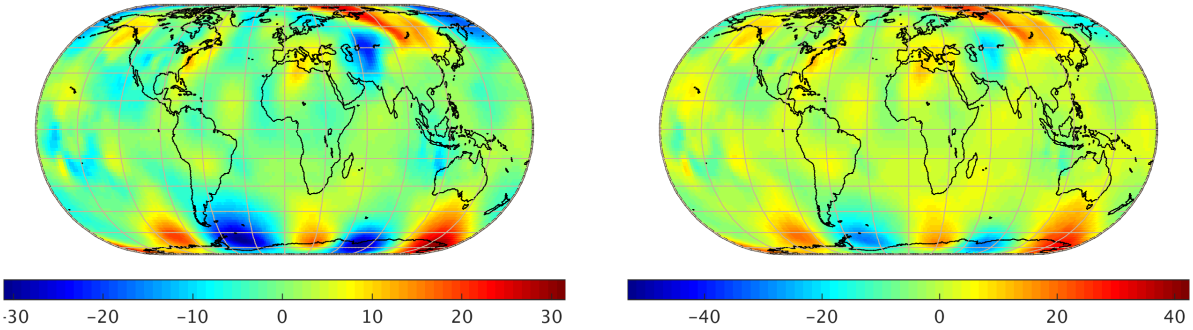
Table 4.3: RMSE values of the EnKF-SC and the LETKF analyses for the Temperature and the Specific Humidity from the SPEEDY model.

Variable (units)	ζ	p	EnKF-SC	LETKF
Temperature (K)	1	12%	3.854×10^2	6.033×10^2
		6%	6.063×10^2	6.704×10^2
		4%	7.847×10^2	8.073×10^2
	2	12%	3.068×10^2	6.693×10^2
		6%	3.830×10^2	5.556×10^2
		4%	4.678×10^2	5.529×10^2
	3	12%	2.728×10^2	1.217×10^3
		6%	3.308×10^2	6.458×10^2
		4%	3.966×10^2	6.073×10^2
	4	12%	2.576×10^2	1.817×10^3
		6%	3.097×10^2	1.030×10^3
		4%	3.664×10^2	7.464×10^2
	5	12%	2.561×10^2	1.600×10^3
		6%	3.015×10^2	1.473×10^3
		4%	3.511×10^2	1.172×10^3
Specific Humidity (g/Kg)	1	12%	9.998×10^1	9.026×10^1
		6%	1.634×10^2	1.449×10^2
		4%	2.063×10^2	1.941×10^2
	2	12%	8.081×10^1	1.125×10^2
		6%	1.082×10^2	1.137×10^2
		4%	1.393×10^2	1.321×10^2
	3	12%	7.751×10^1	1.341×10^2
		6%	9.813×10^1	1.418×10^2
		4%	1.222×10^2	1.458×10^2
	4	12%	7.762×10^1	1.640×10^2
		6%	9.692×10^1	1.652×10^2
		4%	1.167×10^2	1.739×10^2
	5	12%	8.094×10^1	2.078×10^2
		6%	9.775×10^1	1.950×10^2
		4%	1.158×10^2	2.068×10^2



(a) Reference

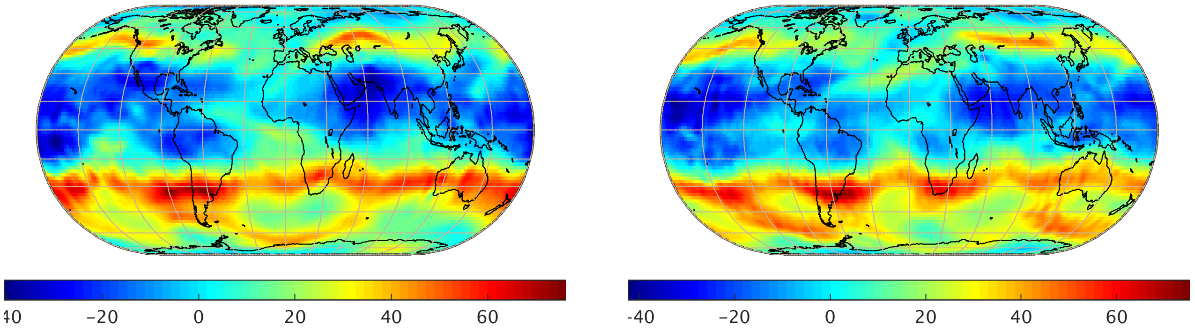
(b) Background



(c) EnKF-SC

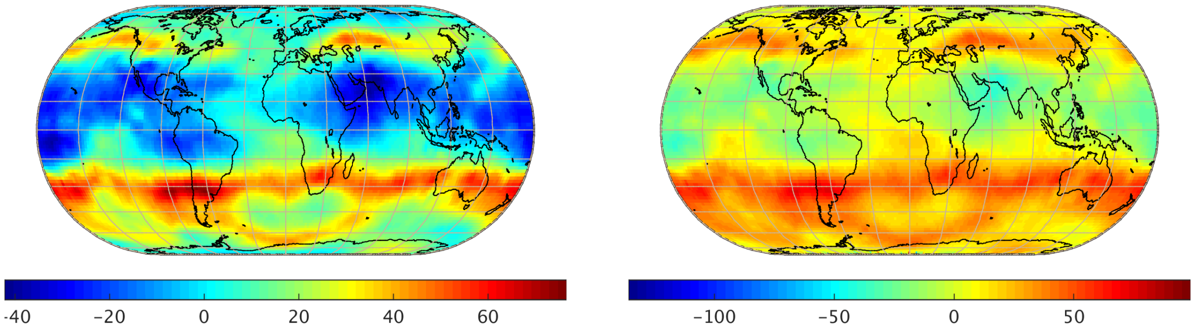
(d) LETKF

Figure 4.7: Snapshots of the reference solution, background state and analyses fields from the EnKF-SC and LETKF for the 5th layer of the meridional wind component (v). The percentage of observed component from the model state is $p \sim 4\%$.



(a) Reference

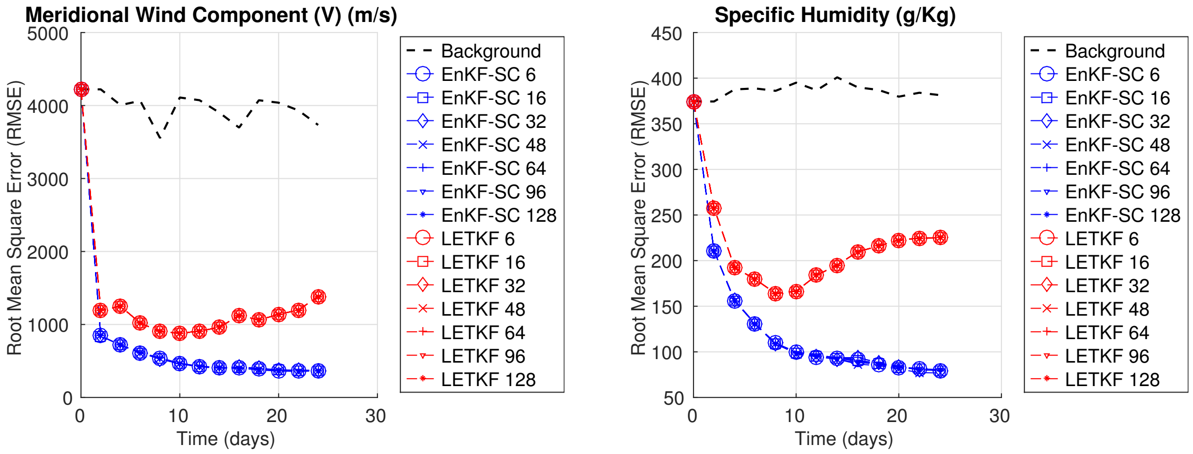
(b) Background



(c) EnKF-SC

(d) LETKF

Figure 4.8: Snapshots of the reference solution, background state and analyses fields from the EnKF-SC and LETKF for the 2th layer of the zonal wind component (u). The percentage of observed component from the model state is $p \sim 4\%$.



(a) Meridional wind component

(b) Specific humidity

Figure 4.9: RMSE values for different number of processors (sub-domains) for the EnKF-SC and the LETKF implementations. The number of computing nodes (x 16 processors) is next to each label.

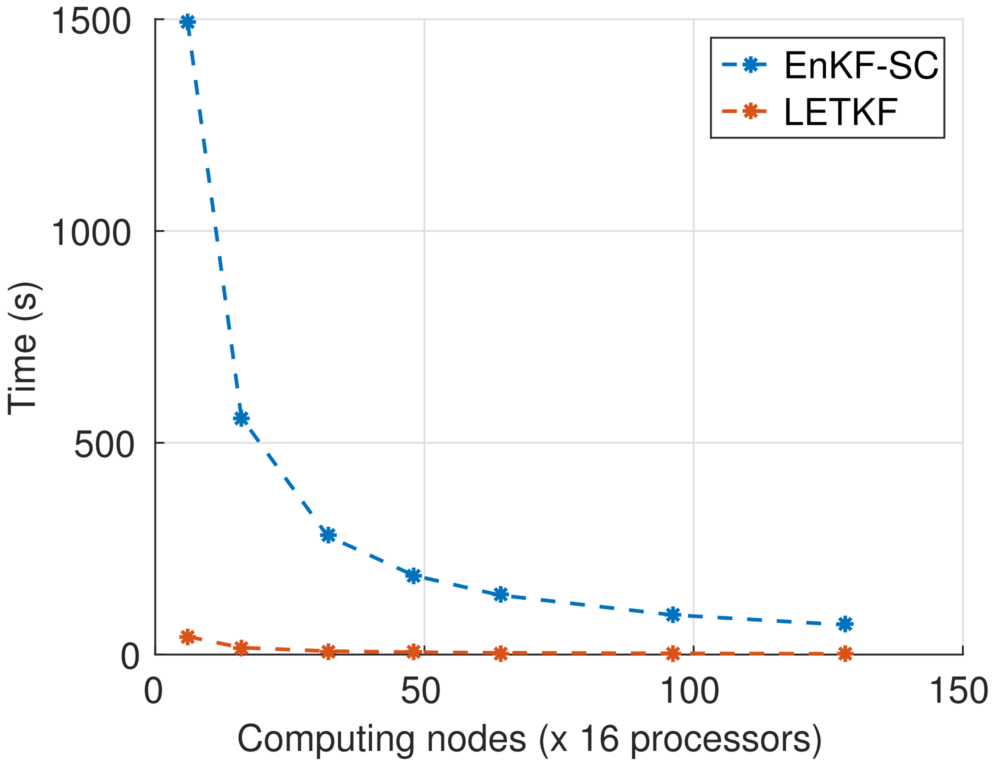


Figure 4.10: Computational time for the EnKF-SC and the LETKF implementations for different number of computing nodes (16 x processors) and $p \sim 4\%$.

4.5 Conclusions

This chapter develops two new implementations of the ensemble Kalman filter (EnKF) based on shrinkage covariance estimation. The background error covariance matrices used in analysis are obtained via the Rao-Blackwell Ledoit and Wolf estimator, which has been proved optimal in the estimation of high-dimensional covariance matrices from a small number of samples. This covariance matrix together with the ensemble mean are the parameters of the estimated normal background error distribution. Two implementations are proposed: EnKF Full-Space (EnKF-FS) performs the analysis in the model space, while EnKF Reduce-Space (EnKF-RS) computes the analysis state in the space spanned by the ensemble members. In the EnKF-RS context, additional samples from the estimated normal background error distribution distribution are taken in order to decrease the sampling error, and to increase the number of degrees of freedom in the assimilation. Covariance localization and covariance inflation are discussed in the context of the proposed methods. Even more, an adjoint-free four dimensional approximation is proposed for the EnKF-RS method. Besides, the parallel implementation EnKF-SC based on the full space approach EnKF-FS is presented. Three sets of experiments are carried out in order to assess the accuracy of the proposed EnKF implementations: in the first set of experiments, the Lorenz-96 model is used. The results reveals that the use of synthetic members can mitigate the impact of spurious correlations in the EnKF-RS context. In addition, the quality of the analysis innovations can be improved by increasing the number of synthetic members which do not require the use of the forward model operator. The solutions obtained by the EnKF-RS with small ensemble sizes (model propagations) and large synthetic members are of similar quality as those obtained by the LETKF with large ensemble sizes. Since the total compute time is dominated by the multiple model runs, considerable savings are expected from reducing the number of real ensemble members without deteriorating the quality of the results. In the second set of experiments, a quasi-geostrophic model is used. The results shows that the new implementations perform better than the widely popular local ensemble transform Kalman filter. The accuracy of the EnKF implementations is strictly related to the quality of the background error estimation since no representativeness errors are considered in the preparation of the synthetic data for the experiments. In the last set of experiments, an atmospheric general circulation model is used and the EnKF-SC is tested. The results reveal that domain decomposition can be used in order to reduce the computational time of the analysis step without compromising the quality of the results by the local approximations. In general, the experimental results support the idea that shrinkage covariance matrix estimation can be exploited in order to reduce the impact of spurious correlations in the context of data assimilation.

Chapter 5

An Ensemble Kalman Filter Implementation Based on Modified Cholesky Decomposition

5.1 Introduction

This chapter develops an efficient implementation of the ensemble Kalman filter based on a modified Cholesky decomposition for inverse covariance matrix estimation. This implementation is named EnKF-MC. Background errors corresponding to distant model components with respect to some radius of influence are assumed to be conditionally independent. This allows to obtain sparse estimators of the inverse background error covariance matrix. The computational effort of the proposed method is discussed and different formulations based on varios matrix identities are provided. Furthermore, an asymptotic proof of convergence with regard to the ensemble size is presented. In order to assess the performance and the accuracy of the proposed method, experiments are performed making use of the Atmospheric General Circulation Model SPEEDY. The results are compared against those obtained using the local ensemble transform Kalman filter (LETKF). Tests are performed for dense observations (100% and 50% of the model components are observed) as well as for sparse observations (only 12%, 6%, and 4% of model components are observed). The results reveal that the use of modified Cholesky for inverse covariance matrix estimation can reduce the impact of spurious correlations during the assimilation cycle, i.e., the results of the proposed method are of better quality than those obtained via the LETKF in terms of root mean square error.

This paper is organized as follows. In Section 5.2 the modified Cholesky decomposition is introduced. Section 5.3 discusses the proposed ensemble Kalman filter based on a modified Cholesky decomposition for inverse covariance matrix estimation; a theoretical convergence of the estimator in the context of data assimilation as well as its computational effort are discussed. Section 5.4 presents numerical experiments using the Atmospheric General Circulation Model SPEEDY; the results of the new filter are compared against those obtained by the local ensemble transform Kalman filter. Future work is discussed in Section 5.4.5 and conclusions are drawn in Section 5.5.

5.2 Modified Cholesky decomposition for inverse covariance matrix estimation

Consider a sample of N Gaussian random vectors:

$$\mathbf{S} = \left[\mathbf{s}^{[1]},\, \mathbf{s}^{[2]},\, \dots,\, \mathbf{s}^{[N]}\right] \in \mathbb{R}^{n \times N},$$

with statistical moments:

$$\mathbf{s}^{[j]} \sim \mathcal{N}\left(\mathbf{0}_n,\, \mathbf{Q}\right), \text{ for } 1 \leq j \leq N,$$

where $\mathbf{s}^{[j]} \in Re^{n \times 1}$ denotes the j-th sample. Denote by $\mathbf{x}_{[i]} \in \mathbb{R}^{N \times 1}$ the vector holding the i-th component across all the samples (the i-th row of $\mathbf{S}$, transposed). The modified Cholesky decomposition arises from regressing each component on his predecessors [BL08] according to some component ordering:

$$\mathbf{x}_{[i]} = \sum_{j=1}^{i-1} \mathbf{x}_{[j]} \cdot \beta_{i,j} + \boldsymbol{\varepsilon}_{[i]} \in \mathbb{R}^{N \times 1}, \quad 2 \leq i \leq n, \tag{5.1}$$

where $\mathbf{x}_{[j]}$ is the j-th model component which precedes $\mathbf{x}_{[i]}$ for $1 \leq j \leq i-1$, $\boldsymbol{\varepsilon}_{[1]} = \mathbf{x}_{[1]}$, and $\boldsymbol{\varepsilon}_{[i]} \in \mathbb{R}^{N \times 1}$ is the error in the i-th component regression for $i \geq 2$. Likewise, the coefficients $\beta_{i,j}$ in (5.1) can be computed by solving the optimization problem:

$$\boldsymbol{\beta}_{[i]} = \arg\min_{\boldsymbol{\beta}} \left\| \mathbf{x}_{[i]} - \mathbf{Z}_{[i]} \cdot \boldsymbol{\beta} \ \right\|_2^2 \tag{5.2}$$

where

$$\begin{aligned} \mathbf{Z}_{[i]} &= \left[\mathbf{x}^{[1]},\, \mathbf{x}^{[2]},\, \dots,\, \mathbf{x}^{[i-1]}\right]^T \in \mathbb{R}^{(i-1) \times N}, \quad 2 \leq i \leq n, \\ \boldsymbol{\beta}_{[i]} &= \left[\beta_{i,1},\, \beta_{i,2},\, \dots,\, \beta_{i,i-1}\right]^T \in \mathbb{R}^{(i-1) \times 1}. \end{aligned}$$

The regression coefficients form the lower triangular matrix

$$\left\{\widehat{\mathbf{T}}\right\}_{i,j} = \begin{cases} -\beta_{i,j} \text{ for } 1 \leq j < i, \\ \quad 1 \text{ for } j = i, \qquad\quad 1 \leq i \leq n, \\ \quad 0 \text{ for } j > i, \end{cases} \tag{5.3a}$$

where $\left\{\widehat{\mathbf{T}}\right\}_{i,j}$ denotes the (i,j)-th component of matrix $\widehat{\mathbf{T}} \in \mathbb{R}^{n\times n}$. The empirical variances $\widehat{\mathbf{var}}$ of the residuals $\boldsymbol{\varepsilon}_{[i]}$ form the diagonal matrix:

$$\widehat{\mathbf{D}} = \operatorname*{diag}_{1\le i\le n}\left(\widehat{\mathbf{var}}(\boldsymbol{\varepsilon}_{[i]})\right) = \operatorname*{diag}_{1\le i\le n}\left(\frac{1}{N-1}\sum_{j=1}^{N}\left\{\boldsymbol{\varepsilon}_{[i]}\right\}_j^2\right) \in \mathbb{R}^{n\times n}\,. \tag{5.3b}$$

where $\left\{\widehat{\mathbf{D}}\right\}_{1,1} = \widehat{\mathbf{var}}\left(\mathbf{x}_{[1]}\right)$. Then an estimate of $\mathbf{Q}^{-1}$ can be computed as follows:

$$\widehat{\mathbf{Q}}^{-1} = \widehat{\mathbf{T}}^{T} \cdot \widehat{\mathbf{D}}^{-1} \cdot \widehat{\mathbf{T}} \in \mathbb{R}^{n\times n}, \tag{5.4a}$$

or, by basic matrix algebra identities the estimate of $\mathbf{Q}$ reads:

$$\widehat{\mathbf{Q}} = \widehat{\mathbf{T}}^{-1} \cdot \widehat{\mathbf{D}} \cdot \widehat{\mathbf{T}}^{-T} \in \mathbb{R}^{n\times n}\,. \tag{5.4b}$$

Note that the structure of $\widehat{\mathbf{Q}}^{-1}$ is strictly related to the structure of $\widehat{\mathbf{T}}$. This can be exploited in order to obtain sparse estimators of $\mathbf{Q}^{-1}$ by imposing that some entries of $\widehat{\mathbf{T}}$ are zero. This is important for high dimensional probability distributions where the explicit computation of $\widehat{\mathbf{Q}}$ or $\widehat{\mathbf{Q}}^{-1}$ is prohibitive. The zero components in $\widehat{\mathbf{T}}$ can be justified as follows: when two components are *conditionally independent* their corresponding entry in $\widehat{\mathbf{Q}}^{-1}$ is zero. In the context of data assimilation, the conditional independence of background errors between different model components can be achieved by making use of domain localization. We can consider zero correlations between background errors corresponding to model components located at distances that exceed a radius of influence ζ. In the next section we present an ensemble Kalman filter implementation based on modified Cholesky decomposition for inverse covariance matrix estimation.

5.3 Ensemble Kalman Filter Based On Modified Cholesky Decomposition

In this section we discuss the new ensemble Kalman filter based on modified Cholesky decomposition for inverse covariance matrix estimation (EnKF-MC).

5.3.1 Estimation of the inverse background covariance

The columns of matrix (2.13a)

$$\mathbf{U}^b = \left[\mathbf{u}^{b[1]},\, \mathbf{u}^{b[2]},\, \dots,\, \mathbf{u}^{b[N]}\right] \in \mathbb{R}^{n\times N}$$

can be seen as samples of the (approximately normal) distribution:

$$\mathbf{x}^{b[j]} - \overline{\mathbf{x}}^b = \mathbf{u}^{b[j]} \sim \mathcal{N}\left(\mathbf{0},\, \mathbf{B}\right), \text{ for } 1 \le j \le N\,,$$

and therefore, if we let $\mathbf{x}_{[i]} \in \mathbb{R}^{N\times 1}$ in (5.1) to be the vector formed by the i-th row of matrix (2.13a), for $1 \leq i \leq n$, according to equations (5.4), an estimate of the inverse background error covariance matrix reads:

$$\mathbf{B}^{-1} \approx \widehat{\mathbf{B}}^{-1} = \widehat{\mathbf{T}}^{T} \cdot \widehat{\mathbf{D}}^{-1} \cdot \widehat{\mathbf{T}} \in \mathbb{R}^{n\times n}, \tag{5.5a}$$

and therefore:

$$\mathbf{B} \approx \widehat{\mathbf{B}} = \widehat{\mathbf{T}}^{-1} \cdot \widehat{\mathbf{D}} \cdot \widehat{\mathbf{T}}^{-T} \in \mathbb{R}^{n\times n}\,. \tag{5.5b}$$

As we mentioned before, the structure of $\widehat{\mathbf{B}}^{-1}$ depends on that of $\widehat{\mathbf{T}}$. If we assume that the correlations between model components are local, and there are no correlations outside a radius of influence ζ, we obtain lower-triangular sparse estimators of $\widehat{\mathbf{T}}$. Consequently, the resulting $\widehat{\mathbf{B}}^{-1}$ will also be sparse, and $\widehat{\mathbf{B}}$ will be localized. Since the regression (5.1) is performed only on the predecessors of each model component, an ordering (labeling) must be set on the model components prior the computation of $\widehat{\mathbf{T}}$. Since we work with gridded models we consider column-major and row-major orders. They are illustrated in Figure 5.1 for a two-dimensional domain. Figure 5.2 shows the local domain and the predecessors of the model component 6 when column-major order is utilized.

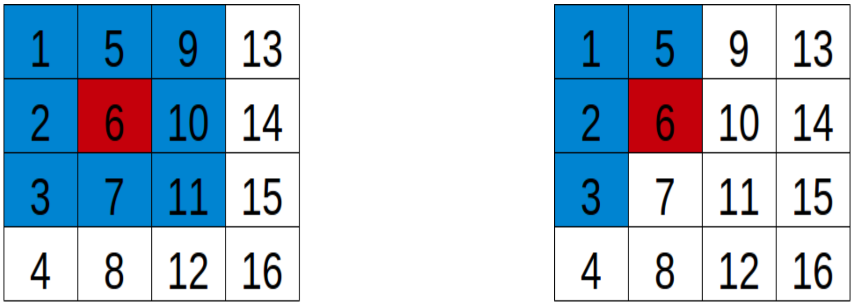
1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

(a) Column-major order

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

(b) Row-major order

Figure 5.1: Row-major and column-major ordering for a 4×4 domain. The total number of model components is 16.



(a) In blue, local box for the model component 6 when $\zeta = 2$.

(b) In blue, predecessors of the model component 6 for $\zeta = 2$.

Figure 5.2: Local model components (local box) and local predecessors for the model component 6 when $\zeta = 2$. Column-major ordering is utilized to label the model components.

The estimation of $\widehat{\mathbf{B}}^{-1}$ proceeds as follows:

1. Form the matrix $\mathbf{Z}_{[i]} \in \mathbb{R}^{p_i \times N}$ with the predecessors of the i-th model component:

$$\mathbf{Z}_{[i]} = \left[\mathbf{x}^{[q(i,1)]},\, \mathbf{x}^{[q(i,2)]},\, \dots,\, \mathbf{x}^{[q(i,p_i)]}\right]^T \in \mathbb{R}^{p_i \times N}, \tag{5.6}$$

   where $\mathbf{x}^{[e]}$ is the e-th row of matrix (2.13a), p_i is the number of predecessors of component i, and $1 \leq q(i,j) \leq n$ is the index of the j-th predecessor of the i-th model component.

2. For the i-th model components the regression coefficients are obtained as follows:

$$\mathbf{x}_{[i]} = \sum_{j=1}^{p_i} \beta_{i,j} \cdot \mathbf{x}^{[q(i,j)]} + \boldsymbol{\varepsilon}_{[i]} \in \mathbb{R}^{N \times 1}.$$

   For $2 \leq i \leq n$, compute $\boldsymbol{\beta}_{[i]} = [\beta_{i,1},\, \beta_{i,2},\, \dots,\, \beta_{i,p_i}] \in \mathbb{R}^{p_i \times 1}$ by solving the optimization problem (5.2) with $\mathbf{Z}_{[i]}$ given by (5.6).

3. Build the matrices

$$\left\{\widehat{\mathbf{T}}\right\}_{i,q(i,j)} = -\beta_{i,j} \quad \text{for} \;\; 1 \leq j \leq p_i, \;\; 1 < i \leq n\,; \quad \left\{\widehat{\mathbf{T}}\right\}_{i,i} = 1,$$

   and $\widehat{\mathbf{D}}$ according to equation (5.3b). Note that the number of non-zero elements in the i-th row of $\widehat{\mathbf{T}}$ equals the number of predecessors p_i.

Note that the solution of the optimization problem (5.2) can be obtained as follows:

$$\boldsymbol{\beta}_{[i]} = \left[\mathbf{Z}_{[i]} \cdot \mathbf{Z}_{[i]}{}^T\right]^{-1} \cdot \mathbf{Z}_{[i]} \cdot \mathbf{x}_{[i]} \tag{5.7}$$

and since the ensemble size can be smaller than the number of model components, $\mathbf{Z}_{[i]} \cdot {\mathbf{Z}_{[i]}}^T \in \mathbb{R}^{p_i \times p_i}$ can be rank deficient. To overcome this situation, regularization of the zero singular values of $\mathbf{Z}_{[i]} \cdot {\mathbf{Z}_{[i]}}^T$ can be used. One possibility is Tikhonov regularization [GHO99, KHE07, Mai94]:

$$\boldsymbol{\beta}_{[i]} = \arg\min_{\boldsymbol{\beta}} \left\{ \left\| \mathbf{x}_{[i]} - \mathbf{Z}_{[i]} \cdot \boldsymbol{\beta} \right\|_2^2 + \lambda^2 \cdot \left\| \boldsymbol{\beta} \right\|_2^2 \right\} \tag{5.8}$$

where $\lambda \in \mathbb{R}$. In our context the best choice for λ relies on prior knowledge of the background and the observational errors [Neu88]. Another approach to regularization is to use a truncated singular value decomposition (SVD) of $\mathbf{Z}_{[i]}$:

$$\mathbf{Z}_{[i]} = \mathbf{U}^{\mathbf{Z}_{[i]}} \cdot \boldsymbol{\Sigma}^{\mathbf{Z}_{[i]}} \cdot {\mathbf{V}^{\mathbf{Z}_{[i]}}}^T \in \mathbb{R}^{p_i \times N},$$

where $\mathbf{U}^{\mathbf{Z}_{[i]}} \in \mathbb{R}^{p_i \times p_i}$ and $\mathbf{V}^{\mathbf{Z}_{[i]}} \in \mathbb{R}^{N \times N}$ are the right and the left singular vectors of $\mathbf{Z}_{[i]}$, respectively. Likewise, $\boldsymbol{\Sigma}^{\mathbf{Z}_{[i]}} \in \mathbb{R}^{p_i \times N}$ is a diagonal matrix whose diagonal entries are the singular values of $\mathbf{Z}_{[i]}$ in descending order. The solution of (5.2) can be computed as follows [JB00, Huf91, Han90]:

$$\boldsymbol{\beta}_{[i]} = \sum_{j=1}^{k_i} \frac{1}{\tau_j} \cdot \mathbf{u}_j^{\mathbf{Z}_{[i]}} \cdot {\mathbf{v}_j^{\mathbf{Z}_{[i]}}}^T \cdot \mathbf{x}_{[i]} \quad \text{with} \; \frac{\tau_j}{\tau_{\max}} \geq \sigma_r, \tag{5.9}$$

where τ_j is the j-th singular value with corresponding right and left singular vectors $\mathbf{u}_j^{\mathbf{Z}_{[i]}} \in \mathbb{R}^{p_i \times 1}$ and $\mathbf{v}_j^{\mathbf{Z}_{[i]}} \in \mathbb{R}^{p_i \times 1}$, respectively, $\sigma_r \in (0, 1)$ is a predefined threshold, and $\tau_{\max} = \max\{\tau_1, \tau_2, \ldots, \tau_{N-1}\}$. Since small singular values are more sensitive to the noise in $\mathbf{x}_{[i]}$, the threshold $\tau_j > \tau_{\max} \cdot \sigma_r$ seeks to neglect their contributions.

5.3.2 Formulation of EnKF-MC

Once $\widehat{\mathbf{B}}^{-1}$ is estimated, the EnKF based on modified Cholesky decomposition (EnKF-MC) computes the analysis using Kalman's formula:

$$\mathbf{x}^{\mathrm{a}} = \mathbf{x}^b + \widehat{\mathbf{A}} \cdot \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \boldsymbol{\Delta} \in \mathbb{R}^{n \times N}, \tag{5.10a}$$

where $\widehat{\mathbf{A}} \in \mathbb{R}^{n \times n}$ is the estimated analysis covariance matrix

$$\widehat{\mathbf{A}} = \left[\widehat{\mathbf{B}}^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H} \right]^{-1},$$

and $\boldsymbol{\Delta} \in \mathbb{R}^{m \times N}$ is the innovation matrix on the perturbed observations $\mathbf{Y}^s \in \mathbb{R}^{m \times N}$, $\boldsymbol{\Delta} = \mathbf{Y}^s - \mathbf{H} \cdot X^b \in \mathbb{R}^{m \times N}$.

Computationally-friendlier alternatives to (5.10c) can be obtained by making use of elementary matrix identities:

$$\begin{aligned}
\mathbf{x}^{\mathrm{a}} &= \widehat{\mathbf{A}} \cdot \left[\widehat{\mathbf{B}}^{-1} \cdot \mathbf{x}^{b} + \mathbf{H}^{T} \cdot \mathbf{R}^{-1} \cdot \mathbf{Y}^{\mathrm{s}}\right] \in \mathbb{R}^{n \times N}, && (5.10\text{b})\\
\mathbf{x}^{\mathrm{a}} &= \mathbf{x}^{b} + \widehat{\mathbf{T}}^{-1} \cdot \widehat{\mathbf{D}}^{1/2} \cdot \mathbf{V}_{\widehat{\mathbf{B}}}^{T} \cdot \left[\mathbf{R} + \mathbf{V}_{\widehat{\mathbf{B}}} \cdot \mathbf{V}_{\widehat{\mathbf{B}}}^{T}\right]^{-1} \cdot \mathbf{\Delta}, && (5.10\text{c})\\
\mathbf{V}_{\widehat{\mathbf{B}}} &= \mathbf{H} \cdot \widehat{\mathbf{T}}^{-1} \cdot \widehat{\mathbf{D}}^{1/2} \in \mathbb{R}^{n \times m},
\end{aligned}$$

where $\mathbf{Y}^{\mathrm{s}}$ are the perturbed observations. The formulation (5.10c) is well-known as the EnKF dual formulation, (5.10b) is known as the EnKF primal formulation, and the equation (5.10a) is the incremental form of the primal formulation. In the next subsection, we discuss the computational effort of the EnKF-MC implementations (5.10).

5.3.3 Computational effort of EnKF-MC implementations

The computational cost of the different EnKF-MC implementations depend, in general, on the model state dimension n, the number of observed components m, the radius of influence ζ, and the ensemble size N. Typically [TAB^{+}03] the data error covariance matrix $\mathbf{R}$ has a simple structure (e.g., block diagonal), the ensemble size is much smaller than the model dimension ($n \gg N$), and the observation operator $\mathbf{H}$ is sparse or can be applied efficiently. We analyze the computational effort of the formulation (5.10a); similar analyses can be carried out for the other formulations. The incremental formulation can be written as follows:

$$\mathbf{x}^{\mathrm{a}} = \mathbf{x}^{b} + \boldsymbol{\delta}\mathbf{X}^{\mathrm{a}}\,,$$

where the analysis increments $\boldsymbol{\delta}\mathbf{X}^{\mathrm{a}} \in \mathbb{R}^{n \times N}$ are given by the solution of the linear system:

$$\left[\widehat{\mathbf{B}}^{-1} + \mathbf{R}_{\mathbf{H}} \cdot \mathbf{R}_{\mathbf{H}}^{T}\right] \cdot \boldsymbol{\delta}\mathbf{X}^{\mathrm{a}} = \mathbf{\Delta}_{\mathbf{H}}\,.$$

with $\mathbf{R}_{\mathbf{H}} = \mathbf{H}^{T} \cdot \mathbf{R}^{-1/2} \in \mathbb{R}^{n \times m}$, $\mathbf{\Delta}_{\mathbf{H}} = \mathbf{H}^{T} \cdot \mathbf{R}^{-1} \cdot \mathbf{\Delta} \in \mathbb{R}^{n \times N}$. This linear system can be solved making use of the iterative Sherman Morrison formula [NRSA14] as follows:

1. Compute:

$$\begin{aligned}
\mathbf{W}_{\mathbf{Z}}^{(0)[i]} &= \left[\widehat{\mathbf{T}}^{T} \cdot \mathbf{D}^{-1} \cdot \widehat{\mathbf{T}}\right]^{-1} \cdot \mathbf{\Delta}_{\mathbf{H}}^{[i]}, \text{ for } 1 \le i \le N, && (5.11\text{a})\\
\mathbf{W}_{\mathbf{U}}^{(0)[j]} &= \left[\widehat{\mathbf{T}}^{T} \cdot \mathbf{D}^{-1} \cdot \widehat{\mathbf{T}}\right]^{-1} \cdot \mathbf{R}_{\mathbf{H}}^{[j]}, \text{ for } 1 \le j \le m\,. && (5.11\text{b})
\end{aligned}$$

   where $\mathbf{\Delta}_{\mathbf{H}}^{[i]} \in \mathbb{R}^{n \times 1}$ and $\mathbf{R}_{\mathbf{H}}^{[j]} \in \mathbb{R}^{n \times 1}$ denote the i and j columns of matrices $\mathbf{\Delta}_{\mathbf{H}}$ and $\mathbf{R}_{\mathbf{H}}$, respectively. Since $\widehat{\mathbf{T}}$ is a sparse unitary lower triangular matrix, the direct solution of the linear system (5.11) can be obtained by making use of forward and backward substitutions. Hence, this step can be performed with:

$$\mathcal{O}\left(n_{nz} \cdot n \cdot N + n_{nz} \cdot n \cdot m\right) \qquad (5.12)$$

long computations, where n_{nz} denotes the maximum number of non-zero elements across all rows of $\widehat{\mathbf{T}}$, this is

$$n_{nz} = \max\{p_1,\, p_2,\, \ldots,\, p_n\}$$

where p_i is the number of predecessors of model component i, for $1 \leq i \leq n$.

2. For $1 \leq i \leq m$ compute:

$$\begin{aligned}
\mathbf{h}^{(i)} &= \frac{1}{\gamma^{(i)}} \cdot \mathbf{W}_{\mathbf{U}}^{(i-1)[i]}, \text{ with } \gamma^{(i)} = \left[1 + \mathbf{R}_{\mathbf{H}}^{[i]^T} \cdot \mathbf{W}_{\mathbf{U}}^{(i-1)[i]}\right]^{-1}, \\
\mathbf{W}_{\mathbf{Z}}^{(i)[j]} &= \mathbf{W}_{\mathbf{Z}}^{(i-1)[j]} - \mathbf{h}^{(i)} \cdot \left[\mathbf{R}_{\mathbf{H}}^{[i]^T} \cdot \mathbf{W}_{\mathbf{Z}}^{(i-1)[j]}\right], \text{ for } 1 \leq j \leq N\,, \\
\mathbf{W}_{\mathbf{U}}^{(i)[k]} &= \mathbf{W}_{\mathbf{U}}^{(i-1)[k]} - \mathbf{h}^{(i)} \cdot \left[\mathbf{R}_{\mathbf{H}}^{[i]^T} \cdot \mathbf{W}_{\mathbf{U}}^{(i-1)[k]}\right], \text{ for } i+1 \leq k \leq m\,.
\end{aligned}$$

Note that, at each step, $\mathbf{h}^{(i)}$ can be computed with n long computations, while $\mathbf{W}_{\mathbf{Z}}$ and $\mathbf{W}_{\mathbf{U}}$ can be obtained with $n \cdot N$ and $n \cdot m$ long computations, respectively. This leads to the next bound for the number of long computations:

$$\mathcal{O}\left(m \cdot n + m \cdot n \cdot N + m^2 \cdot n\right)\,.$$

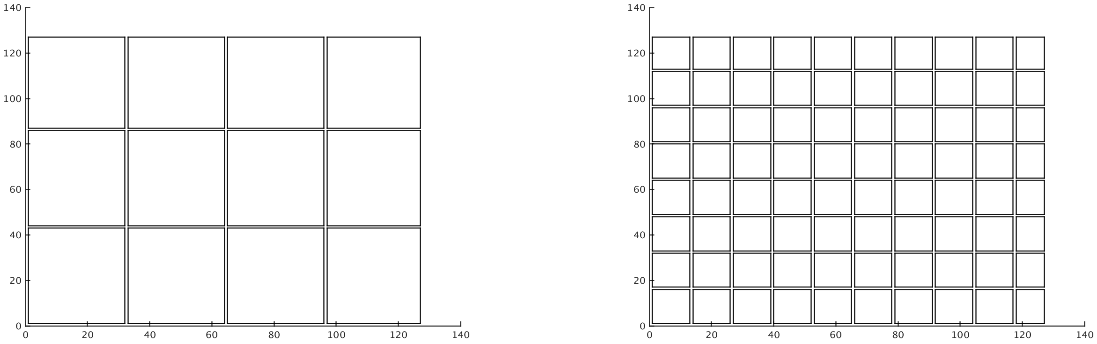
Hence, the computational effort involved during the assimilation step of formulation (5.10a) can be bounded by:

$$\mathcal{O}\left(m \cdot n + m \cdot n \cdot N + m^2 \cdot n + n_{nz} \cdot n \cdot N + n_{nz} \cdot n \cdot m\right),$$

which is linear with respect to the number of model components. For dense observational networks, when local observational operators can be approximated, domain decomposition can be exploited in order to reduce the computational effort during the assimilation cycle. This can be done as follows:

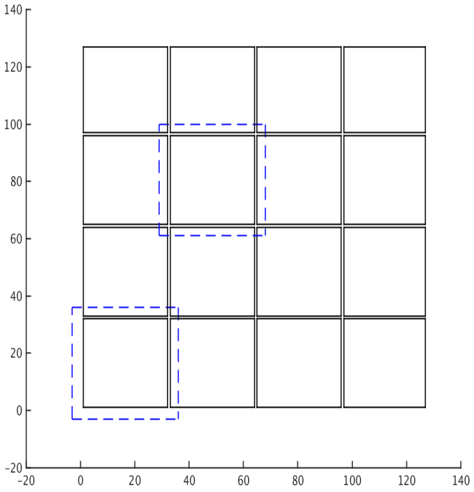
1. The domain is split in certain number of sub-domains (typically matching a given number of processors).
2. Background error correlations are estimated locally.
3. The assimilation is performed on each local domain.
4. The analysis sub-domains are mapped back onto the model domain from which the global analysis state is obtained.

Figure 5.3 shows the global domain splitting for different sub-domain sizes. In Figure 5.3c the boundary information needed during the assimilation step for two particular sub-domains is shown in dashed blue lines. Note that each sub-domain can be assimilated independently. Note that we only use domain decomposition in order to reduce the computational effort of the proposed implementation (and its derivations) and not in order to reduce the impact of spurious correlations.



(a) Number of sub-domains 12

(b) Number of sub-domains 80

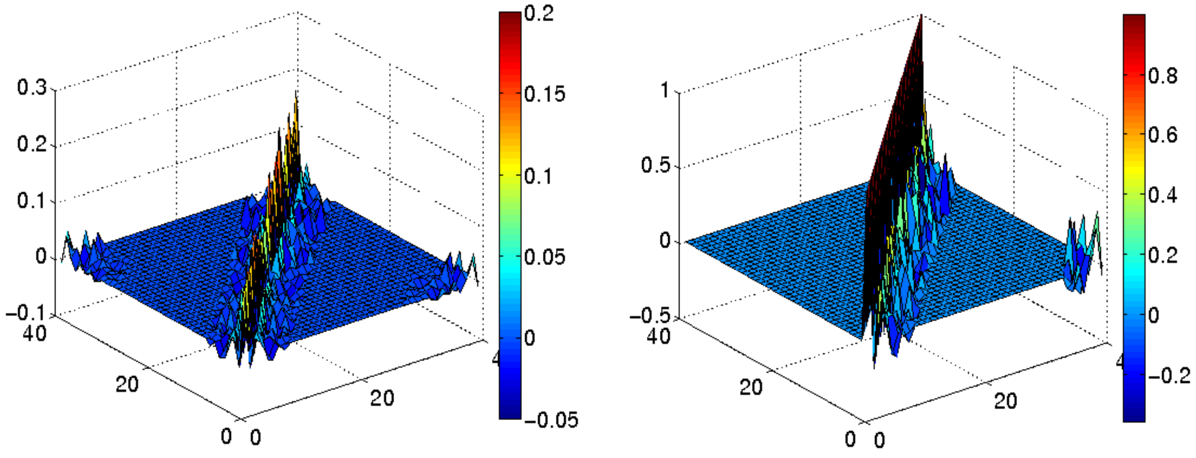


(c) Number of sub-domains 16.

Figure 5.3: Global domain splitting in different sub-domain sizes. Blue local boxes reflects the boundary information utilized in order to perform local data assimilation.

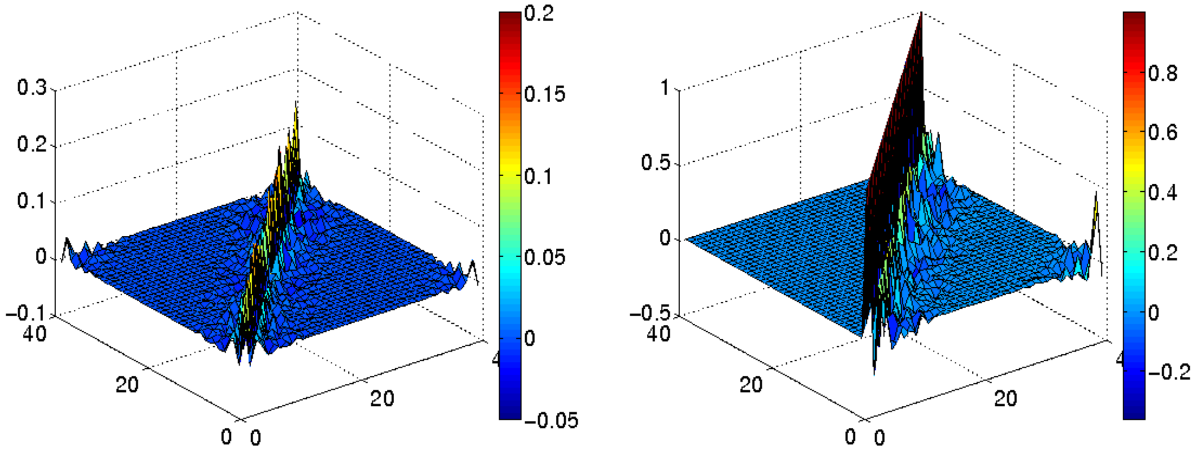
5.3.4 Convergence of the covariance inverse estimator

In this section we prove the convergence of the $\widehat{\mathbf{B}}^{-1}$ estimator in the context of data assimilation.



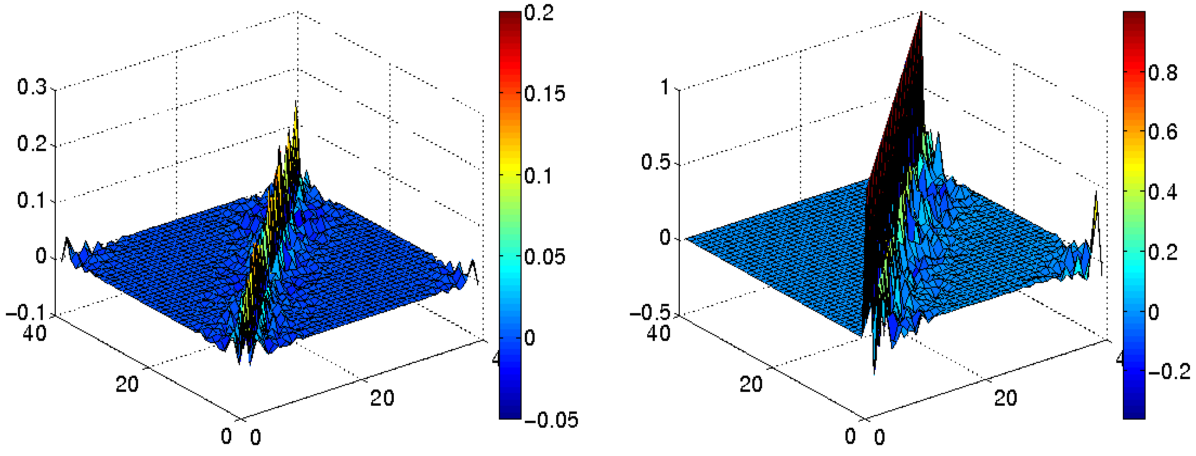
(a) Exact $\mathbf{B}^{-1} \approx \mathbf{P}^{b^{-1}}$ for $N = 10^5$

(b) $\mathbf{T}$, $\mathbf{B}^{-1} = \mathbf{T}^T \cdot \mathbf{D}^{-1} \cdot \mathbf{T}$



(c) Localized ensemble estimate $\widehat{\mathbf{P}^b}^{-1}$

(d) $\mathbf{T_L}$, $\widehat{\mathbf{P}^b}^{-1} = \mathbf{T}_\mathbf{L}^T \cdot \mathbf{D_L} \cdot \mathbf{T_L}$



(e) Cholesky estimate $\widehat{\mathbf{B}}^{-1}$

(f) $\widehat{\mathbf{T}}$, $\widehat{\mathbf{B}}^{-1} = \widehat{\mathbf{T}}^T \cdot \widehat{\mathbf{D}}^{-1} \cdot \widehat{\mathbf{T}}$

Figure 5.4: Decay of correlations in the Cholesky factors for different approximations of $\mathbf{B}^{-1}$.

We consider a two-dimensional square domain with $s \times s$ grid points. Our proof below can be extended immediately to non-square domains, as well as to three-dimensional domains. In our domain each space point is described by two indices $(i,\, j)$, a zonal component i and a meridional component j, for $1 \le i, j \le s$. A particular case for $s = 4$ is shown in Figure 5.5a. We make use of row-major order in order to map model grid components to the one dimensional "index space":

$$k = f(i,j) = (j-1)\cdot s + i, \quad \text{for } 1 \le k \le n.$$

where here, $n = s^2$. For a particular grid component $(i,\, j)$, the resulting $k = f(i,j)$ denotes the row index in $\widehat{\mathbf{B}}^{-1}$. The results of labeling each model component in this manner can be seen in Figure 5.1b.

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

(a) Grid components (i,j)

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

(b) Index space $f(i,j)$

Figure 5.5: Grid distribution of model components and corresponding index terms in $\widehat{\mathbf{B}}^{-1}$.

To start our proof, the inverse of the (exact) background error covariance matrix $\mathbf{B}^{-1}$ and of the its estimator $\widehat{\mathbf{B}}^{-1}$ can be written as

$$\widehat{\mathbf{B}}^{-1} = \left[\mathbf{I} - \widehat{\mathbf{C}}\right]^T \cdot \widehat{\mathbf{D}}^{-1} \cdot \left[\mathbf{I} - \widehat{\mathbf{C}}\right] \in \mathbb{R}^{n \times n} \tag{5.13a}$$

and

$$\mathbf{B}^{-1} = \left[\mathbf{I} - \mathbf{C}\right]^T \cdot \mathbf{D}^{-1} \cdot \left[\mathbf{I} - \mathbf{C}\right] \in \mathbb{R}^{n \times n}, \tag{5.13b}$$

respectively, where $\widehat{\mathbf{C}} = \mathbf{I} - \widehat{\mathbf{T}} \in \mathbb{R}^{n\times n}$ and $\mathbf{C} = \mathbf{I} - \mathbf{T} \in \mathbb{R}^{n\times n}$. Moreover, $\mathbf{D}$ and $\widehat{\mathbf{D}}$ are diagonal matrices:

$$\begin{aligned} \mathbf{D} &= \operatorname{diag}\left\{d_1^2,\, d_2^2,\, \dots,\, d_n^2\right\} \\ \widehat{\mathbf{D}} &= \operatorname{diag}\left\{\widehat{d}_1^2,\, \widehat{d}_2^2,\, \dots,\, \widehat{d}_n^2\right\} \end{aligned}$$

where $\{\mathbf{D}\}_{i,i} = d_i^2$ and $\left\{\widehat{\mathbf{D}}\right\}_{i,i} = \widehat{d}_i^2$, for $1 \le i \le n$. In what follows we denote by $\widehat{\mathbf{c}}^{\{j\}} \in \mathbb{R}^{n\times 1}$ and $\mathbf{c}^{\{j\}} \in \mathbb{R}^{n\times 1}$ the j-th columns of matrices $\widehat{\mathbf{C}}$ and $\mathbf{C}$, respectively, for $1 \le j \le n$.

Definition 5.3.1 (Class of matrices under consideration.)**.** We consider the class of covariance matrices matrices with correlations decreasing quickly:

$$\begin{aligned}\mathcal{U}^{-1}\left(\varepsilon_0, C, \alpha\right) \;&=\; \Bigg\{\mathbf{B} : 0 < \varepsilon_0 \leq \lambda_{min}\left(\mathbf{B}\right) \leq \lambda_{max}\left(\mathbf{B}\right) \leq \varepsilon_0^{-1}, \\ &\qquad \max_k \sum_{\ell=1}^{n} \left|\gamma_{k,\ell} \cdot \{\mathbf{T}\}_{k,\ell}\right| \leq C \cdot \zeta^{-\alpha}, \;\text{ for } \zeta \leq s-1 \Bigg\}\end{aligned} \tag{5.14a}$$

where $\mathbf{B}^{-1} = \mathbf{T}^T \mathbf{D}^{-1} \mathbf{T}$, α is the decay rate (related to the dynamics of the numerical model),

$$\gamma_{k_{(i,j)},\ell_{(p,q)}} = \begin{cases} 0 & j-\zeta \leq q \leq j-1 \text{ and } i-\zeta \leq p \leq i+\zeta \\ 0 & q = j \text{ and } i-\zeta \leq p \leq i \\ 1 & \text{otherwise} \end{cases},$$

and the grid components (i,j) and (p,q), for $1 \leq i,j,p,q \leq s$ are related to the $(k_{(i,j)}, \ell_{(p,q)})$ matrix entry by $k_{(i,j)} = f(i,j)$ and $\ell = f(p,q)$.

Theorem 4 (Error in the covariance inverse estimation)**.** *Uniformly for* $\mathbf{B} \in \mathcal{U}^{-1}\left(\varepsilon_0, C, \alpha\right)$, *if* $\zeta \approx \left[N^{-1} \cdot \log n\right]^{-1/2(\alpha+1)}$ *and* $N^{-1} \cdot \log n = o(1)$,

$$\left\|\widehat{\mathbf{B}}^{-1} - \mathbf{B}^{-1}\right\|_{\infty} = \mathcal{O}\left(\left[\frac{\log(n)}{N}\right]^{\alpha(\alpha+1)/2}\right) \tag{5.14b}$$

where $\|\cdot\|_{\infty}$ *denotes the infinity norm (matrix or vector)*

In order to prove Theorem 4, we need the following result.

Lemma 5. *Under the conditions of Theorem 4, uniformly on* $\mathcal{U}^{-1}$

$$\max\left\{\left\|\widehat{\mathbf{c}}^{\{j\}} - \mathbf{c}^{\{j\}}\right\|_{\infty} : 1 \leq j \leq n\right\} = \mathcal{O}\left(N^{-1/2} \log^{1/2} n\right), \tag{5.15a}$$

$$\max\left\{\left|\widehat{d}_j^2 - d_j^2\right| : 1 \leq j \leq n\right\} = \mathcal{O}\left(\left[N^{-1} \log n\right]^{\alpha/(2(\alpha+1))}\right), \tag{5.15b}$$

and

$$\left\|\mathbf{C}\right\|_{\infty} = \left\|\mathbf{D}^{-1}\right\|_{\infty} = \mathcal{O}\left(1\right). \tag{5.15c}$$

The proof of Lemma 5 is based on the following results of Bickel and Levina in [BL08].

Lemma 6. *[[BL08, Lemma A.2]] Let* $\boldsymbol{\nu}^{[k]} \sim \mathcal{N}(\mathbf{0}, \mathbf{B})$ *and* $\lambda_{\max}(\mathbf{B}) \leq \varepsilon_0^{-1} < \infty$, *for* $1 \leq k \leq N$. *Then, if* $\{\mathbf{B}\}_{i,j}$ *denotes the* $(i,\, j)$*-th component of* $\mathbf{B}$, *for* $1 \leq i \leq j \leq n$,

$$\begin{aligned}\text{Prob}\left[\sum_{k=1}^{N}\left[\left\{\boldsymbol{\nu}^{[k]}\right\}_i \cdot \left\{\boldsymbol{\nu}^{[k]}\right\}_j - \{\mathbf{B}\}_{i,j}\right] \geq N \cdot \nu\right] \\ \leq C_1 \cdot \exp\left(-C_2 \cdot N \cdot \nu^2\right),\end{aligned} \tag{5.16}$$

for $|\nu| \leq \delta$, *where* $\left\{\boldsymbol{\nu}^{[k]}\right\}_i$ *is the* i*-th component of the sample* $\boldsymbol{\nu}^{[k]}$, *for* $1 \leq k \leq N$, *and* $1 \leq i \leq n$. *Likewise,* C_1, C_2 *and* δ *depend on* ε_0 *only.*

Proof of Lemma 5. In what follows we denote by $\mathbf{var}$ and $\widehat{\mathbf{var}}$ denote the true and the empirical covariances, respectively. In the context of EnKF we have that $\mathbf{var}\left(\mathbf{U}^b\right) = \mathbf{B}$.

Recall that

$$\widehat{\mathbf{var}}\left(\mathbf{U}^b\right) = \mathbf{P}^b = \frac{1}{N-1} \cdot \mathbf{U}^b \cdot \mathbf{U}^{bT} = \frac{1}{N-1} \cdot \sum_{k=1}^{N} \mathbf{u}^{b[k]} \cdot \mathbf{u}^{b[k]T},$$

and therefore

$$\left\{\widehat{\mathbf{var}}\left(\mathbf{U}^b\right)\right\}_{i,j} = \frac{1}{N-1} \cdot \sum_{k=1}^{N} \left\{\mathbf{u}^{b[k]}\right\}_i \cdot \left\{\mathbf{u}^{b[k]}\right\}_j.$$

For $\nu > 0$, $\left\{\boldsymbol{\nu}^{[k]}\right\}_i \cdot \left\{\boldsymbol{\nu}^{[k]}\right\}_j - \{\mathbf{B}\}_{i,j} \geq N \cdot \nu$ implies $\left\{\boldsymbol{\nu}^{[k]}\right\}_i \cdot \left\{\boldsymbol{\nu}^{[k]}\right\}_j - \{\mathbf{B}\}_{i,j} \geq (N-1) \cdot \nu$, and therefore by Lemma 6 we have:

$$\left\|\mathbf{var}\left(\mathbf{U}^b\right) - \widehat{\mathbf{var}}\left(\mathbf{U}^b\right)\right\|_\infty = \mathcal{O}\left(N^{-1/2} \cdot \log^{1/2} n\right), \tag{5.17a}$$

since the entries of $\mathbf{var}\left(\mathbf{U}^b\right) - \widehat{\mathbf{var}}\left(\mathbf{U}^b\right)$ can be bounded by:

$$\left|\left\{\mathbf{var}\left(\mathbf{U}^b\right) - \widehat{\mathbf{var}}\left(\mathbf{U}^b\right)\right\}_{i,j}\right| \;\leq\; N^{-1} \cdot \sum_{k=1}^{N} \left|\left\{\mathbf{u}^{b[k]}\right\}_i \cdot \left\{\mathbf{u}^{b[k]}\right\}_j - \{\mathbf{B}\}_{i,j}\right|.$$

Lemma 6 ensures that:

$$\begin{aligned}\text{Prob}\left[\max_{i,j}\left|N^{-1} \cdot \sum_{k=1}^{N} \left\{\mathbf{u}^{b[k]}\right\}_i \cdot \left\{\mathbf{u}^{b[k]}\right\}_j - \{\mathbf{B}\}_{i,j}\right| \geq \nu\right] \\ \leq C_1 \cdot n^2 \cdot \exp\left(-C_2 \cdot N \cdot \nu^2\right),\end{aligned}$$

for $|\nu| \leq \delta$. Let $\nu = \left(\frac{\log n^2}{N \cdot C_2}\right)^{1/2} \cdot M$, for M arbitrary.

Since $\mathbf{Z}_{[i]}$ stores the columns of $\mathbf{U}^b$ corresponding to the predecessors of model component i, an immediate consequence of (5.17a) is

$$\max_i \left\|\mathbf{var}\left(\mathbf{Z}_{[i]}\right) - \widehat{\mathbf{var}}\left(\mathbf{Z}_{[i]}\right)\right\|_\infty = \mathcal{O}\left(N^{-1/2} \cdot \log^{1/2} n\right) . \tag{5.17b}$$

Also,

$$\left\|\mathbf{B}^{-1}\right\|_\infty = \left\|\mathbf{var}\left(\mathbf{U}^b\right)^{-1}\right\|_\infty \leq \varepsilon_0^{-1} .$$

According to equation (5.7),

$$\begin{aligned}
\left\{\mathbf{c}^{[i]}\right\}_j &= \left\{\mathbf{var}\left(\mathbf{Z}_{[i]}\right)^{-1} \cdot \mathbf{Z}_{[i]} \cdot \mathbf{x}_{[i]}\right\}_j , \\
\left\{\widehat{\mathbf{c}}^{[i]}\right\}_j &= \left\{\widehat{\mathbf{var}}\left(\mathbf{Z}_{[i]}\right)^{-1} \cdot \mathbf{Z}_{[i]} \cdot \mathbf{x}_{[i]}\right\}_j ,
\end{aligned}$$

therefore:

$$\begin{aligned}
& \max_k \left|\left\{\mathbf{c}^{[i]}\right\}_k - \left\{\widehat{\mathbf{c}}^{[i]}\right\}_k\right| \\
= \; & \max_k \left|\left\{\mathbf{var}\left(\mathbf{Z}_{[i]}\right)^{-1} \cdot \mathbf{Z}_{[i]} \cdot \mathbf{x}_{[i]}\right\}_k - \left\{\widehat{\mathbf{var}}\left(\mathbf{Z}_{[i]}\right)^{-1} \cdot \mathbf{Z}_{[i]} \cdot \mathbf{x}_{[i]}\right\}_k\right| \qquad (5.18) \\
= \; & \max_k \left|\left\{\left[\mathbf{var}\left(\mathbf{Z}_{[i]}\right)^{-1} - \widehat{\mathbf{var}}\left(\mathbf{Z}_{[i]}\right)^{-1}\right] \cdot \mathbf{Z}_{[i]} \cdot \mathbf{x}_{[i]}\right\}_k\right| \\
= \; & \mathcal{O}\left(N^{-1/2} \cdot \log^{1/2} n\right) \qquad (5.19)
\end{aligned}$$

from which (5.15a) follows. Note that:

$$\begin{aligned}
& \mathbf{x}_{[i]} = \sum_{j=1}^{n} \tilde{\gamma}_{i,j} \cdot \left\{\widehat{\mathbf{c}}^{[i]}\right\}_j \cdot \mathbf{x}_{[j]} + \widehat{\boldsymbol{\varepsilon}}^{[i]} \\
\Leftrightarrow \; & \widehat{\mathbf{var}}\left(\mathbf{x}_{[i]}\right) = \widehat{\mathbf{var}}\left(\sum_{j=1}^{n} \tilde{\gamma}_{i,j} \cdot \left\{\widehat{\mathbf{c}}^{[i]}\right\}_j \cdot \mathbf{x}_{[j]} + \widehat{\boldsymbol{\varepsilon}}^{[i]}\right) \\
\Leftrightarrow \; & \widehat{\mathbf{var}}\left(\mathbf{x}_{[i]}\right) = \widehat{\mathbf{var}}\left(\sum_{j=1}^{n} \tilde{\gamma}_{i,j} \cdot \left\{\widehat{\mathbf{c}}^{[i]}\right\}_j \cdot \mathbf{x}_{[j]}\right) + \widehat{\mathbf{var}}\left(\widehat{\boldsymbol{\varepsilon}}^{[i]}\right) \\
\Leftrightarrow \; & \widehat{d}_i^2 = \widehat{\mathbf{var}}\left(\mathbf{x}_{[i]}\right) - \widehat{\mathbf{var}}\left(\sum_{j=1}^{n} \tilde{\gamma}_{i,j} \cdot \left\{\widehat{\mathbf{c}}^{[i]}\right\}_j \cdot \mathbf{x}_{[j]}\right) ,
\end{aligned}$$

and similarly

$$d_i^2 = \mathbf{var}\left(\mathbf{x}_{[i]}\right) - \mathbf{var}\left(\sum_{j=1}^{n} \tilde{\gamma}_{i,j} \cdot \left\{\mathbf{c}^{[i]}\right\}_j \cdot \mathbf{x}_{[j]}\right) .$$

The claim (5.15b) and the first part of (5.15c) follow from (5.17a), (5.17b) and (5.19). Since

$$\begin{aligned} \left|\widehat{d_i^2} - d_i^2\right| &\leq \left|\mathbf{var}\left(\mathbf{x}_{[i]}\right) - \widehat{\mathbf{var}}\left(\mathbf{x}_{[i]}\right)\right| \\ &+ \left|\widehat{\mathbf{var}}\left(\sum_{j=1}^{n} \tilde{\gamma}_{i,j} \cdot \left[\left\{\widehat{\mathbf{c}}^{[i]}\right\}_j - \left\{\mathbf{c}^{[i]}\right\}_j\right] \cdot \mathbf{x}_{[j]}\right)\right| \\ &+ \left|\widehat{\mathbf{var}}\left(\sum_{j=1}^{n} \tilde{\gamma}_{i,j} \cdot \left\{\widehat{\mathbf{c}}^{[i]}\right\}_j \cdot \mathbf{x}_{[j]}\right) - \mathbf{var}\left(\sum_{j=1}^{n} \tilde{\gamma}_{i,j} \cdot \left\{\widehat{\mathbf{c}}^{[i]}\right\}_j \cdot \mathbf{x}_{[j]}\right)\right| \end{aligned} \tag{5.20}$$

where $\tilde{\gamma}_{i,j} = 1 - \gamma_{i,j}$. By Lemma 6 the maximum over i of the first term is:

$$\max_i \left|\mathbf{var}\left(\mathbf{x}_{[i]}\right) - \widehat{\mathbf{var}}\left(\mathbf{x}_{[i]}\right)\right| = \mathcal{O}\left(N^{-1/2} \cdot \log^{1/2} n\right).$$

The second term can be bounded as follows:

$$\begin{aligned} &\left|\sum_{j=1}^{n} \tilde{\gamma}_{i,j}^2 \cdot \left[\left\{\widehat{\mathbf{c}}^{[i]}\right\}_j - \left\{\mathbf{c}^{[i]}\right\}_j\right]^2 \cdot \widehat{\mathbf{var}}\left(\mathbf{x}_{[j]}\right)\right| \\ \leq\ & \sum_{j=1}^{n} \tilde{\gamma}_{i,j}^2 \cdot \left[\left\{\widehat{\mathbf{c}}^{[i]}\right\}_j - \left\{\mathbf{c}^{[i]}\right\}_j\right]^2 \cdot \left|\widehat{\mathbf{var}}\left(\mathbf{x}_{[j]}\right)\right| \\ \leq\ & \max_k \left[\left\{\widehat{\mathbf{c}}^{[i]}\right\}_k - \left\{\mathbf{c}^{[i]}\right\}_k\right]^2 \cdot \max_i \left|\widehat{\mathbf{var}}\left(\mathbf{x}_{[i]}\right)\right| \cdot \sum_{j=1}^{n} \tilde{\gamma}_{i,j}^2 \\ =\ & \mathcal{O}\left(\zeta^2 \cdot N^{-1} \cdot \log n\right) \\ =\ & \mathcal{O}\left(\left[N^{-1} \cdot \log n\right]^{\alpha/2\cdot(\alpha+1)}\right) \end{aligned}$$

by (5.15a) and $\|\mathbf{B}\| \leq \varepsilon_0^{-1}$. Recall that $\zeta = \left[N^{-1} \cdot \log n\right]^{1/2\cdot(\alpha+1)}$ and even more, note that:

$$\sum_{j=1}^{n} \tilde{\gamma}_{i,j}^2 = \frac{\left[\zeta + 1\right]^2}{2} = \frac{\zeta^2}{2} + \zeta + \frac{1}{2} = \mathcal{O}\left(\zeta^2\right).$$

The third term can be bounded similarly. Thus (5.15b) follows. Furthermore,

$$d_i^2 = \mathbf{var}\left(\mathbf{x}_{[i]} - \sum_{j=1}^{n} \tilde{\gamma}_{i,j} \cdot \left\{\widehat{\mathbf{c}}^{[i]}\right\}_j \cdot \mathbf{x}_{[j]}\right) \geq \varepsilon_0 \cdot \left(1 + \sum_{i=1}^{n} \left[\widehat{\mathbf{c}}_j^{[i]}\right]^2\right) \geq \varepsilon_0\,,$$

and the lemma follows. □

We now are ready to prove Theorem 4.

Proof of Theorem 4. We need only check that:

$$\left\|\widehat{\mathbf{B}}^{-1}-\mathbf{B}^{-1}\right\|_{\infty}=\mathcal{O}\left(N^{-1/2}\cdot\log^{1/2}(n)\right) \tag{5.21a}$$

and

$$\left\|\mathbf{B}^{-1}-\Phi_{\zeta}\left(\mathbf{B}^{-1}\right)\right\|_{\infty}=\mathcal{O}\left(\zeta^{-\alpha}\right) \tag{5.21b}$$

where the entries of $\Phi_{\zeta}\left(\mathbf{B}^{-1}\right)$ are given by:

$$\left\{\Phi_{\zeta}\left(\mathbf{B}^{-1}\right)\right\}_{k,\ell}=\delta_{k,\ell}\cdot\left\{\mathbf{B}^{-1}\right\}_{k,\ell},\ \text{ for } 1\leq k,\ell\leq n \tag{5.21c}$$

where $k=f(i,j)$ and $\ell=f(q,p)$ for $1\leq i,j,p,q\leq s$, and

$$\delta_{k,\ell} \;=\; \begin{cases}1 & j-\zeta\leq q\leq j+\zeta \text{ and } i-\zeta\leq p\leq i+\zeta\\ 0 & \text{otherwise}\end{cases}$$

We first prove (5.21a). By definition,

$$\widehat{\mathbf{B}}^{-1}-\mathbf{B}^{-1}=\widehat{\mathbf{T}}^{T}\cdot\widehat{\mathbf{D}}^{-1}\cdot\widehat{\mathbf{T}}-\mathbf{T}^{T}\cdot\mathbf{D}^{-1}\cdot\mathbf{T}. \tag{5.22}$$

Applying the standard inequality:

$$\begin{aligned}
\left\|\mathbf{T}^{T}\cdot\mathbf{D}^{-1}\cdot\mathbf{T}-\widehat{\mathbf{T}}^{T}\cdot\widehat{\mathbf{D}}^{-1}\cdot\widehat{\mathbf{T}}^{T}\right\| \;\leq\;& \left\|\mathbf{T}^{T}-\widehat{\mathbf{T}}^{T}\right\|\cdot\left\|\widehat{\mathbf{D}}\right\|\cdot\left\|\widehat{\mathbf{T}}\right\|\\
+\;& \left\|\mathbf{D}-\widehat{\mathbf{D}}\right\|\cdot\left\|\widehat{\mathbf{T}}^{T}\right\|\cdot\left\|\widehat{\mathbf{T}}\right\|\\
+\;& \left\|\mathbf{T}-\widehat{\mathbf{T}}\right\|\cdot\left\|\widehat{\mathbf{T}}\right\|\cdot\left\|\widehat{\mathbf{D}}\right\|\\
+\;& \left\|\widehat{\mathbf{T}}\right\|\cdot\left\|\mathbf{D}-\widehat{\mathbf{D}}\right\|\cdot\left\|\widehat{\mathbf{T}}^{T}-\mathbf{T}^{T}\right\|\\
+\;& \left\|\widehat{\mathbf{D}}\right\|\cdot\left\|\mathbf{T}-\widehat{\mathbf{T}}\right\|\cdot\left\|\widehat{\mathbf{T}}^{T}-\mathbf{T}^{T}\right\|\\
+\;& \left\|\widehat{\mathbf{T}}^{T}\right\|\cdot\left\|\mathbf{D}-\widehat{\mathbf{D}}\right\|\cdot\left\|\widehat{\mathbf{T}}-\mathbf{T}\right\|\\
+\;& \left\|\mathbf{D}-\widehat{\mathbf{D}}\right\|\cdot\left\|\mathbf{T}-\widehat{\mathbf{T}}\right\|\cdot\left\|\mathbf{T}^{T}-\widehat{\mathbf{T}}^{T}\right\|
\end{aligned}$$

all previous terms can be bounded making use of Lemma 5 and therefore, (5.21a) follows. Likewise, for (5.21b), we need to note that for any matrix $\mathbf{M}$,

$$\begin{aligned}
\left\|\mathbf{M}\cdot\mathbf{M}^{T}-\Phi_{\zeta}(\mathbf{M})\cdot\Phi_{\zeta}(\mathbf{M})^{T}\right\|_{\infty} \;\leq\;& 2\cdot\|\mathbf{M}\|_{\infty}\cdot\left\|\Phi_{\zeta}(\mathbf{M})-\mathbf{M}^{-1}\right\|_{\infty}\\
+\;& \left\|\Phi_{\zeta}(\mathbf{M})-\mathbf{M}\right\|_{\infty}^{2}
\end{aligned}$$

and by letting $\mathbf{M}=\mathbf{T}^{T}\cdot\mathbf{D}^{-1/2}$, the theorem follows from Definition 5.3.1. ☐

5.4 Numerical Experiments

In this section we study the performance of the proposed EnKF-MC implementation. The experiments are performed using the atmospheric general circulation model SPEEDY [Mol03b, KMB06a]. SPEEDY is a hydrostatic, spectral coordinate, spectral transform model in the vorticity-divergence form, with semi-implicit treatment of gravity waves. The number of layers in the SPEEDY model is 8 and the T-63 model resolution (192×96 grids) is used for the horizontal space discretization of each layer. Four model variables are part of the assimilation process: the temperature (K), the zonal and the meridional wind components (m/s), and the specific humidity (g/kg). The total number of model components is $n = 589,824$. The number of ensemble members is $N = 94$ for all the scenarios. The model state space is approximately 6,274 times larger than the number of ensemble members ($n \gg N$). The tests are performed on the super computer Blueridge cluster at the university of Virginia Tech. BlueRidge is a 408-node Cray CS-300 cluster. Each node is outfitted with two octa-core Intel Sandy Bridge CPUs and 64 GB of memory, for a total of 6528 cores and 27.3 TB of memory systemwide.

Starting with the state of the system $\mathbf{x}^{\text{ref}}_{-3}$ at time t_{-3}, the model solution $\mathbf{x}^{\text{ref}}_{-3}$ is propagated in time over one year:

$$\mathbf{x}^{\text{ref}}_{-2} = \mathcal{M}_{t_{-3} \rightarrow t_{-2}} \left(\mathbf{x}^{\text{ref}}_{-3} \right) .$$

The reference solution $\mathbf{x}^{\text{ref}}_{-2}$ is used to build a perturbed background solution:

$$\widehat{\mathbf{x}}^{b}_{-2} = \mathbf{x}^{\text{ref}}_{-2} + \boldsymbol{\epsilon}^{b}_{-2}, \quad \boldsymbol{\epsilon}^{b}_{-2} \sim \mathcal{N} \left(\mathbf{0}_n, \, \operatorname*{diag}_{i} \left\{ (0.05 \, \{\mathbf{x}^{\text{ref}}_{-2}\}_i)^2 \right\} \right) . \tag{5.23}$$

The perturbed background solution is propagated over another year to obtain the background solution at time t_{-1}:

$$\mathbf{x}^{b}_{-1} = \mathcal{M}_{t_{-2} \rightarrow t_{-1}} \left(\widehat{\mathbf{x}}^{b}_{-2} \right) . \tag{5.24}$$

This model propagation attenuates the random noise introduced in (5.23) and makes the background state (5.24) consistent with the physics of the SPEEDY model. Then, the background state (5.24) is utilized in order to build an ensemble of perturbed background states:

$$\widehat{\mathbf{x}}^{b[i]}_{-1} = \mathbf{x}^{b}_{-1} + \boldsymbol{\epsilon}^{b}_{-1}, \quad \boldsymbol{\epsilon}^{b}_{-1} \sim \mathcal{N} \left(\mathbf{0}_n, \, \operatorname*{diag}_{i} \left\{ (0.05 \, \{\mathbf{x}^{b}_{-1}\}_i)^2 \right\} \right), \quad 1 \le i \le N, \tag{5.25}$$

from which, after three months of model propagation, the initial ensemble is obtained at time t_0:

$$\mathbf{x}^{b[i]}_{0} = \mathcal{M}_{t_{-1} \rightarrow t_0} \left(\widehat{\mathbf{x}}^{b[i]}_{-1} \right) .$$

Again, the model propagation of the perturbed ensemble ensures that the ensemble members are consistent with the physics of the numerical model.

The experiments are performed over a period of 24 days, where observations are taken every 2 days ($M = 12$). At time k synthetic observations are built as follows:

$$\mathbf{y}_k = \mathbf{H}_k \cdot \mathbf{x}_k^{\text{ref}} + \boldsymbol{\epsilon}_k, \quad \boldsymbol{\epsilon}_k \sim \mathcal{N}\left(\mathbf{0}_m,\, \mathbf{R}_k\right), \quad \mathbf{R}_k = \text{diag}_i \left\{ (0.01\, \{\mathbf{H}_k\, \mathbf{x}_k^{\text{ref}}\}_i)^2 \right\}.$$

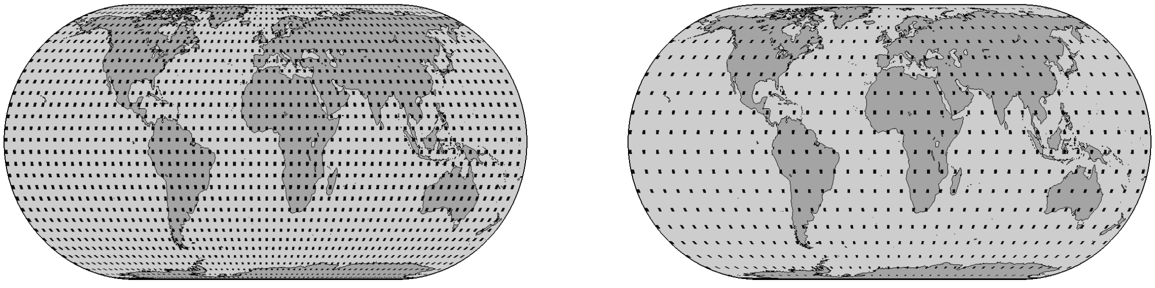
The observation operators $\mathbf{H}_k$ are fixed throughout the time interval. We perform experiments with several operators characterized by different proportions p of observed components from the model state $\mathbf{x}_k^{\text{ref}}$ ($m \approx p \cdot n$). We consider four different values for p: 0.50, 0.12, 0.06 and 0.04 which represent 50%, 12 %, 6 % and 4 % of the total number of model components, respectively. Some of the observational networks used during the experiments are shown in Figure 5.6 with their corresponding percentage of observed components from the model state.

The analyses of the EnKF-MC are compared against those obtained making use of the LETKF implementation proposed by Hunt et al in [OHS+04b, OHS+04a, OHS+08] . The analysis accuracy is measured by the root mean square error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{M} \cdot \sum_{k=1}^{M} \left[\mathbf{x}_k^{\text{ref}} - \mathbf{x}_k^{\text{a}}\right]^T \cdot \left[\mathbf{x}_k^{\text{ref}} - \mathbf{x}_k^{\text{a}}\right]} \tag{5.26}$$

where $\mathbf{x}^{\text{ref}} \in \mathbb{R}^{n\times 1}$ and $\mathbf{x}_k^{\text{a}} \in \mathbb{R}^{n\times 1}$ are the reference and the analysis solutions at time k, respectively, and M is the number of assimilation times.

The threshold used in (5.9) during the computation of $\widehat{\mathbf{B}}^{-1}$ is $\sigma_r = 0.10$. During the assimilation steps, the data error covariance matrices $\mathbf{R}_k$ are used (no representativeness errors are involved during the assimilations) and therefore. The different EnKF implementations are performed making use of FORTRAN and specialized libraries such as BLAS and LAPACK are used in order to perform the algebraic computations.



(a) $p = 12\%$

(b) $p = 4\%$

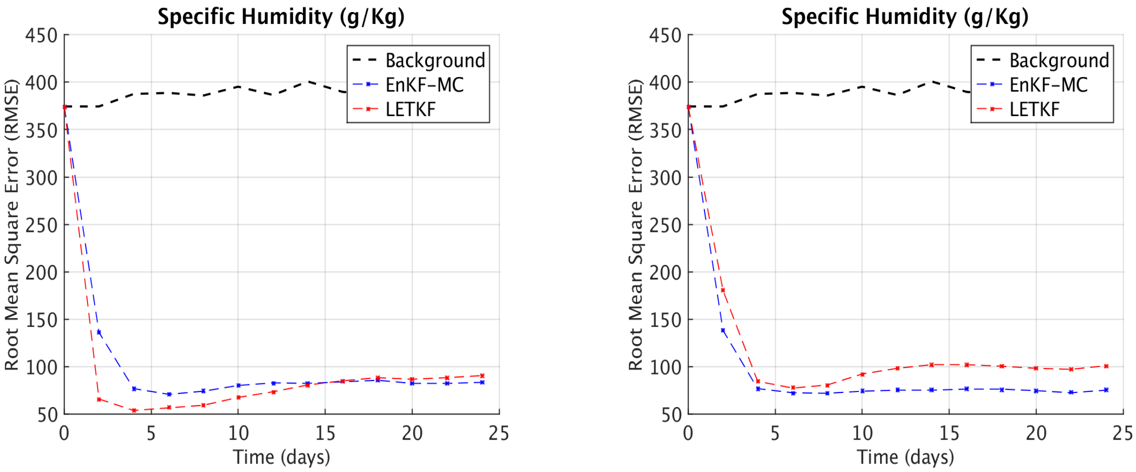
Figure 5.6: Observational networks for different values of p. Dark dots denote the location of the observed components. The observed model variables are the zonal and the meridional wind components, the specific humidity, and the temperature.

5.4.1 Results with dense observation networks

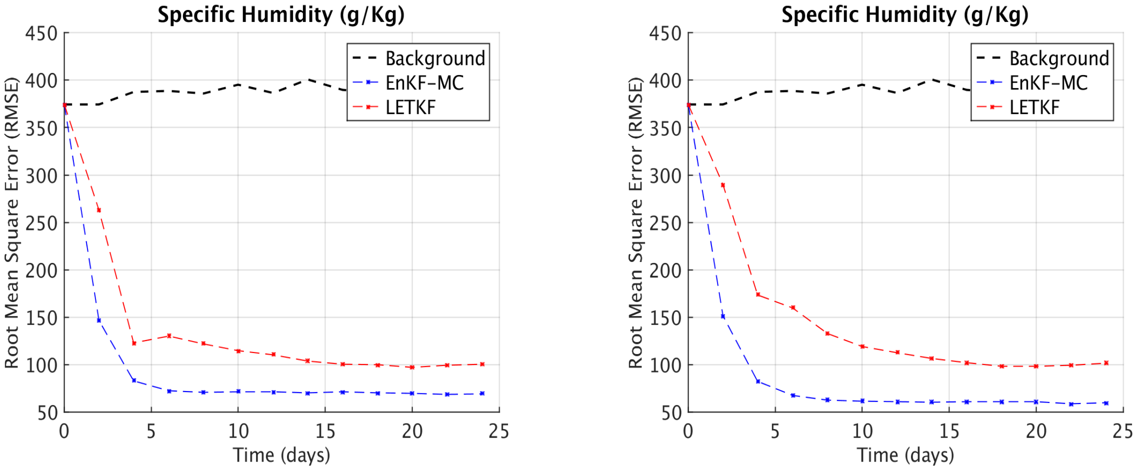
We first consider dense observational networks in which 100% and 50% of the model components are observed. We vary the radius of influence ζ from 1 to 5 grid points.

Figure 5.7 shows the RMSE values for the LETKF and EnKF-MC analyses for different values of ζ for the specific humidity when 50% of model components are observed. When the radius of influence is increased the quality of the LETKF results degrades due to spurious correlations. This is expected since the local estimation of correlations in the context of LETKF is the sample covariance matrix. For instance, for a radius of influence of 1, the total number of local components for each local box is 36 which matches the dimension of the local background error distribution. Now, when we compare it against the ensemble size (96 ensemble members), sufficient degrees of freedom (95 degrees of freedom) are available in order to estimate the local background error distribution onto the ensemble space, and consequently all directions of the local probability error distribution are accounted during the estimation and posterior assimilation. On the other hand, when the radius of influence is 5, the local box sizes have dimension 484 (model components) which is approximately 5 times larger than the ensemble size. Thus, when the analysis increments are computed onto the ensemble space, just part of the local background error distribution is accounted during the assimilation. Consequently, the larger the local box, the more local background error information cannot be represented in the ensemble space.

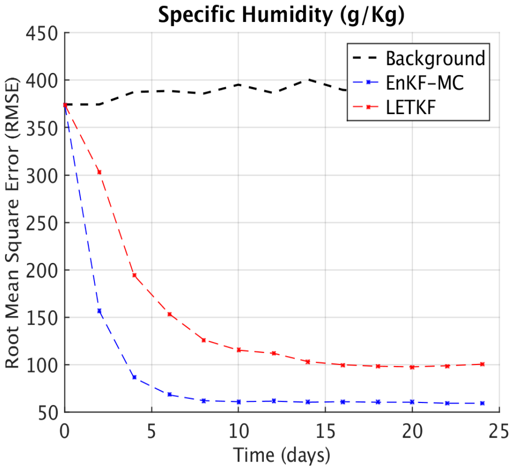
Figure 5.7 shows that EnKF-MC analyses improve with increasing radius of influence ζ. Since a dense observational network is considered during the assimilation, when the radius of influence is increased, a better estimation of the state of the system is obtained by the EnKF-MC. This can be seen clearly in Figure 5.8, where the RMSE values within the assimilation window are shown for the LETKF and the EnKF-MC solutions for the specific humidity variable and different values of ζ and p. The quality of the EnKF-MC analysis for $\zeta = 5$ is better than that of the LETKF with $\zeta = 1$. Likewise, when a full observational network is considered ($p = 100\%$), the proposed implementation outperforms the LETKF implementation. EnKF-MC is able to exploit the large amount of information contained in dense observational networks by properly estimating the local background error correlations. The RMSE values for all model variables and different values for ζ and p are summarized in Table 5.1.



(a) $\zeta = 1$. (b) $\zeta = 2$.

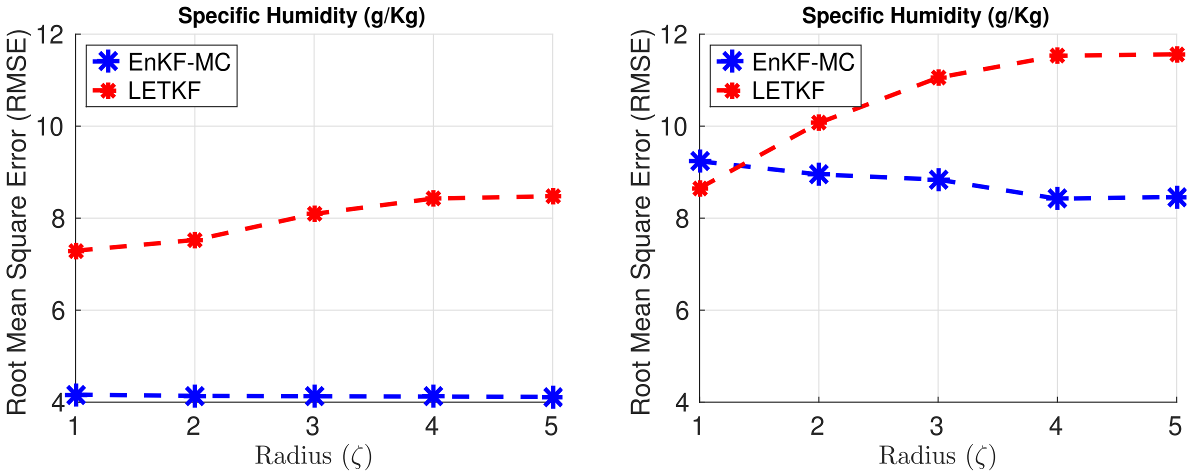


(c) $\zeta = 3$. (d) $\zeta = 4$.



(e) $\zeta = 5$.

Figure 5.7: RMSE of specific humidity analyses with a dense observational network. When the radius of influence ζ is increased the performance of LETKF degrades.



(a) $p = 100\%$.

(b) $p = 50\%$.

Figure 5.8: Analysis RMSE for the specific humidity variable. The RMSE values of the assimilation window are shown for different values of ζ and percentage of observed components p. When the local domain sizes are increased the accuracy of the LETKF analysis degrades, while the accuracy of EnKF-MC analysis improves.

Table 5.1: RMSE values for the EnKF-MC and the LETKF analyses with the SPEEDY model and for different values for ζ and p. Dense observational networks are considered in this experimental setting.

Variable (units)	ζ	p	EnKF-MC	LETKF
Zonal Wind Component (u), (m/s)	1	100%	6.012×10^1	6.394×10^1
		50%	4.264×10^2	9.825×10^2
	2	100%	6.078×10^1	6.820×10^1
		50%	2.255×10^2	1.330×10^3
	3	100%	6.080×10^1	7.969×10^1
		50%	2.341×10^2	1.124×10^3
	4	100%	6.088×10^1	9.687×10^1
		50%	2.418×10^2	1.072×10^3
	5	100%	6.092×10^1	1.190×10^2
		50%	2.673×10^2	1.017×10^3
Meridional Wind Component (v) (m/s)	1	100%	3.031×10^1	6.418×10^1
		50%	2.632×10^2	3.247×10^2
	2	100%	3.046×10^1	6.597×10^1
		50%	1.641×10^2	4.138×10^2
	3	100%	3.047×10^1	7.565×10^1
		50%	1.964×10^2	4.418×10^2
	4	100%	3.052×10^1	9.332×10^1
		50%	2.084×10^2	4.832×10^2
	5	100%	3.054×10^1	1.151×10^2
		50%	2.428×10^2	5.029×10^2
Temperature (K)	1	100%	9.404×10^2	5.078×10^2
		50%	6.644×10^2	7.059×10^2
	2	100%	9.416×10^2	4.112×10^2
		50%	6.129×10^2	1.138×10^3
	3	100%	9.425×10^2	3.447×10^2
		50%	5.815×10^2	1.389×10^3
	4	100%	9.432×10^2	2.939×10^2
		50%	5.585×10^2	1.355×10^3
	5	100%	9.432×10^2	2.554×10^2
		50%	5.500×10^2	1.104×10^3
Specific Humidity (g/Kg)	1	100%	1.733×10^1	5.427×10^1
		50%	8.680×10^1	7.602×10^1
	2	100%	1.712×10^1	5.669×10^1
		50%	8.204×10^1	1.045×10^2
	3	100%	1.705×10^1	6.630×10^1
		50%	8.089×10^1	1.298×10^2
	4	100%	1.699×10^1	7.344×10^1
		50%	7.525×10^1	1.431×10^2
	5	100%	1.694×10^1	7.617×10^1
		50%	7.642×10^1	1.458×10^2

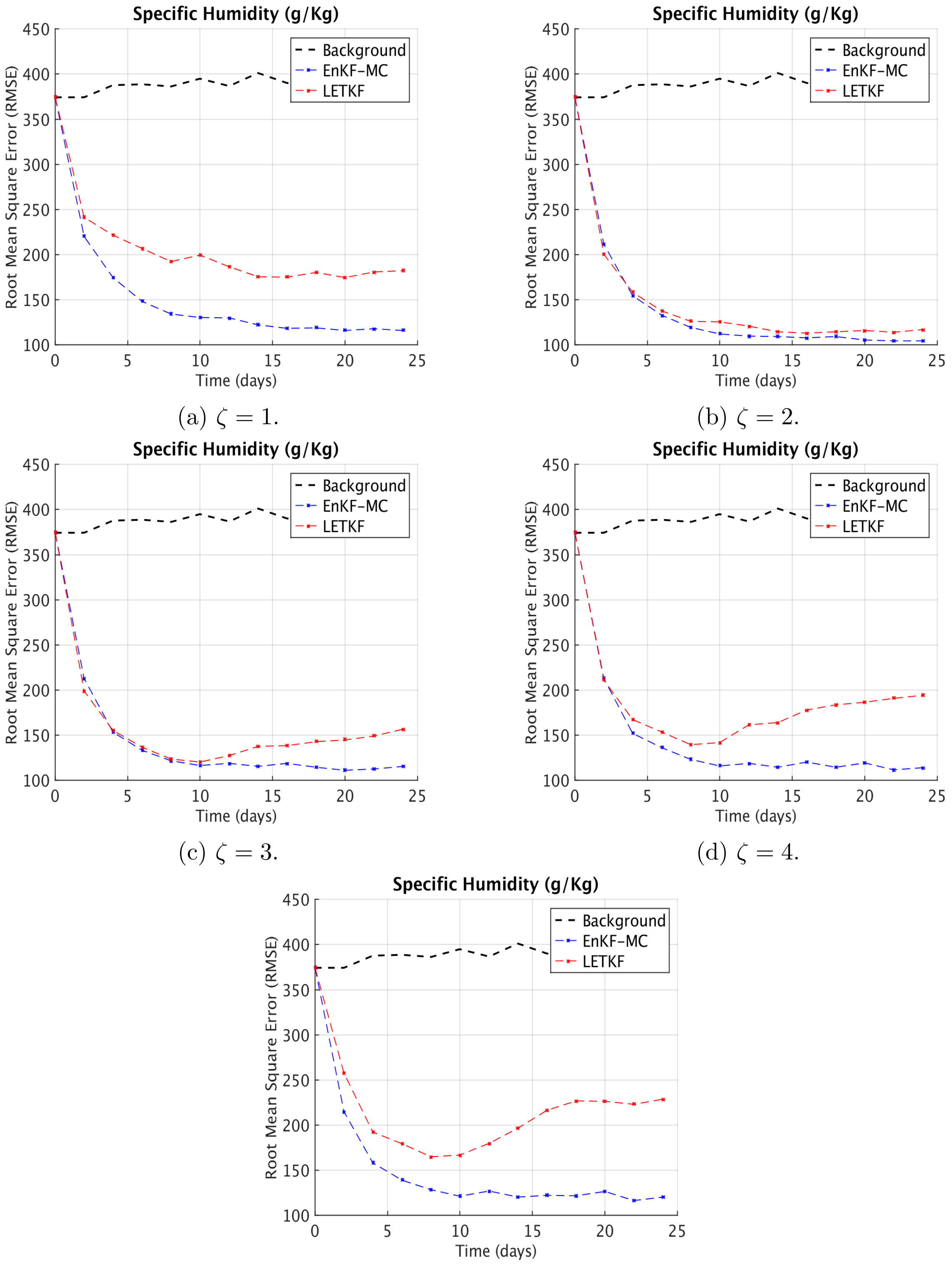
5.4.2 Results with sparse observation networks

For sparse observational networks, in general, the results obtained by the EnKF-MC are more accurate than those obtained by the LETKF, as reported in the Tables 5.2 and 5.3. We vary the values of ζ from 1 to 5. Three sparse observational networks with $p = 12\%$, 6%, and 4%, respectively are considered.

Figure 5.9 shows the RMSE values of the specific humidity analyses for different radii of influence and 4% of the model components being observed. The best performance of the LETKF analyses is obtained when the radius of influence is set to 2. Note that for $\zeta = 1$ the LETKF performs poorly, which is expected since during the assimilation most of model components will not have observations in their local boxes. For $\zeta \geq 3$ the effects of spurious correlations degrade the quality of the LETKF analysis. On the other hand, the background error correlations estimated by the modified Cholesky decomposition allows the EnKF-MC formulation to obtain good analyses even for largest radius of influence $\zeta = 5$.

Figure 5.10 shows the RMSE values of the LETKF and the EnKF-MC implementations for different radii of influences and two sparse observational networks. Clearly, when the radius of influence is increased, in the LETKF context, the analysis corrections are impacted by spurious correlations. On the other hand, the quality of the results in the EnKF-MC case is considerably better. When data errors components are uncorrelated ζ can be seen as a free parameter and the choice can be based on the “optimal performance of the filter”. For the largest radius of influence $\zeta = 5$ the RMSE values of the ENKF-MC and the LETKF implementations differ by one order of magnitude.

Figure 5.11 reports the RMSE values for the zonal and the meridional wind component analyses, and for different values of p and ζ. As can be seen, the estimation of background errors via $\widehat{\mathbf{B}}$ can reduce the impact of spurious correlations; the RMSE values of the EnKF-MC analyses remain small at all assimilation times, from which we infer that the background error correlations are properly estimated. On the other hand, the impact of spurious correlations is evident in the context of LETKF. Since most of the model components are unobserved, the background error correlations drive the quality of the analysis, and spurious correlations lead to a poor performance of the filter at many assimilation times.



(a) $\zeta = 1$.

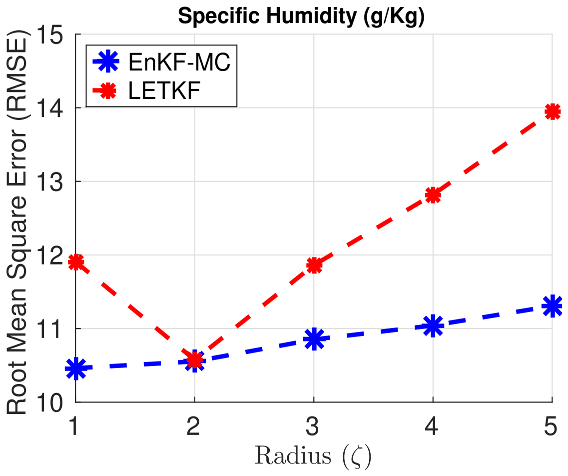
(b) $\zeta = 2$.

(c) $\zeta = 3$.

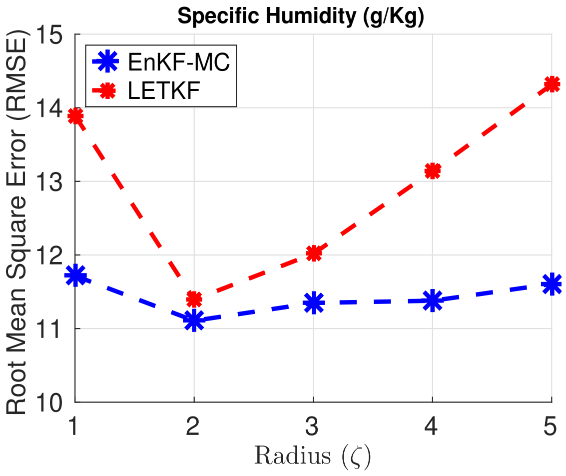
(d) $\zeta = 4$.

(e) $\zeta = 5$.

Figure 5.9: RMSE of specific humidity analyses with a sparse observational network ($p \sim 4\%$) and different values of ζ.



(a) $p = 6\%$.



(b) $p = 4\%$.

Figure 5.10: Analysis RMSE for the specific humidity variable with sparse observation networks. RMSE values are shown for different values of ζ and percentage of observed components p.

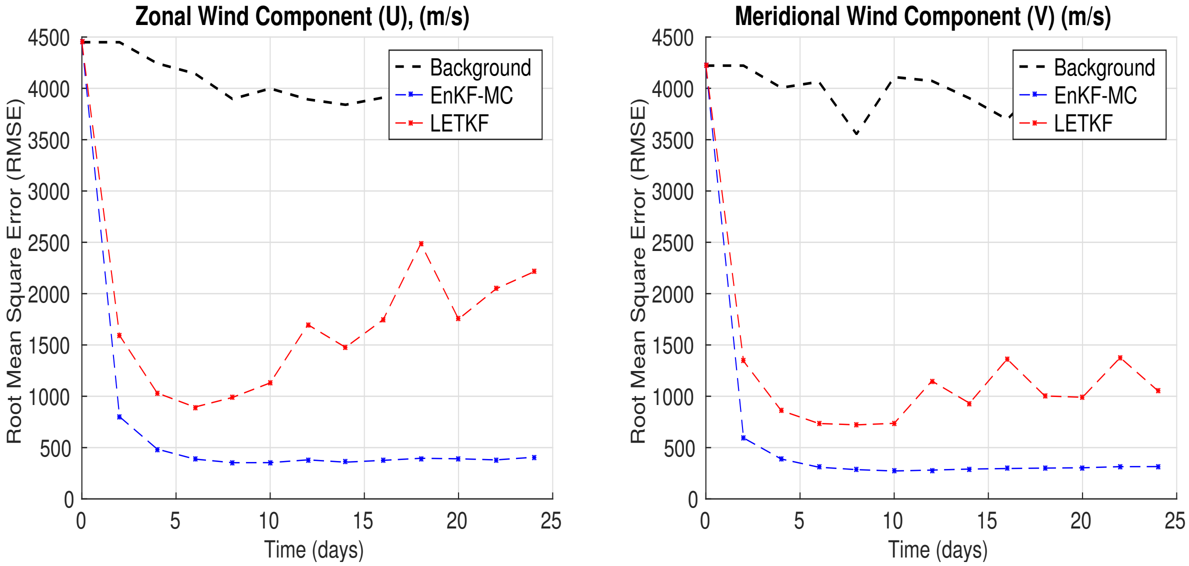
Figures 5.12 and 5.13 provide snapshots of the meridional and the zonal wind components, respectively, at the first assimilation time. For this particular case the percentage of observed model components is $p = 4\%$. At this step, only the initial observation has been assimilated in order to compute the analysis corrections by the EnKF-MC and the LETKF methods. The background solution contains erroneous waves for the zonal and the meridional wind components. For instance, for the u model variable, such waves are clearly present near the poles. After the first assimilation step, the LETKF analysis solution dissipates the erroneous waves but, the numerical values of the wind components are slightly greater than those of the reference solutions. This numerical difference increases at later times due to the highly-nonlinear dynamics of SPEEDY, as can bee seen in Figure 5.11. On the other hand, the EnKF-MC implementation recovers the reference shape, and the analysis values of the numerical model components are close to that of the reference solution. This shows again that the use of the modified Cholesky decomposition as the estimator of the background error correlations can mitigate the impact of spurious error correlations.

Table 5.2: RMSE values of the wind-components for the EnKF-MC and LETKF making use of the SPEEDY model.

Variable (units)	ζ	p	EnKF-MC	LETKF
Zonal Wind Component (u), (m/s)	1	12%	5.514×10^{2}	5.471×10^{2}
		6%	6.972×10^{2}	1.168×10^{3}
		4%	9.393×10^{2}	1.737×10^{3}
	2	12%	4.187×10^{2}	1.275×10^{3}
		6%	6.090×10^{2}	7.591×10^{2}
		4%	7.853×10^{2}	8.569×10^{2}
	3	12%	4.388×10^{2}	1.661×10^{3}
		6%	6.146×10^{2}	1.237×10^{3}
		4%	7.438×10^{2}	9.997×10^{2}
	4	12%	4.323×10^{2}	1.752×10^{3}
		6%	5.990×10^{2}	1.608×10^{3}
		4%	7.124×10^{2}	1.258×10^{3}
	5	12%	4.456×10^{2}	1.862×10^{3}
		6%	6.106×10^{2}	1.983×10^{3}
		4%	7.160×10^{2}	1.602×10^{3}
Meridional Wind Component (v) (m/s)	1	12%	3.540×10^{2}	4.496×10^{2}
		6%	5.165×10^{2}	1.158×10^{3}
		4%	7.770×10^{2}	1.749×10^{3}
	2	12%	3.009×10^{2}	7.285×10^{2}
		6%	4.605×10^{2}	5.520×10^{2}
		4%	6.217×10^{2}	7.420×10^{2}
	3	12%	3.172×10^{2}	9.510×10^{2}
		6%	4.735×10^{2}	8.334×10^{2}
		4%	6.014×10^{2}	7.455×10^{2}
	4	12%	3.399×10^{2}	1.048×10^{3}
		6%	4.812×10^{2}	1.146×10^{3}
		4%	5.913×10^{2}	9.026×10^{2}
	5	12%	3.626×10^{2}	1.101×10^{3}
		6%	5.107×10^{2}	1.575×10^{3}
		4%	6.122×10^{2}	1.102×10^{3}

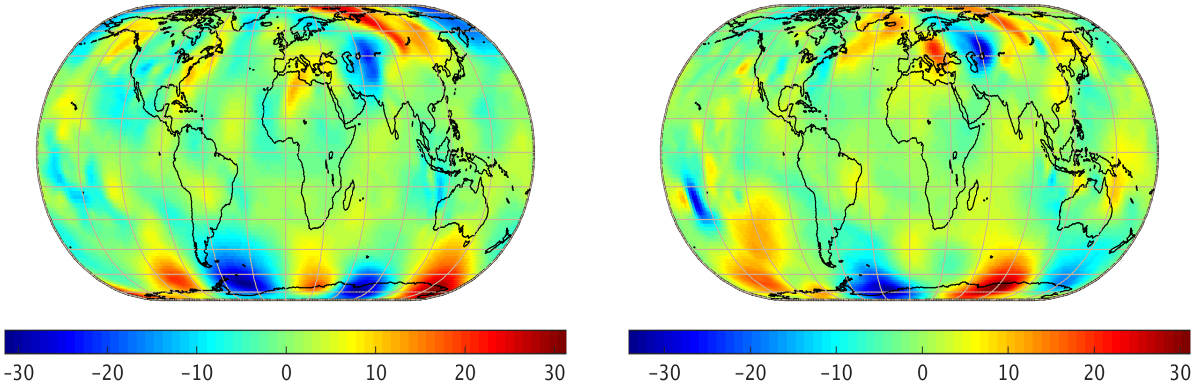
Table 5.3: RMSE values for the EnKF-MC and LETKF making use of the SPEEDY model.

Variable (units)	ζ	p	EnKF-MC	LETKF
Temperature (K)	1	12%	6.054×10^2	6.033×10^2
		6%	5.692×10^2	6.704×10^2
		4%	6.522×10^2	8.073×10^2
	2	12%	5.680×10^2	6.693×10^2
		6%	5.193×10^2	5.556×10^2
		4%	5.299×10^2	5.529×10^2
	3	12%	5.279×10^2	1.217×10^3
		6%	4.982×10^2	6.458×10^2
		4%	4.926×10^2	6.073×10^2
	4	12%	5.023×10^2	1.817×10^3
		6%	4.757×10^2	1.030×10^3
		4%	4.766×10^2	7.464×10^2
	5	12%	4.898×10^2	1.600×10^3
		6%	4.644×10^2	1.473×10^3
		4%	4.684×10^2	1.172×10^3
Specific Humidity (g/Kg)	1	12%	9.862×10^1	9.026×10^1
		6%	1.133×10^2	1.449×10^2
		4%	1.405×10^2	1.941×10^2
	2	12%	1.029×10^2	1.125×10^2
		6%	1.146×10^2	1.137×10^2
		4%	1.270×10^2	1.321×10^2
	3	12%	1.068×10^2	1.341×10^2
		6%	1.205×10^2	1.418×10^2
		4%	1.317×10^2	1.458×10^2
	4	12%	1.065×10^2	1.640×10^2
		6%	1.246×10^2	1.652×10^2
		4%	1.324×10^2	1.739×10^2
	5	12%	1.089×10^2	2.078×10^2
		6%	1.301×10^2	1.950×10^2
		4%	1.373×10^2	2.068×10^2



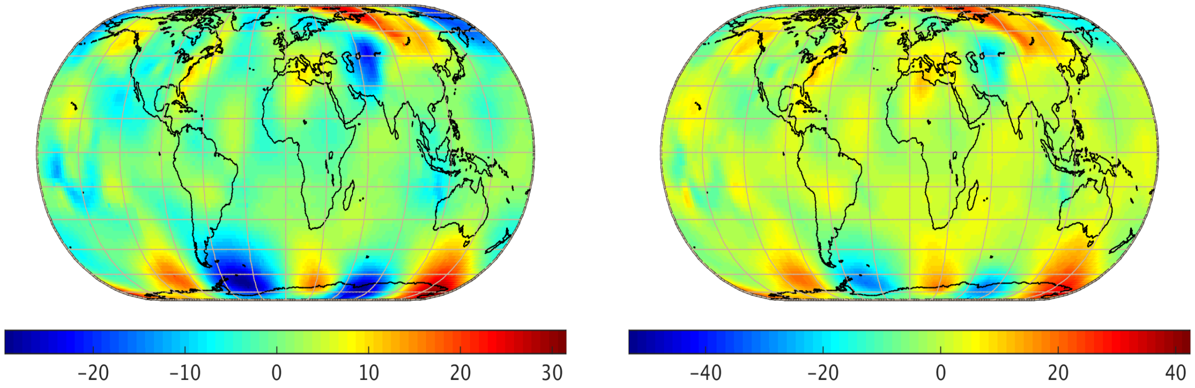
(a) $\zeta = 3$ and $p = 12\%$ (b) $\zeta = 4$ and $p = 12\%$

Figure 5.11: RMSE of the LETKF and EnKF-MC implementations for different model variables, radii of influence and observational networks.



(a) Reference

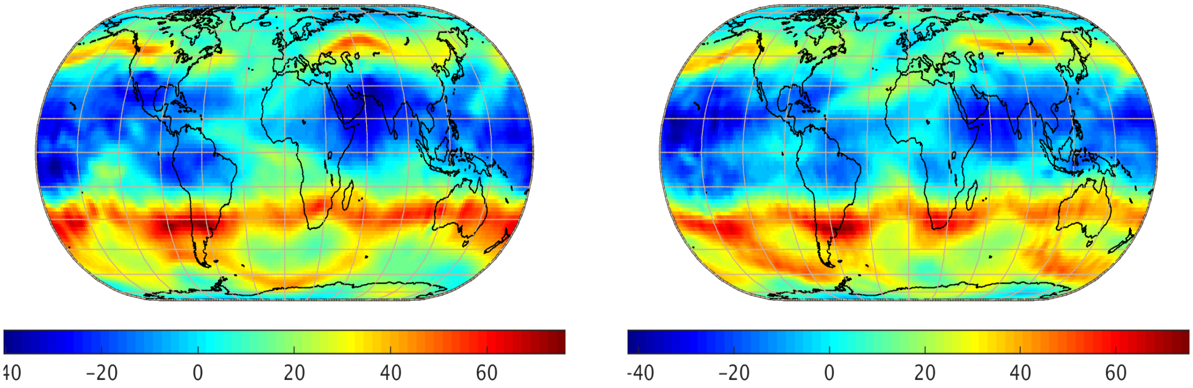
(b) Background



(c) EnKF-MC

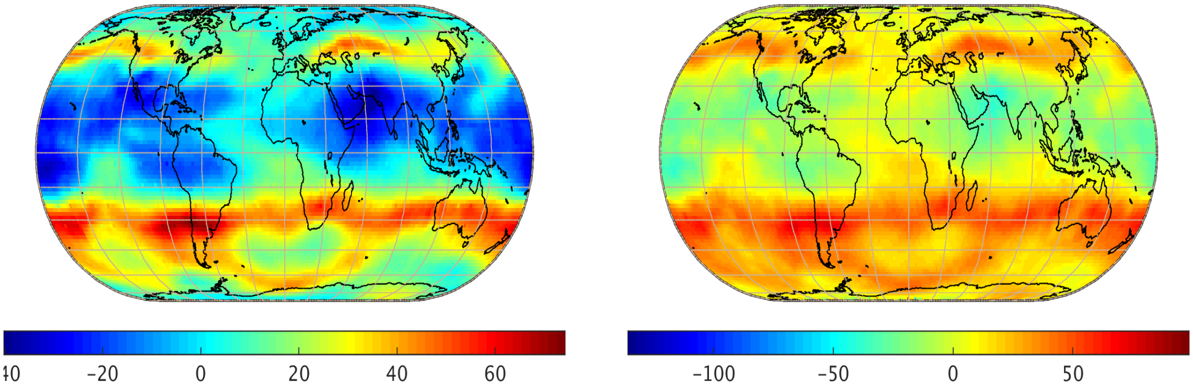
(d) LETKF

Figure 5.12: Snapshots of the reference solution, background state, and analysis fields from the EnKF-MC and LETKF for the fifth layer of the meridional wind component (v).



(a) Reference

(b) Background



(c) EnKF-MC

(d) LETKF

Figure 5.13: Snapshots of the reference solution, background state, and analysis fields from the EnKF-MC and LETKF for the second layer of the zonal wind component (u).

5.4.3 Parallel performance

We compare the elapsed times of both implementations when the number of processors (subdomains) are increased. We vary the number of computing nodes from 6 (96 processors) to 128 (2,048 processors) We let the radius of influence $r = 5$ and an observational network with $p = 4\%$. The elapsed times for different numbers of computing nodes for the EnKF-MC and LETKF are shown in figure 5.14. As is expected, the elapsed time of the LETKF is lesser than that in the EnKF-MC formulation since no covariance estimation is performed. Nevertheless, the difference between the elapsed times is in the order of seconds which is is even more irrelevant when the EnKF-MC results are more accurate than those obtained by the LETKF.

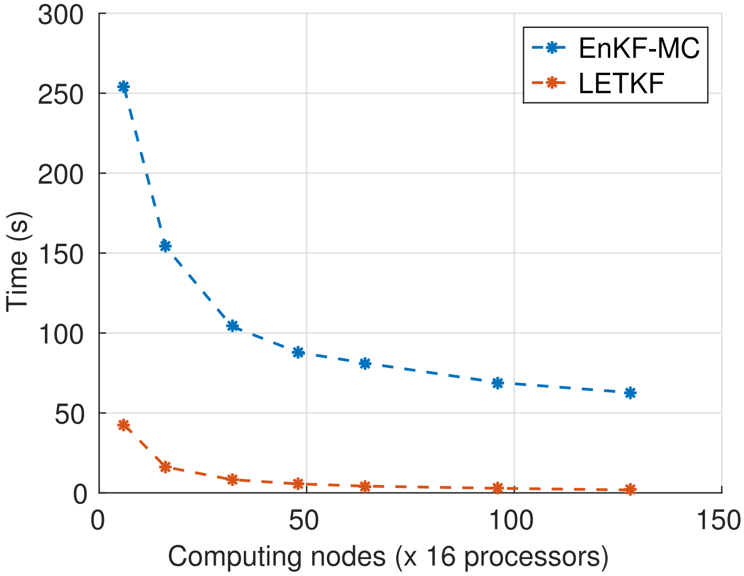
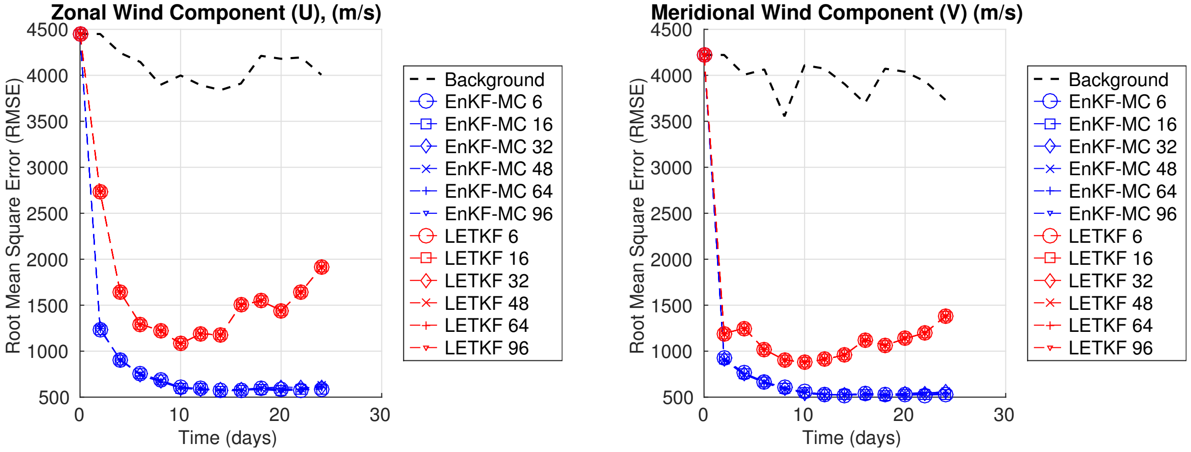
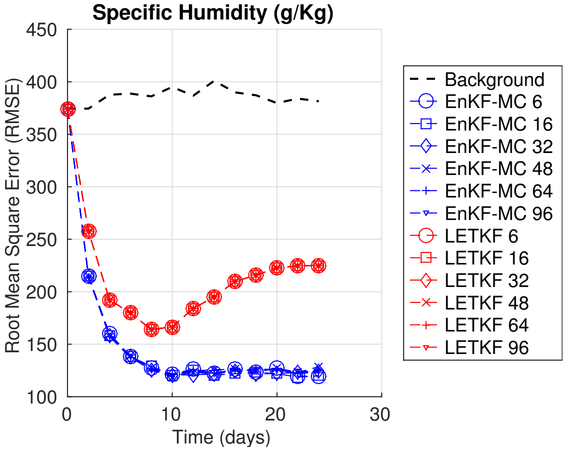


Figure 5.14: Elapsed times of the EnKF-MC and LETKF for different number of computing nodes (x 16 processors).

An important concern to address in the EnKF-MC formulation is how its accuracy is impacted when the number of processors (sub-domains) is increased. As we mentioned before, the model domain is decomposed in order to speedup computations but not for increasing the accuracy of the method (i.e., the impact of spurious correlations can be small for small sub-domain sizes) Two main reasons are that we have a well-conditioned estimated of $\mathbf{B}^{-1}$ and even more, the conditional independence of model components makes the sub-domain size to have no impact in the accuracy of the EnKF-MC. As can be seen in figure 5.15, for some model variables and values of r and p, the EnKF-MC provides almost the same accurate results among all configurations. The small variations in the RMSE values of the EnKF-MC obey to the synthetic data built at different processors during the assimilation step. For instance, the random number generators used in the experiments depends on the processors id and therefore, the exact synthetic data is not replicated when the number of processors is changed. In the LETKF context we obtain the exact same results for all configurations since it is a deterministic filter and even more, the assimilation is performed for each grid point in the sub-domain. The relations between the accuracy of the methods and the radii CPU-time for 96 and 768 processors are shown in figures 5.16 and 5.17, respectively. Based on the results, the accuracy of the EnKF-MC formulation can be improved by increasing the radius of influence ζ. This will demand more computational time which can be mitigated by increasing the number of processors during the assimilation process.

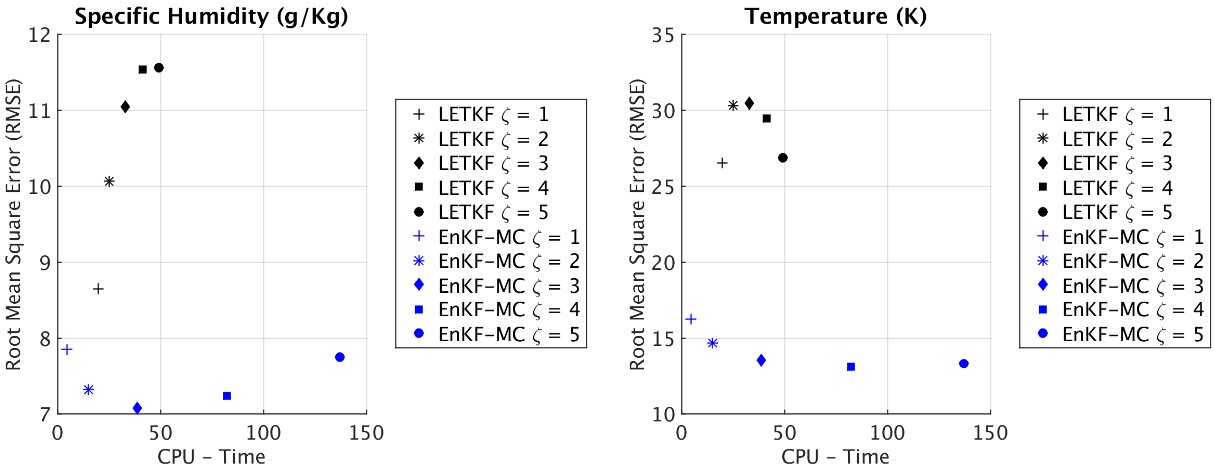


(a) RMSE for the zonal wind component (u) (b) RMSE for the meridional wind component (v)

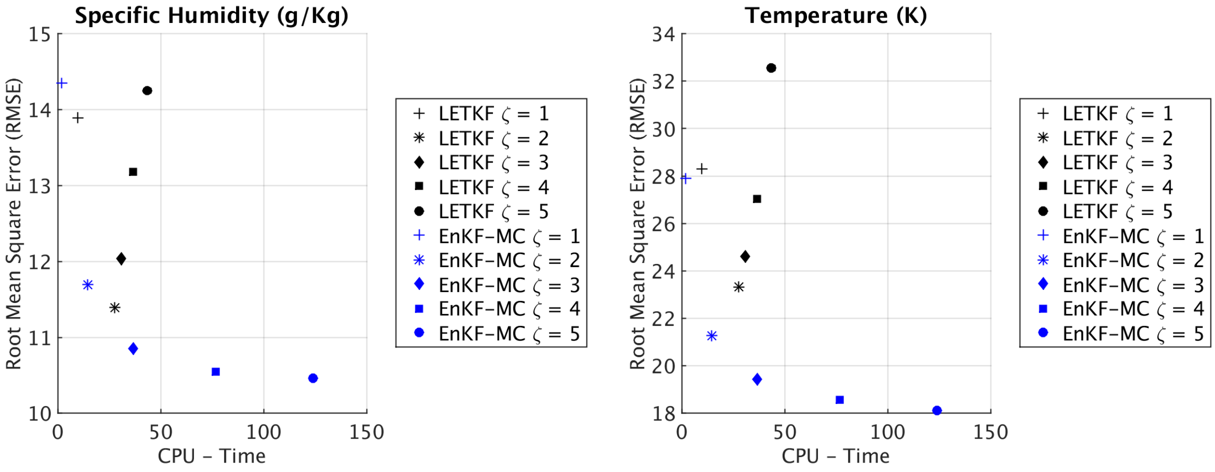


(c) RMSE for the specific humidity (sh)

Figure 5.15: RMSE of the LETKF and EnKF-MC implementations for different model variables and number of computing nodes. The number of computing nodes is next to the method name.

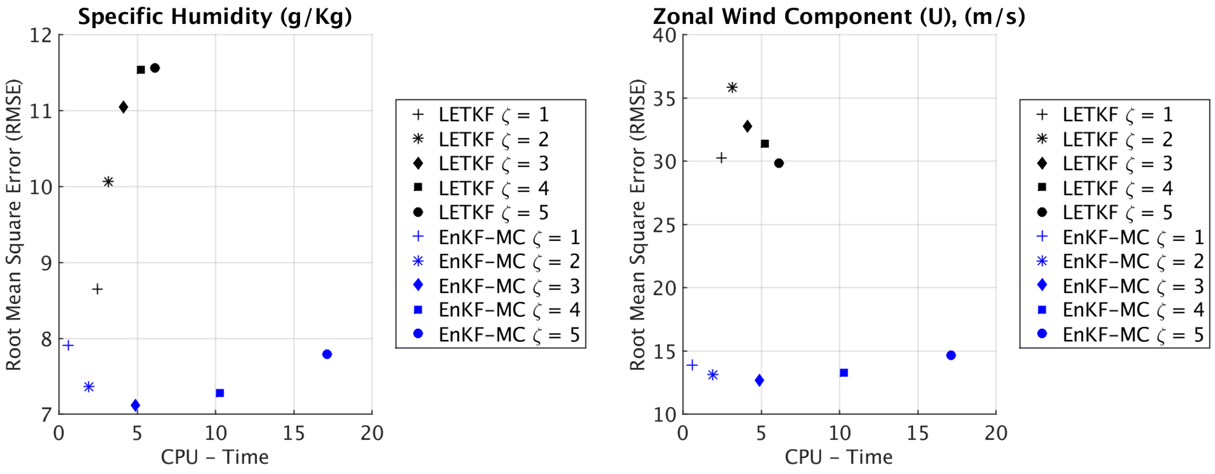


(a) $p \sim 50\%$ (b) $p \sim 50\%$

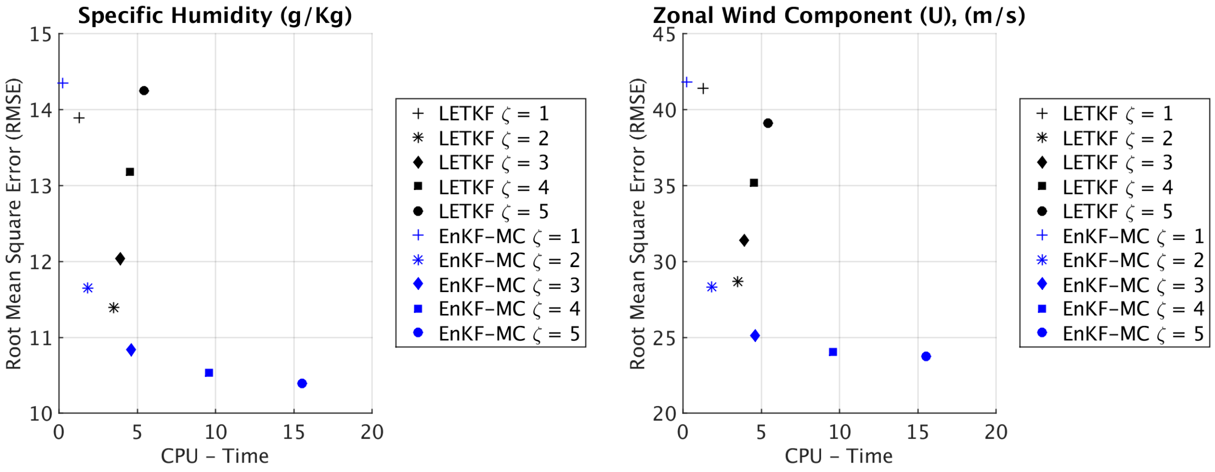


(c) $p \sim 4\%$ (d) $p \sim 4\%$

Figure 5.16: Relation between CPU-time and accuracy of the compared EnKF implementations for different radii of influence when the number of computing nodes is 6 (96 processors)



(a) $p \sim 50\%$ (b) $p \sim 50\%$

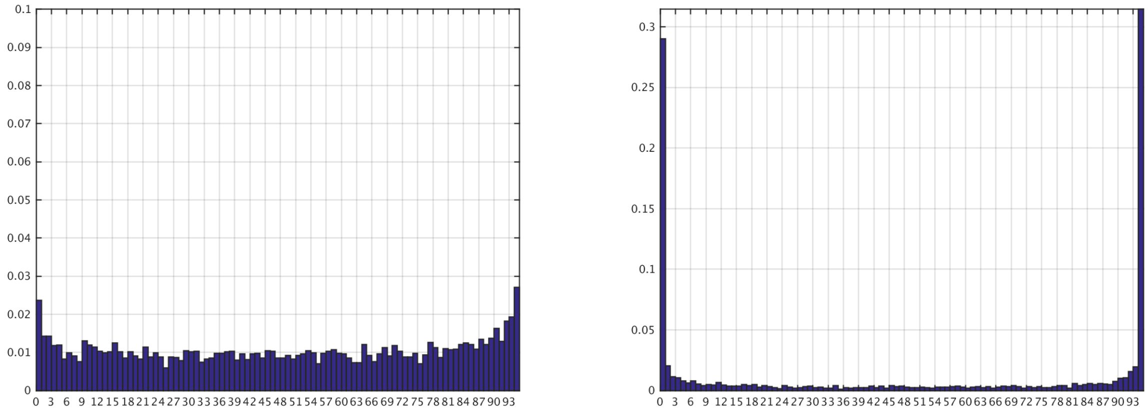


(c) $p \sim 4\%$ (d) $p \sim 4\%$

Figure 5.17: Relation between CPU-time and accuracy of the compared EnKF implementations for different radii of influence when the number of computing nodes is 48 (768 processors)

5.4.4 Statistics of the ensemble

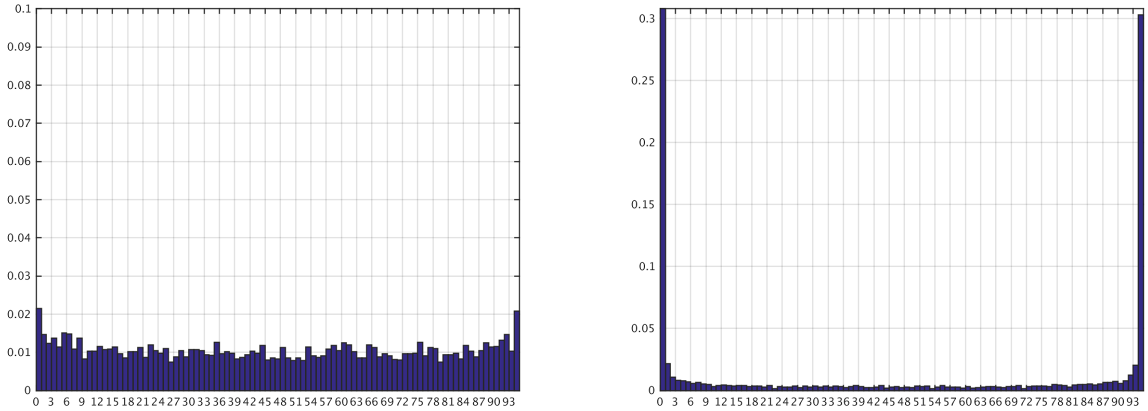
In this section, we briefly discuss the spread of the ensemble making use of rank histograms. Of course, we do not claim this to be a verification procedure but, it provides useful insights about the dispersion of the members and the level of uncertainty about the ensemble mean. The plots are based on the 5-th numerical layer of the atmosphere. We collect information across all model variables and the plots are shown in figures 5.18, 5.21, 5.19, and 5.20. Based on the results, the proposed implementation seems to be lesser sensitive to the intrinsic need of inflation than the LETKF formulation. For instance, after the assimilation, the ensemble members from the EnKF-MC are spread almost uniformly across different observation times. On the other hand, the spread in the context of the LETKF is impacted by the constant inflation factor used during the experiments (1.04) In practice, the inflation factor is set up according to historical information and/or heuristically with regard to some properties of the dynamics of the numerical model. This implies that, the dispersion of the LETKF members after the analysis will rely in how-well we estimate the optimal inflation factor for such filter. In operational data assimilation, an answer to this question can be hard to find. We think that inflation methodologies such as adaptive inflation can lead to better spread of the ensemble members in the context of the LETKF. For the proposed method, based on the experimental results, such methodology is not needed.



(a) EnKF-MC

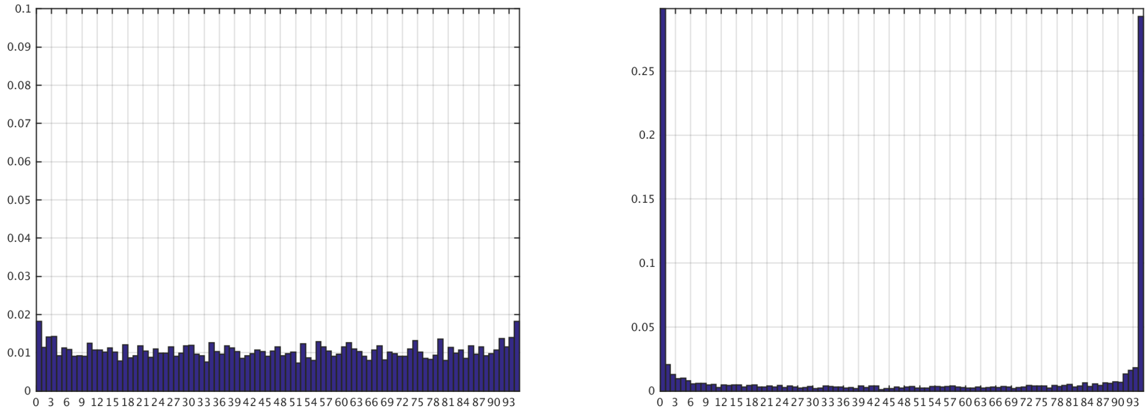
(b) LETKF

Figure 5.18: Rank-histograms for the Specific Humidity model variable. The information is collected from the 5-th model layer.



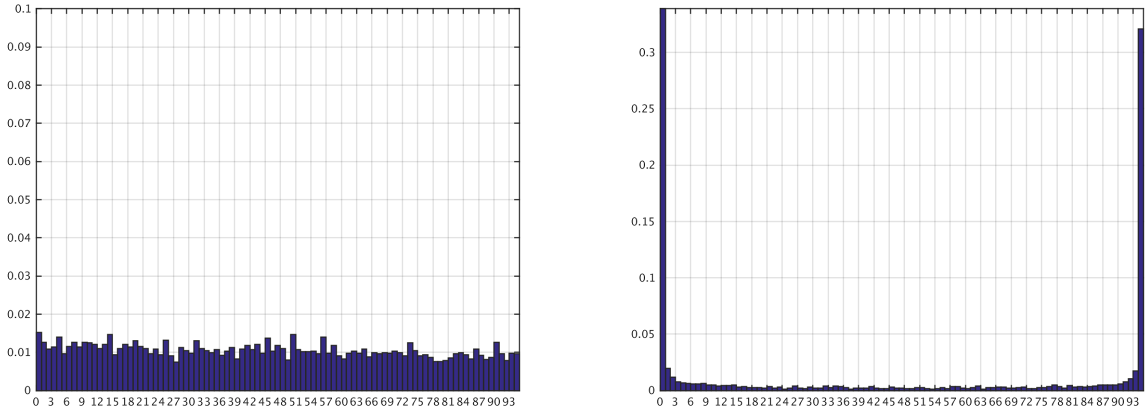
(a) EnKF-MC (b) LETKF

Figure 5.19: Rank-histograms for the Zonal Wind Component model variable. The information is collected from the 5-th model layer.



(a) EnKF-MC (b) LETKF

Figure 5.20: Rank-histograms for the Meridional Wind Component model variable. The information is collected from the 5-th model layer.



(a) EnKF-MC

(b) LETKF

Figure 5.21: Rank-histograms for the Temperature model variable. The information is collected from the 5-th model layer.

5.4.5 The impact of SVD truncation threshold

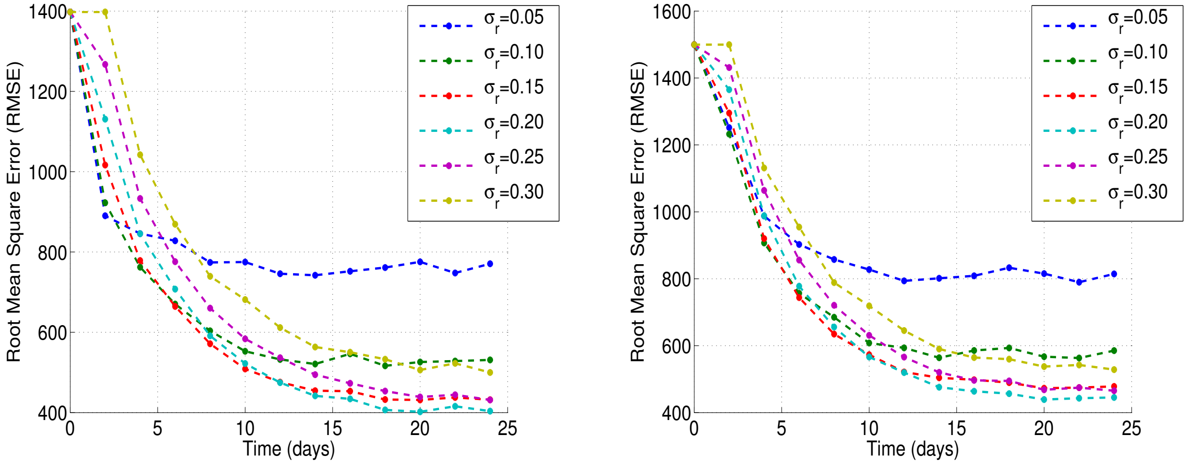
An important question arising from this research is the number of singular values/vectors to be used in (5.9). To study this question we use the same experimental setting and the sparse observational network where only 4% of the model components are observed. We apply EnKF-MC algorithm and truncate the summation (5.9) based on different thresholds σ_r.

The results are reported in Figure 5.22. Different thresholds lead to different levels of accuracy for the EnKF-MC analyses. There is no unique value of σ_r that provides the best ensemble trajectory in general; for instance, the best performance at the beginning of the assimilation window is obtained for $\sigma_r = 0.05$, but, at the end the best solution is obtained with $\sigma_r = 0.2$. This indicates that the results can be improved when σ_r is dynamically and optimally chosen. Note that, on average, the results obtained by the EnKF-MC with $\sigma_r \in \{0.15, 0.20, 0.25\}$ are much better than those when $\sigma_r = 0.10$ (and therefore much better than the results obtained by the LETKF). In Figure 5.23 snapshots of the specific humidity for different σ_r are shown. It can be seen that the spurious errors can be quickly decreased when σ_r is chosen accordingly.

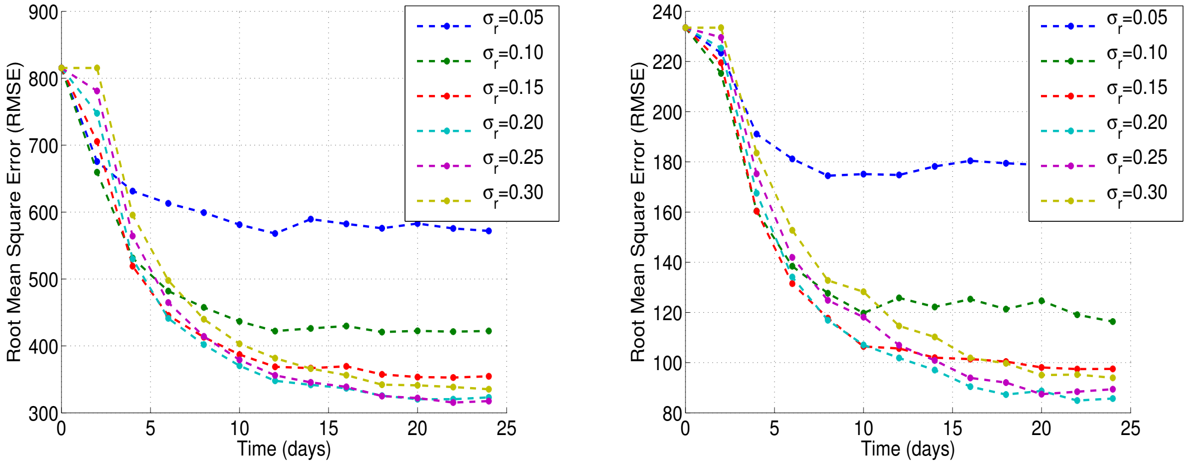
In order to understand the optimal truncation level note that the summation (5.9) can be written as follows:

$$\begin{aligned}
\boldsymbol{\beta}_{[i]} &= \sum_{j=1}^{N} \alpha_j \cdot \mathbf{u}_j^{\mathbf{Z}_{[i]}}, \qquad (5.27)\\
\alpha_j &= \frac{1}{\tau_j} \cdot {\mathbf{v}_j^{\mathbf{Z}_{[i]}}}^T \cdot \mathbf{x}_{[i]} = \frac{1}{\tau_j} \cdot {\mathbf{v}_j^{\mathbf{Z}_{[i]}}}^T \cdot \left[\widetilde{\mathbf{x}}_{[i]} + \boldsymbol{\theta}_{[i]} \right]\\
&= \underbrace{\frac{1}{\tau_j} \cdot {\mathbf{v}_j^{\mathbf{Z}_{[i]}}}^T \cdot \widetilde{\mathbf{x}}_{[i]}}_{\text{Uncorrupted data}} + \underbrace{\frac{1}{\tau_j} \cdot {\mathbf{v}_j^{\mathbf{Z}_{[i]}}}^T \cdot \boldsymbol{\theta}_{[i]}}_{\text{Error}}
\end{aligned}$$

where $\widetilde{\mathbf{x}}_{[i]}$ is the perfect data ($\mathbf{x}_{[i]} = \widetilde{\mathbf{x}}_{[i]} + \boldsymbol{\theta}_{[i]}$). The components with small singular values τ_j will amplify the error more. The threshold should be large enough to include useful information from $\widetilde{\mathbf{x}}_{[i]}$, but small enough in order to prune out the components with large error amplification. We expect that model components with large variances will need more basis vectors from (5.9) than those with lesser variance. An upper bound for the number of basis vectors (and therefore the threshold σ_r) can be obtained by inspection of the values α_j in (5.27). Figure 5.24 shows the weights α_j for different singular values for the 500-th model component of the SPEEDY model. The large zig-zag behaviors are evidence of error amplifications and therefore, we can truncate the summation (5.27) before this pattern starts to take place in the values of α_j.

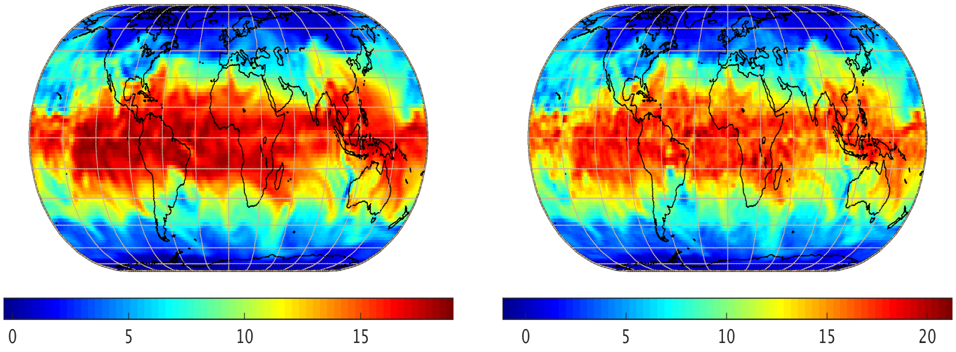


(a) Meridional wind component (m/s) (b) Zonal wind component (m/s)

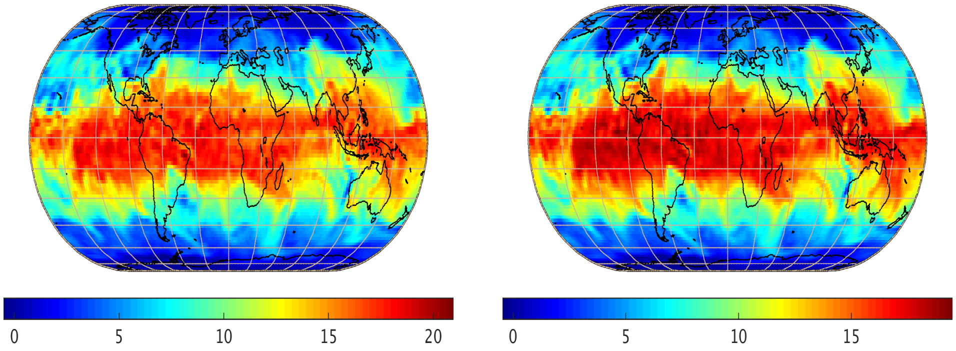


(c) Temperature (K) (d) Specific humidity (g/kg)

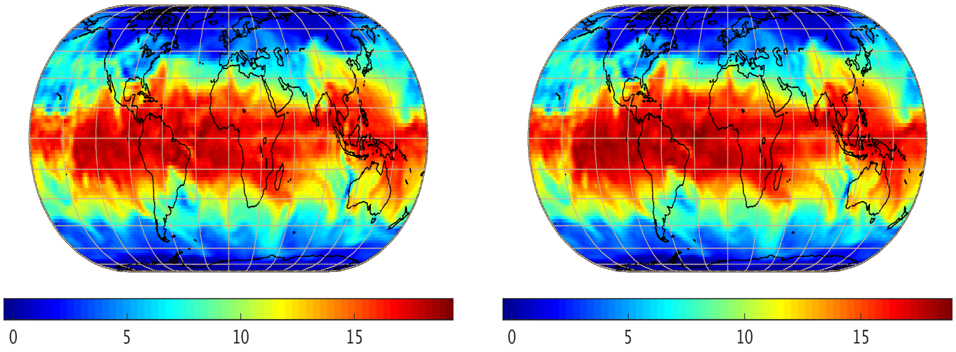
Figure 5.22: RMSE for the SPEEDY analyses obtained using different SVD truncation levels based on the σ_r values.



(a) Reference (b) $\sigma_r = 0.05$

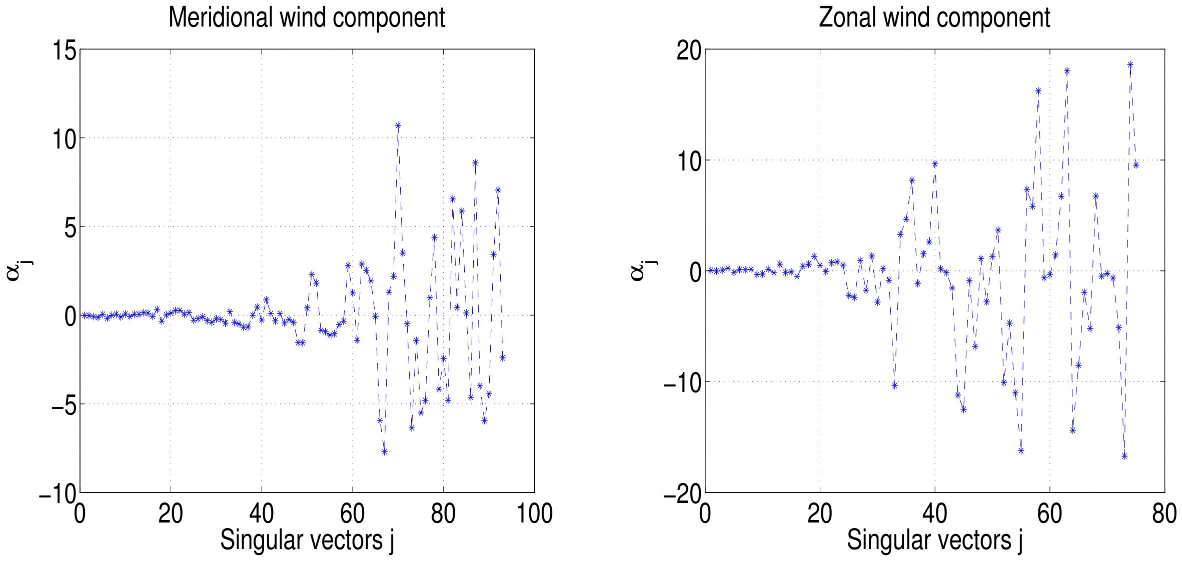


(c) $\sigma_r = 0.10$ (d) $\sigma_r = 0.15$



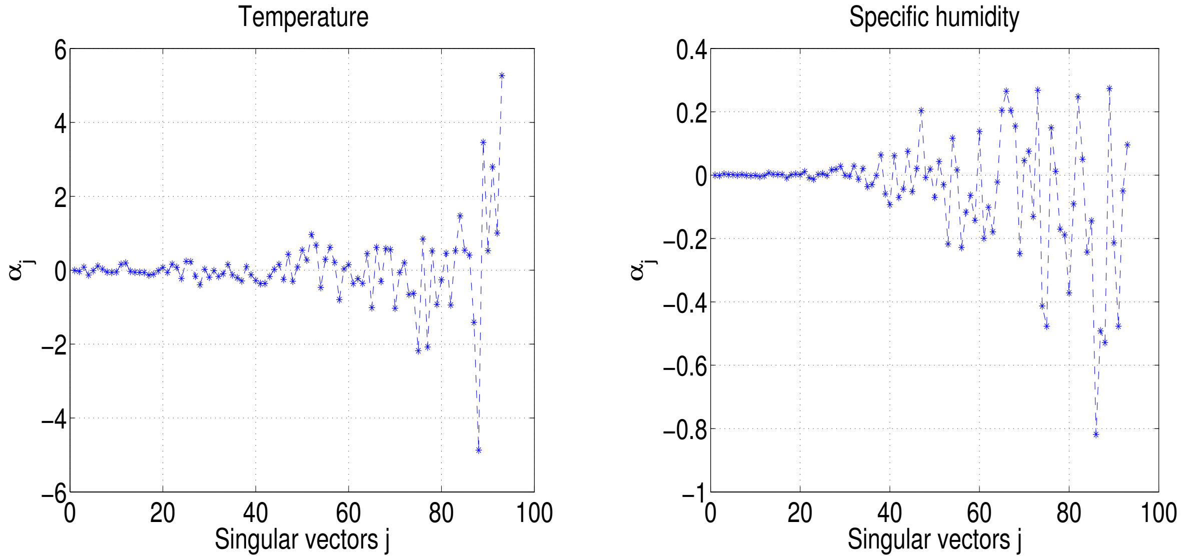
(e) $\sigma_r = 0.20$ (f) $\sigma_r = 0.30$

Figure 5.23: Snapshots at the final assimilation time (day 22) of the EnKF-MC analysis making use of different thresholds σ_r for $\zeta = 5$ and $p = 4\%$.



(a) Meridional wind component (m/s)

(b) Zonal wind component (m/s)



(c) Temperature (K)

(d) Specific humidity (g/kg)

Figure 5.24: The effect of $\boldsymbol{\theta}$ on the weights α_j for some model component i of the SPEEDY model when $\zeta = 5$ and $p = 4\%$.

5.5 Conclusions

This chapter develops an efficient implementation of the ensemble Kalman filter, named EnKF-MC, that is based on a modified Cholesky decomposition to estimate the inverse background covariance matrix. This new approach has several advantages over classical formulations. First, a predefined sparsity structure can be built into the factors of the inverse covariance. This reflects the fact that if two distant model components are uncorrelated then the corresponding entry in the inverse covariance matrix is zero; the only nonzero entries in the Cholesky factors correspond to components of the model that are located in each other's proximity. Therefore, imposing a sparsity structure on the inverse background covariance matrix is a form of covariance localization. Second, the formulation allows for a rigorous theoretical analysis; we prove the convergence of the covariance estimator for a number of ensemble members that is proportional to the logarithm of the number of states of the model therefore, when $N \approx \log n$, the background error correlations can be well-estimated making use of the modified Cholesky decomposition.

We discuss different implementations of the new EnKF-MC, and asses their computational effort. We show that domain decomposition can be used in order to decrease even more the computational effort of the proposed implementation. Numerical experiments are carried out using the Atmospheric General Circulation Model SPEEDY reveal that the analyses obtained by EnKF-MC are better than those of the LETKF in the root mean square sense when sparse observations are used in the analysis. For dense observation grids the EnKF-MC solutions are improved when the radius of influence increases, while the opposite holds true for LETKF analyses. (We stress the fact that these conclusions are true for our implementation of the basic LETKF; other implementations may incorporate advances that could make the filter perform considerably better). The use of modified Cholesky decomposition can mitigate the impact of spurious correlation during the assimilation of observations.

Chapter 6

A Derivative-Free Trust Region Framework for Variational Data Assimilation

This chapter develops a hybrid ensemble-variational approach for solving data assimilation problems. The method, called TR-4D-EnKF, is based on a trust region framework and consists of three computational steps. First an ensemble of model runs is propagated forward in time and snapshots of the state are stored. Next, a sequence of basis vectors is built and a low-dimensional representation of the data assimilation system is obtained by projecting the model state onto the space spanned by the ensemble perturbations. Finally, the low-dimensional optimization problem is solved in the reduced-space using a trust region approach; the size of the trust region is updated according to the relative decrease of the reduced order surrogate cost function. The analysis state is projected back onto the full space, and the process is repeated with the current analysis serving as a new background. A heuristic approach based on the trust region size is proposed in order to adjust the background error statistics from one iteration to the next. Experimental simulations are carried out using the Atmospheric General Circulation Model (SPEEDY). The results show that TR-4D-EnKF is an efficient computational approach, and is more accurate than the current state of the art 4D-EnKF implementations such as the POD-4D-EnKF and the Iterative Subspace Minimization methods.

The chapter is organized as follows. Section 6.1 develops the novel derivative free TR-4D-EnkF method. Aspects regarding the computational effort of the TR-4D-EnKF are discussed in Section 6.2. Numerical results using the Lorenz-96 and the quasi-geostrophic models are reported in Section 6.3, and conclusions are presented in Section 6.4.

6.1 The TR-4D-EnKF method

In this section we develop a Trust Region 4D-EnKF (TR-4D-EnKF) approach to data assimilation. We start with a general overview of the method and then present the computational algorithm in detail.

The initial solution and background covariance matrix in the model space are given by the initial approximation of the background $\mathbf{x}_0^{(0)} = \mathbf{x}_0^{\mathrm{b}}$ and error covariance matrix $\mathbf{B}_0^{(0)} = \mathbf{B}_0$, respectively. The initial ensemble is drawn from $\mathcal{N}\left(\mathbf{x}_0^{(0)}, \mathbf{B}_0^{(0)}\right)$,

$$\mathbf{X}_0^b = \left[\mathbf{x}_0^{b[1]}, \mathbf{x}_0^{b[2]}, \dots, \mathbf{x}_0^{b[N]}\right] \in \mathbb{R}^{n\times N}, \tag{6.1}$$

The ensemble members are propagated in time and $M+1$ snapshots are stored

$$\mathbf{X}^s = \begin{bmatrix} \mathbf{x}_0^{b[1]} & \mathbf{x}_0^{b[2]} & \dots & \mathbf{x}_0^{b[N]} \\ \mathbf{x}_1^{b[1]} & \mathbf{x}_1^{b[2]} & \dots & \mathbf{x}_1^{b[N]} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_M^{b[1]} & \mathbf{x}_M^{b[2]} & \dots & \mathbf{x}_M^{b[N]} \end{bmatrix} \in \mathbb{R}^{(n\cdot(M+1))\times N}. \tag{6.2}$$

We initialize the vector of weights to $\boldsymbol{\alpha} = \mathbf{0}_N$ and $j = 0$. Recall the strong 4D-Var optimization problem:

$$\mathbf{x}_0^a = \arg\min_{\mathbf{x}_0} \mathcal{J}(\mathbf{x}_0) \qquad \text{subject to} \qquad \mathbf{x}_k = \mathcal{M}_{t_{k-1}\to t_k}\left(\mathbf{x}_{k-1}\right), \tag{6.3}$$

where $\mathcal{J}(\mathbf{x}_0)$ is given in (2.14). In order to solve the numerical optimization problem (6.3) we build a quadratic model for the cost function (2.14). The standard approach makes use of the full space gradient, and possibly Hessian, of (2.14). We seek to avoid the implementation of a full adjoint model to compute exact derivatives. The idea is to approximate the derivatives of $\mathcal{J}(\mathbf{x}_0)$ by the ensemble space derivatives (2.23a) and Hessian (2.23b). The resulting quadratic model is:

$$
\begin{aligned}
\mathcal{Q}^{(j)}\left(\mathbf{s}_{\boldsymbol{\alpha}}\right) &= \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha} + \mathbf{s}_{\boldsymbol{\alpha}}) \\
&= \frac{1}{2} \cdot \left\| \mathbf{d}^{\text{b}} - \boldsymbol{\Psi}_0 \cdot (\boldsymbol{\alpha} + \mathbf{s}_{\boldsymbol{\alpha}}) \right\|^2_{\mathbf{B}_0^{-1}} + \frac{1}{2} \sum_{k=0}^{M} \left\| \mathbf{d}_k^{\text{o}} - \mathbf{Q}_k \cdot (\boldsymbol{\alpha} + \mathbf{s}_{\boldsymbol{\alpha}}) \right\|^2_{\mathbf{R}_k^{-1}} \\
&= \underbrace{\frac{1}{2} \cdot \left\| \mathbf{d}^{\text{b}} - \boldsymbol{\Psi}_0 \cdot \boldsymbol{\alpha} \right\|^2_{\mathbf{B}_0^{-1}} + \frac{1}{2} \sum_{k=0}^{M} \left\| \mathbf{d}_k^{\text{o}} - \mathbf{Q}_k \cdot \boldsymbol{\alpha} \right\|^2_{\mathbf{R}_k^{-1}}}_{\mathcal{J}_{\text{ens}}(\boldsymbol{\alpha})} \\
&+ \underbrace{\left\{ \left[\boldsymbol{\Psi}_0^T \cdot \mathbf{B}_0^{-1} \cdot \boldsymbol{\Psi}_0 + \sum_{k=0}^{M} \mathbf{Q}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{Q}_k \right] \cdot \boldsymbol{\alpha} - \mathbf{c} \right\}^T \cdot \mathbf{s}_{\boldsymbol{\alpha}}}_{\nabla \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha})^T \cdot \mathbf{s}_{\boldsymbol{\alpha}}} \\
&+ \underbrace{\frac{1}{2} \mathbf{s}_{\boldsymbol{\alpha}}{}^T \left[\boldsymbol{\Psi}_0^T \cdot \mathbf{B}_0^{-1} \cdot \boldsymbol{\Psi}_0 + \sum_{k=0}^{M} \mathbf{Q}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{Q}_k \right] \mathbf{s}_{\boldsymbol{\alpha}}}_{\frac{1}{2} \mathbf{s}_{\boldsymbol{\alpha}}{}^T \nabla^2 \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) \mathbf{s}_{\boldsymbol{\alpha}}} \,,
\end{aligned}
$$

where $\mathbf{c} = \boldsymbol{\Psi}_0{}^T \cdot \mathbf{B}_0^{-1} \cdot \mathbf{d}^{\text{b}} + \sum_{k=0}^{M} \mathbf{Q}_k^T \cdot \mathbf{R}_k^{-1} \cdot \mathbf{d}_k \in \mathbb{R}^{N \times 1}$. This can be rewritten as

$$
\mathcal{Q}^{(j)}(\mathbf{s}_{\boldsymbol{\alpha}}) = \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) + \mathbf{s}_{\boldsymbol{\alpha}}{}^T \nabla_{\boldsymbol{\alpha}} \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) + \frac{1}{2} \mathbf{s}_{\boldsymbol{\alpha}}{}^T \nabla^2_{\boldsymbol{\alpha},\boldsymbol{\alpha}} \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) \mathbf{s}_{\boldsymbol{\alpha}} \,. \tag{6.4}
$$

The optimal step $\mathbf{s}_{\boldsymbol{\alpha}}{}^*$ in the ensemble space is given by the solution of the constrained optimization sub-problem

$$
\mathbf{s}_{\boldsymbol{\alpha}}{}^* = \arg\min_{\mathbf{s}_{\boldsymbol{\alpha}}} \mathcal{Q}^{[\text{j}]}(\mathbf{s}_{\boldsymbol{\alpha}}) \,, \tag{6.5a}
$$

$$
\text{subject to } \left\| \boldsymbol{\Psi}_0 \cdot (\boldsymbol{\alpha} + \mathbf{s}_{\boldsymbol{\alpha}}) \right\| \leq \Delta^{(j)} \,. \tag{6.5b}
$$

The trust region constraint is formulated such as to use the trust region radius $\Delta^{(j)}$ from the full model space.

The solution of (6.5) provides the following trial point in the ensemble space

$$
\boldsymbol{\alpha}' = \boldsymbol{\alpha} + \mathbf{s}_{\boldsymbol{\alpha}}{}^* \,,
$$

which corresponds to the following state in the model space

$$
\mathbf{x}_0' = \mathbf{x}_0^{(j)} + \underbrace{\boldsymbol{\Psi}_0 \cdot \overbrace{(\boldsymbol{\alpha} + \mathbf{s}_{\boldsymbol{\alpha}}{}^*)}^{\boldsymbol{\alpha}^{\text{trial}}}}_{\boldsymbol{\delta}\mathbf{x}^*} \,. \tag{6.6a}
$$

The problem (6.5) is solved using Lagrangian multipliers. The first and second derivatives of the model (6.4) are

$$
\nabla \mathcal{Q}^{(j)}(\mathbf{s}_{\boldsymbol{\alpha}}) = \nabla \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) + \nabla^2 \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) \cdot \mathbf{s}_{\boldsymbol{\alpha}} \in \mathbb{R}^{N \times 1} \,,
$$

and

$$\nabla^2 \mathcal{Q}^{(j)}(\mathbf{s}_{\boldsymbol{\alpha}}) = \nabla^2 \mathcal{J}_{\text{ens}}(\boldsymbol{\alpha}) \in \mathbb{R}^{N \times N}\,,$$

respectively. The trust region constraint (6.5b) can be written as

$$\|\boldsymbol{\alpha} + \mathbf{s}_{\boldsymbol{\alpha}}\|_{\mathbf{P}}^2 - {\Delta^{(j)}}^2 + \varsigma^2 = 0\,,$$

where $\varsigma \in \mathbb{R}$ is a slack variable and $\mathbf{P} = {\boldsymbol{\Psi}_0}^T \cdot \boldsymbol{\Psi}_0 \in \mathbb{R}^{N \times N}$. Consider the Lagrangian

$$\mathcal{L}\left(\mathbf{s}_{\boldsymbol{\alpha}},\, \lambda,\, \varsigma\right) = \mathcal{Q}(\mathbf{s}_{\boldsymbol{\alpha}}) + \lambda \cdot \left[\|\boldsymbol{\alpha} + \mathbf{s}_{\boldsymbol{\alpha}}\|_{\mathbf{P}}^2 - {\Delta^{[\mathrm{j}]}}^2 + \varsigma^2\right]\,, \tag{6.8}$$

The constrained problem (6.5) becomes the unconstrained optimization problem

$${\mathbf{s}_{\boldsymbol{\alpha}}}^* = \arg\min_{\mathbf{s}_{\boldsymbol{\alpha}}} \mathcal{L}\left(\mathbf{s}_{\boldsymbol{\alpha}},\, \lambda,\, \varsigma\right)\,. \tag{6.9}$$

The stationarity conditions for (6.8) read:

$$\nabla \mathcal{L}\left(\mathbf{s}_{\boldsymbol{\alpha}},\, \lambda,\, \varsigma\right) = \begin{bmatrix} \mathcal{L}_{\mathbf{s}_{\boldsymbol{\alpha}}}\left(\mathbf{s}_{\boldsymbol{\alpha}},\, \lambda,\, \varsigma\right) \\ \mathcal{L}_{\lambda}\left(\mathbf{s}_{\boldsymbol{\alpha}},\, \lambda,\, \varsigma\right) \\ \mathcal{L}_{\varsigma}\left(\mathbf{s}_{\boldsymbol{\alpha}},\, \lambda,\, \varsigma\right) \end{bmatrix} = 0\,,$$

where

$$\begin{aligned} \mathcal{L}_{\mathbf{s}_{\boldsymbol{\alpha}}}\left(\mathbf{s}_{\boldsymbol{\alpha}},\, \lambda,\, \varsigma\right) &= \nabla \mathcal{Q}\left(\mathbf{s}_{\boldsymbol{\alpha}}\right) + 2\lambda\, \mathbf{P} \cdot (\boldsymbol{\alpha} + \mathbf{s}_{\boldsymbol{\alpha}}) = \mathbf{0} \in \mathbb{R}^N\,, \\ \mathcal{L}_{\lambda}\left(\mathbf{s}_{\boldsymbol{\alpha}},\, \lambda,\, \varsigma\right) &= \|\boldsymbol{\alpha} + \mathbf{s}_{\boldsymbol{\alpha}}\|_{\mathbf{P}}^2 - {\Delta^{[\mathrm{j}]}}^2 + \varsigma^2 = 0 \in \mathbb{R}\,, \\ \mathcal{L}_{\varsigma}\left(\mathbf{s}_{\boldsymbol{\alpha}},\, \lambda,\, \varsigma\right) &= 2 \cdot \lambda \cdot \varsigma = 0 \in \mathbb{R}\,, \end{aligned}$$

which provides all the information needed to solve (6.9). Note that, when the full step is taken in the ensemble space

$$\left\|\boldsymbol{\Psi}_0 \cdot \boldsymbol{\alpha}^{'}\right\| \leq \Delta^{(j)}\,,$$

the exact solution (2.21) can be employed. Then, $M+1$ snapshots of the full model solution started from $\mathbf{x}_0^{'}$ (6.6a) are stored. The following ratio is computed:

$$\rho^{(j)} = \frac{\mathcal{J}\left(\mathbf{x}^{(j)}\right) - \mathcal{J}\left(\mathbf{x}^{'}\right)}{\mathcal{Q}\left(\mathbf{0}_N\right) - \mathcal{Q}\left({\mathbf{s}_{\boldsymbol{\alpha}}}^*\right)} = \frac{\mathcal{J}\left(\mathbf{x}^{[\mathrm{j}]}\right) - \mathcal{J}\left(\mathbf{x}^{'}\right)}{\mathcal{J}_{\text{ens}}\left(\boldsymbol{\alpha}\right) - \mathcal{J}_{\text{ens}}\left(\boldsymbol{\alpha}^{'}\right)}\,. \tag{6.10}$$

Based on the $\rho^{(j)}$ value, the next updates are made for the solution in the model space

$$\mathbf{x}^{(\mathrm{j}+1)} := \begin{cases} \mathbf{x}^{(\mathrm{j})} & \text{for } \rho^{(j)} \leq \eta, \\ \mathbf{x}^{'} & \text{otherwise,} \end{cases} \tag{6.11}$$

and for the TR radius size

$$\Delta^{(j+1)} \;:=\; \begin{cases} \Delta^{(j)} \cdot \gamma_{\text{dec}} & \text{for } \rho^{(j)} < \theta_1, \\ \Delta^{(j)} & \text{for } \theta_1 \le \rho^{(j)} < \theta_2 \text{ or } \rho^{(j)} > 1, \\ \min\left(\Delta^{(j)} \cdot \gamma_{\text{inc}},\, \Delta_{\max}\right) & \text{for } \theta_2 \le \rho^{(j)} \le 1. \end{cases} \tag{6.12}$$

The current solution becomes the new background and therefore, a new ensemble of full model solutions is generated, snapshots are taken, a new set of basis vectors is built, and the overall process is repeated.

Since a partial assimilation of observations has been carried out the uncertainty associated with the new background is changed. As an analogy, in the EnKF the spread of the ensemble members around the background is decreased after the analysis step. Consequently, before generating a new ensemble, we want to adjust the spread of the background errors. This is done according to the heuristic formula

$$\mathbf{B}_0^{(j+1)} := \lambda_{\mathbf{B}}(\Delta) \cdot \mathbf{B}_0^{(j)}\,, \tag{6.13}$$

where $\lambda_{\mathbf{B}}\left(\Delta\right)$ is a function of the current TR radius size. Note that the TR radius is large when the decrease of the current (quadratic) model is a good predictor of the full model function decrease. In our context, if the dynamics of the full (nonlinear) model is well represented by the ensemble, the prediction done using the quadratic model $\mathcal{Q}(\mathbf{s}_{\boldsymbol{\alpha}})$ is close to the actual reduction of the cost function $\mathcal{J}\left(\mathbf{x}\right)$ and the TR radius is increased. In this case, we want the $\lambda_{\mathbf{B}}\left(\Delta\right)$ value to be small in order to decrease the uncertainty of the new ensemble around $\mathbf{x}_0^{\text{b}}$. Vice versa, a small TR radius indicates that the current set of basis vectors does not represent well the dynamics of the model. The current assimilation step is not expected to decrease uncertainty; to keep the same uncertainty level for the next ensemble generation we need $\lambda_{\mathbf{B}}\left(\Delta\right) \approx 1$. Both cases are captured by the following heuristic function

$$\lambda_{\mathbf{B}}\left(\Delta\right) = \frac{\Delta_{\max}}{\Delta_{\max} + \Delta}\,,$$

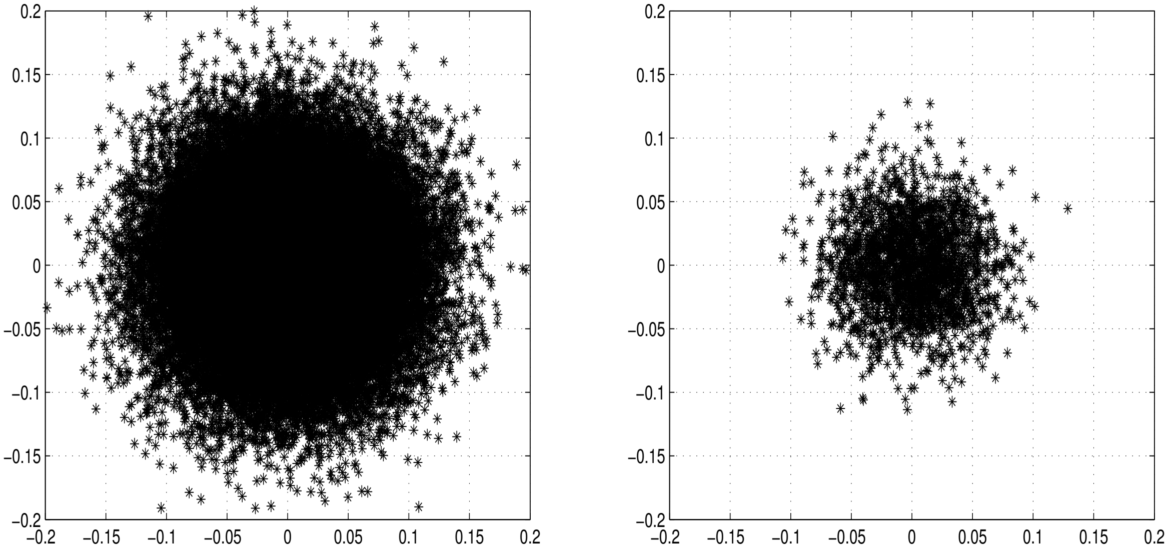
which provides an inverse relation between the TR radius and the spread of the ensemble members. Other functions can be considered as well. In summary, when the TR radius is large the confidence in the current solution is increased

$$\lim_{\Delta \to \Delta_{\max}} \lambda_{\mathbf{B}} = \frac{\Delta_{\max}}{2 \cdot \Delta_{\max}} = \frac{1}{2}\,.$$

On the other hand, when the TR size is small, the current level of background uncertainty remains unchanged for the new ensemble generation

$$\lim_{\Delta \to 0} \lambda_{\mathbf{B}} = \frac{\Delta_{\max}}{\Delta_{\max}} = 1\,.$$

The effects of the scaling of $\mathbf{B}_0$ on the new background ensemble are shown in the Figure 6.1 for a 2D example. The choice $\lambda_{\mathbf{B}} = 1$ keeps the uncertainty unchanged (Figure 6.1a), while $\lambda_{\mathbf{B}} = 1/2$ shrinks the spread by half (Figure 6.1b).



(a) $\lambda_{\mathbf{B}} = 1$

(b) $\lambda_{\mathbf{B}} = \frac{1}{2}$

Figure 6.1: Impact of the scaling of $\mathbf{B}_0$ on the spread of the newly generated ensemble.

The outline of the TR-4D-EnKF method is shown below.

1. **Initialization.** The TR parameters are initialized. $\mathbf{x}_0^{(0)} := \mathbf{x}_0^b$, $\mathbf{B}_0^{(0)} = \mathbf{B}_0$, $\boldsymbol{\alpha} = \mathbf{0}_N$ and $j = 0$.

2. **Ensemble generation.** The initial ensemble is drawn from a distribution with mean $\mathbf{x}_0^{(j)}$ and background error covariance matrix $\mathbf{B}_0^{(j)}$. M snapshots for each ensemble member are stored (2.17) and the basis functions (2.19) are computed.

3. **Model construction.** Build the quadratic model (6.4).

4. **Optimization problem.** Solve the optimization sub-problem (6.5) and compute $\boldsymbol{\alpha}^{'}$ and $\mathbf{x}_0^{'}$.

5. **Ratio of prediction.** Run the full model to obtain M snapshots of the solution started from $\mathbf{x}_0^{'}$ and compute the ratio $\rho^{(j)}$ (6.10).

6. **Solution and TR size update.** Update the solution in the model space and the TR radius according to (6.11) and (6.12), respectively.

7. **Background update.** Set $\boldsymbol{\alpha} := \mathbf{0}_N$, scale the covariance matrix (6.13), $\mathrm{j} \leftarrow \mathrm{j}+1$, and go to Step 2.

The iteration stopping criterion for the TR-4D-EnKF implementation can be based on the total number of iterations, on the trace of $\mathbf{B}_0^{(j)}$, or on the trust region radius $\Delta^{(j)}$.

6.2 Computational effort

In this section, we briefly discus the computational effort of the proposed implementation in terms of memory storage and number of long computations. We assume the estimated data error covariance matrix $\mathbf{R} \in \mathbb{R}^{m \times m}$ to be (block) diagonal. Even more, the observational operator $\mathcal{H} : \mathbb{R}^m \to \mathbb{R}^N$ can be efficiently applied, for instance, it chooses the components from the vector state to be observed and therefore, long computations are not required.

As is usual in 4D-EnKF implementations, the TR-4D-EnKF requires the storage of an ensemble of snapshots and a set of basis vectors, both of size $n \cdot M \cdot N$. Even more, the snapshots of the ensemble mean and the trial point in model space are needed as well, each one of size $n \cdot M$. Likewise, the size of the observations in the assimilation windows reads $m \cdot M$. Thus, the total amount of the data to be stored per iteration by the TR-4D-EnKF reads

$$\mathrm{D}_{\mathrm{TR-4D-EnKF}} = 2 \cdot n \cdot M \cdot N + 2 \cdot n \cdot M + m \cdot M,$$

since the storage of the trial point (vector of weights in the ensemble space) is irrelevant, it has been omitted in the above equation. The main difference between the TR-4D-EnKF and the ISM lies in the computation of the trial point (and its snapshots). Thus, the total amount of data per iteration for the ISM method reads:

$$\mathrm{D}_{\mathrm{ISM}} = \mathrm{D}_{\mathrm{TR-4D-EnKF}} - n \cdot M = 2 \cdot n \cdot M \cdot N + n \cdot M + m \cdot M,$$

which is equivalent to the data size of the POD-4D-EnKF (one iteration of ISM).

In the context of computations, TR-4D-EnKF generates a set of basis vectors from the ensemble in $\mathcal{O}\left(n \cdot N\right)$ long computations. The computation of the optimal step is bounded by $\mathcal{O}\left(K \cdot N^3\right)$, where K depends on the optimization method used. The function evaluations can be efficiently performed with no more than $\mathcal{O}\left(n \cdot S_{\mathbf{B}^{-1}} + m \cdot S_{\mathbf{R}^{-1}}\right)$ multiplications, where $S_{\mathbf{B}^{-1}}$ and $S_{\mathbf{R}^{-1}}$ depend on the level of sparsity of the matrices $\mathbf{B}^{-1}$ and $\mathbf{R}^{-1}$, respectively. Based on this analysis the number of long computations per iteration for the TR-4D-EnKF reads:

$$\mathrm{LC}_{\mathrm{TR-4D-EnKF}} \in \mathcal{O}\left(n \cdot N + K \cdot N^3 + n \cdot S_{\mathbf{B}^{-1}} + m \cdot S_{\mathbf{R}^{-1}}\right).$$

Likewise, the optimal step of the ISM can be computed in $\mathcal{O}\left(N^3\right)$ since no optimization problem (i.e., there is no quadratic model) is used, the other computations are similar to those of the TR-4D-EnKF. Hence, the computations per iteration performed by the ISM method reads:

$$\mathrm{LC}_{\mathrm{ISM}} \in \mathcal{O}\left(n \cdot N + N^3 + n \cdot S_{\mathbf{B}^{-1}} + m \cdot S_{\mathbf{R}^{-1}}\right),$$

which is equivalent to the total number of computations for the POD-4D-EnKF. Although the computational efforts of the compared methods seems to be similar, the TR-4D-EnKF requires an additional model propagation per iteration in order to compute the trial point and its snapshots.

Now we are ready to test our implementation and compare it with other 4D-EnKF implementations discussed in Section 2.3.

6.3 Numerical experiments

In this section we study the accuracy and performance of the TR-4D-EnKF approach. The proposed implementation is compared with the 4D-EnKF implementations discussed in Section 2.3: POD and ISM, using the Atmospheric General Circulation Model (AGCM), better known as SPEEDY [Mol03a, KM03, BKKM04, KMB06b, KMK$^+$12], by the International Centre for Theoretical Physics (ICTP) in Trieste, Italy. The ICTP AGCM is based on a spectral dynamical core developed at the Geophysical Fluid Dynamics Laboratory (GFDL) [HS94] at the Princeton Univeristy Forrestal Campus in Princeton, USA. All the physics are developed on the sphere concisely, the Earth. It is a hydrostatic, σ-coordinate, spectral-transform model in the vorticity-divergence form described by Bourke [Bou74], with semi-implicit treatment of gravity waves. The basic prognostic variables are shown in the Table 6.1.

Table 6.1: Pronostic variables in the SPEEDY model.

Variable	**Symbol**	**Number of Layers**
Vorticity	Ω	8
Divergence	χ	8
Pressure	ψ	8
Specific Humidity	Λ	8
Temperature	τ	1

The number of longitudinal and latitudinal points are 96 and 48, respectively. The longitudinal values ranges evenly in [0, 96] while the latitudinal values ranges evenly in [0, 48]. This provides a total number of 4096 points per layer. Each layer is mapped to the vector state which provides a total number of $n = 152064$ components. Only 50% of the components are observed at each layer, this corresponds to 2048 components per layer and a total number of $m = 67584$ components being observed in the system.

The metrics used in the tests are the CPU time (which is reported per iteration) and the root mean square error

$$\text{RMSE} = \sqrt{\frac{1}{M} \cdot \sum_{k=0}^{M} \left(\mathbf{x}^{*}{}_{k} - \mathbf{x}_{k}^{\text{a}}\right)^{T} \cdot \left(\mathbf{x}^{*}{}_{k} - \mathbf{x}_{k}^{\text{a}}\right)}\,,$$

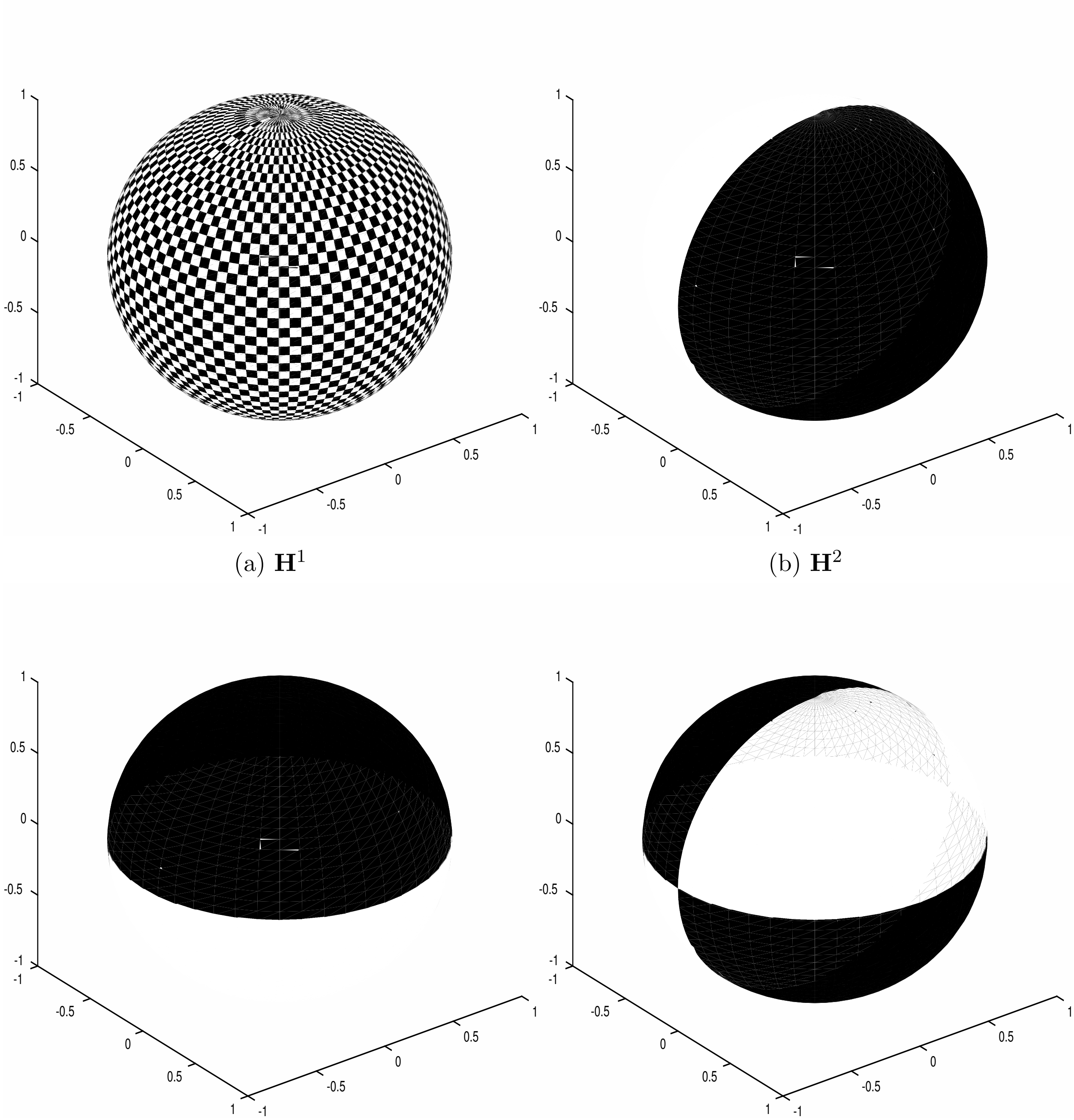
which provides the average of the squared root differences between the reference solution $\mathbf{x}^{*}$ and the analysis $\mathbf{x}^{\text{a}}$ over the observation times.

Some details regarding the numerical implementation of the data assimilation methods:

- Three computational languages are used to carried out the different steps of the compared methods: C, FORTRAN and MATLAB.
- The forecast step of the ensemble members is performed in C language making use of MPI. Each ensemble member is independently propagated in time.
- The number of ensemble members matches the number of processors.
- The initial conditions of the ensemble are coded in FORTRAN 77.
- The assimilation step is carried out in MATLAB.
- The communication between different programming languages is performed in the data level via NetCDF files.
- The main core of the program is written in bash language which integrates the different components of the data assimilation process.
- A two day assimilation window is set-up with observations taken evenly each 3 hours.

Other parameters of the numerical simulation are described below.

- Starting in rest, the ICTP AGMC model is propagated in time for three months, after that, we assume the final state to be the true initial condition $\mathbf{x}_0^{\text{true}}$ for our testing.



(a) $\mathbf{H}^1$ (b) $\mathbf{H}^2$

(c) $\mathbf{H}^3$ (d) $\mathbf{H}^4$

Figure 6.2: Linear observation operators. The dark areas correspond to observed components.

- Four linear observation operators on the Earth are considered, they are evenly and sequentially distributed over the assimilation window. This mimics, for instance, the use of different sets of sensors at different times in the ocean. The observational operators are shown in figure 6.2.

- The initial background state is a sample from the distribution:

  $$\mathbf{x}^{\mathrm{b}} \sim \mathcal{N}\left(\mathbf{x}^{\mathrm{true}},\, \mathbf{B}_0\right)\,,$$

  where $\mathbf{B}_0 = (0.05)^2 \cdot \mathbf{I}_{n\times n}$. Similarly, the initial ensemble are samples drawn from the distribution

  $$\mathbf{x}^{b(i)} \sim \mathcal{N}\left(\mathbf{x}^{\mathrm{b}},\, \mathbf{B}_0\right)\,,$$

  for $1 \le i \le N$.

- Four ensemble sizes N are considered: 10, 20, 40 and 80.

- Data errors are normally distributed with parameters

  $$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}_m,\, \mathbf{R}\right)$$

  with $\mathbf{R} = (0.01)^2 \cdot \mathbf{I}_{m\times m}$.

- Five iterations are considered for the ISM and TR-4D-EnKF implementations. This is more than reasonable, in practice, we are not able to propagate the model many times.

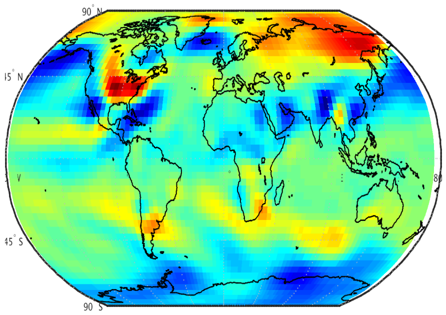
- The parameters for the TR-4D-EnKF optimization are $\gamma_{\mathrm{inc}} = 1.4$, $\gamma_{\mathrm{dec}} = 0.5$, $\Delta_{\max} = 100$, $\Delta_0 = 0.1$, $\eta = 0.1$, $\theta_1 = 0.25$ and $\theta_2 = 0.75$.

The RMSE and the elapsed times are presented in the Tables 6.2 and 6.3, respectively. Notice, all the 4D-EnKF methods presented in this paper are able to improve the background initial condition $\mathbf{x}_0^b$ in terms of RMSE. This behaviour holds even in difficult scenarios such as small ensemble sizes (i.e., $N = 10$), high dimensional vector states and only 50% of observed components from the dynamical system. Notice, the POD-4D-EnKF outperforms, in average, the accuracy of the background state by one order of magnitude. The improving is more notorious when the dimension of the ensemble is increased. This obeys to a better representation of the background error statistics onto the space spanned by the ensemble perturbations. As is expected, the more ensemble members, the lesser sampling errors are involved in the assimilation step. Likewise, the ISM method provides very good results with just five iterations and within a reasonable computational effort. The results are much better than the ones obtained via POD-4D-EnKF (equivalent to a single iteration of the ISM). This justifies the iterative refinement of solutions in the context of reduce space approaches. The proposed TR-4D-EnKF outperforms the initial solution in the context of RMSE and after five iterations, the RMSE values look similar to those obtained via the ISM. The figures 6.3 and 6.4 show the estimated state obtained by each method for the vorticity and the temperature at the Earth's surface. There is no doubt that the initial background state $\mathbf{x}_0^b$ provides a poor estimation of the vorticity and temperature at the Earth's surface. On the contrary,

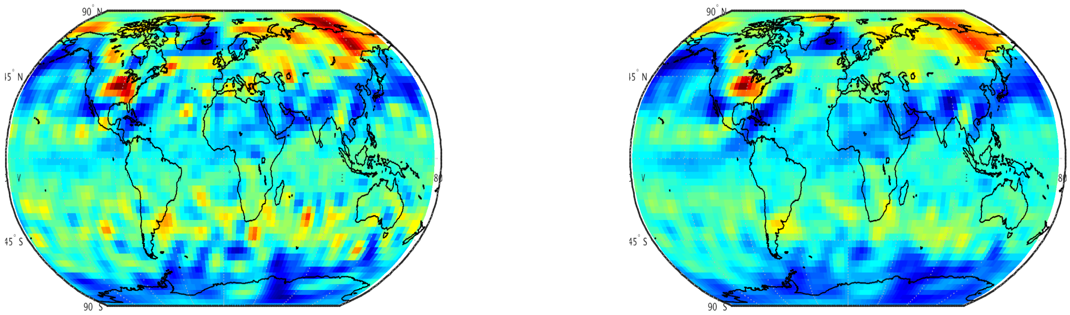
Table 6.2: Root mean square error for different ensemble sizes and data assimilation methods. The notation reads $x(y) = x \times 10^y$.

N	**Method**	$\boldsymbol{\Omega}$	$\boldsymbol{\chi}$	$\boldsymbol{\psi}$	$\boldsymbol{\Lambda}$	$\boldsymbol{\tau}$
N/A	Background	1.64(−3)	1.62(−3)	5.58(2)	2.08(3)	1.05(2)
10	POD-4D-EnKF	9.89(−4)	9.79(−4)	3.29(2)	1.23(3)	6.33(1)
	ISM	5.70(−4)	5.56(−4)	1.82(2)	6.83(2)	3.70(1)
	TR-4D-EnKF	5.66(−4)	5.53(−4)	1.81(2)	6.79(2)	3.66(1)
20	POD-4D-EnKF	9.07(−4)	8.91(−4)	2.97(2)	1.11(3)	5.92(1)
	ISM	4.07(−4)	4.02(−4)	1.29(2)	4.88(2)	2.67(1)
	TR-4D-EnKF	4.07(−4)	4.02(−4)	1.29(2)	4.87(2)	2.67(1)
40	POD-4D-EnKF	8.88(−4)	8.80(−4)	2.88(2)	1.07(3)	5.65(1)
	ISM	3.03(−4)	3.10(−4)	9.79(2)	3.69(2)	2.07(1)
	TR-4D-EnKF	2.93(−4)	3.02(−4)	9.47(2)	3.56(2)	2.01(1)
80	POD-4D-EnKF	7.99(−4)	8.14(−4)	2.67(2)	9.98(2)	5.11(1)
	ISM	2.31(−4)	2.43(−4)	7.49(2)	2.83(2)	1.59(1)
	TR-4D-EnKF	2.11(−4)	2.29(−4)	6.99(2)	2.65(2)	1.46(1)

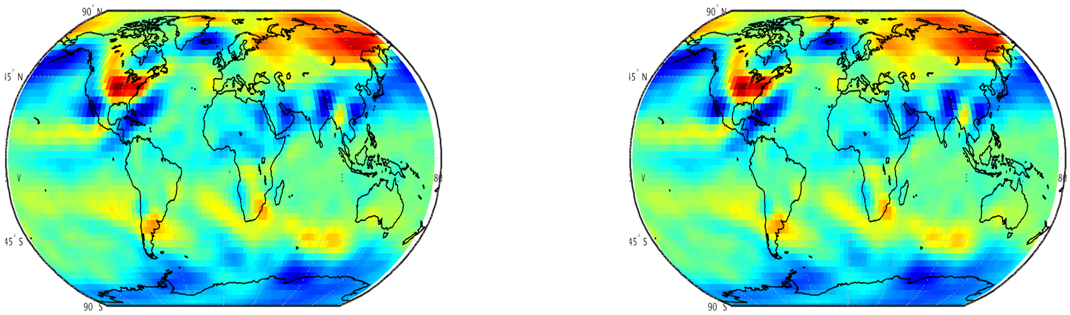
the solutions obtained by the POD-4D-EnKF seems to be relatively “close” to the true state of the system. However, the ISM provides a much better approximation than the POD-4D-EnKF. Likewise, the TR-4D-EnKF approximation is similar to the ISM one. Figure 6.5 provides the time evolution of the errors for the vorticity field among the 8 layers. Note that, the RMSE values in time of the ISM and TR-4D-EnKF are relatively close under the RMSE metric. However, figure 6.6 shows a different perspective of this two methods. We report the background and POD-4D-EnKF solutions for comparison purposes since those methods does not require iterations. Note that, in all the cases, the TR-4D-EnKF method performs better than the ISM in the first three iterations. The performance is much better when large ensemble sizes are used in the assimilation window. Note that, the accuracy obtained by five iterations of the ISM is equivalent to that obtained by two iterations of the TR-4D-EnKF. In practice, this is extremely important since model propagation is a labor-intensive process and therefore, the lesser number of times the model is propagated, the better. This implies we are able to obtain good approximations with lesser number of iterations making use of the TR-4D-EnKF than the ISM. This gap between the two methods can be explained in terms of the initial background distribution at each iteration: the TR-4D-EnKF decreases the uncertainty according to trust region sizes, when a good representation of the background error statistics (ensemble members) is contained in the ensemble, the next iteration of the TR-4D-EnKF method will solve an optimization problem where the uncertainty around the initial condition has been decreased. This is not the case of the ISM method where the same uncertainty is hold among all the iterations.



(a) True State

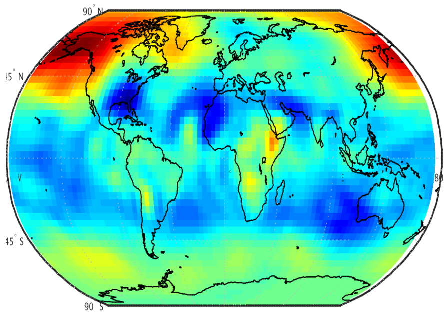


(b) Background (c) POD-4D-EnKF

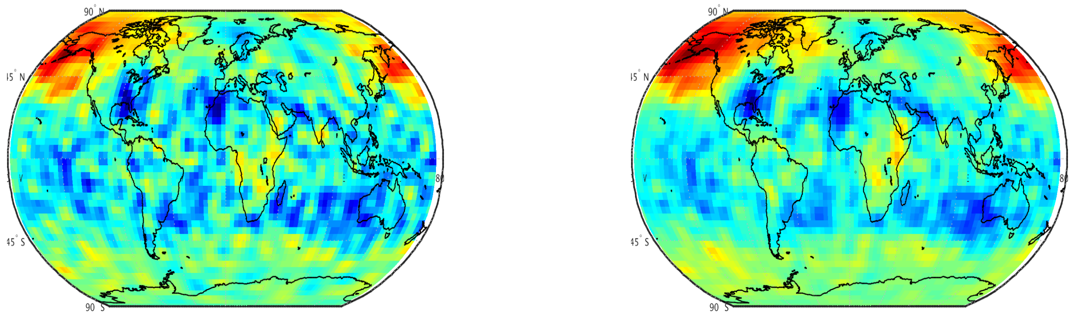


(d) ISM (e) TR-4D-EnKF

Figure 6.3: Initial vorticities at the Earth's surface from the analysis states $\mathbf{x}_0^a$ for the different compared data assimilation methods. After five iterations, the iterative methods ISM and TR-4D-EnKF provide the most accurate results among the compared implementations.

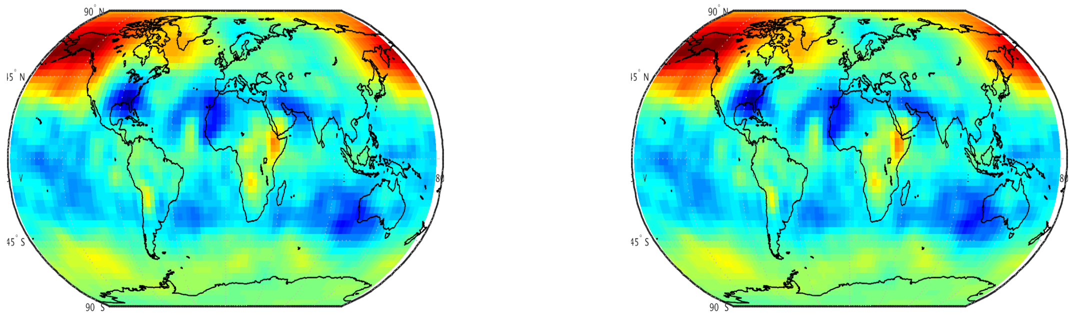


(a) True State



(b) Background

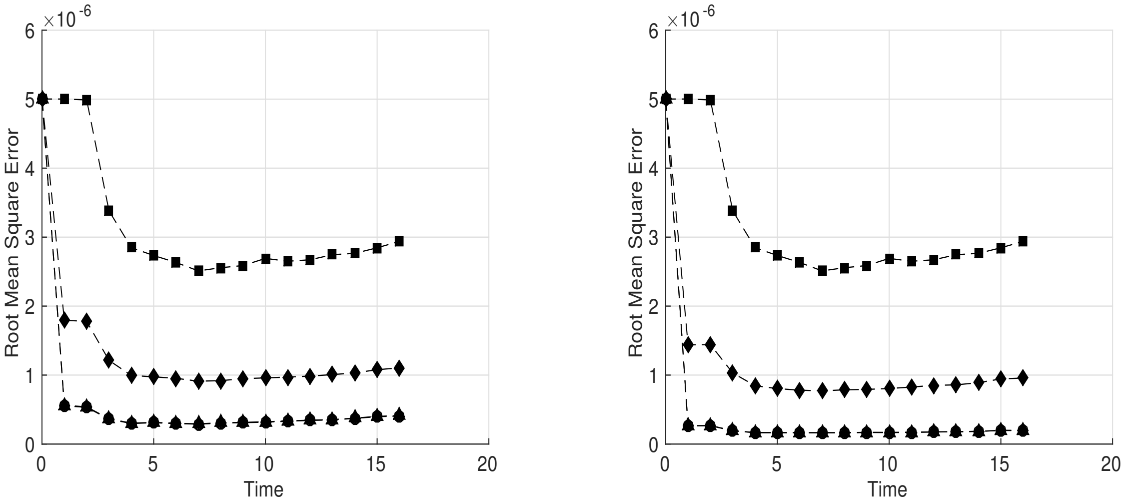
(c) POD-4D-EnKF



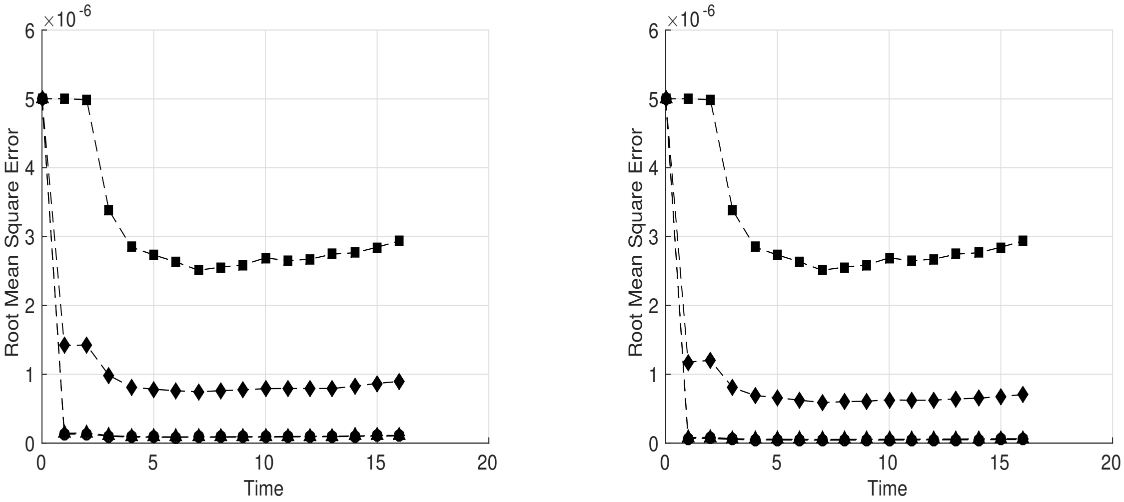
(d) ISM

(e) TR-4D-EnKF

Figure 6.4: Estimated initial states ($\mathbf{x}_0^a$) for the temperature at the Earth's surface for the Background, POD-4D-EnKF, ISM and TR-4D-EnKF. After five iterations, the iterative methods ISM and TR-4D-EnKF provide the most accurate results among the compared implementations.

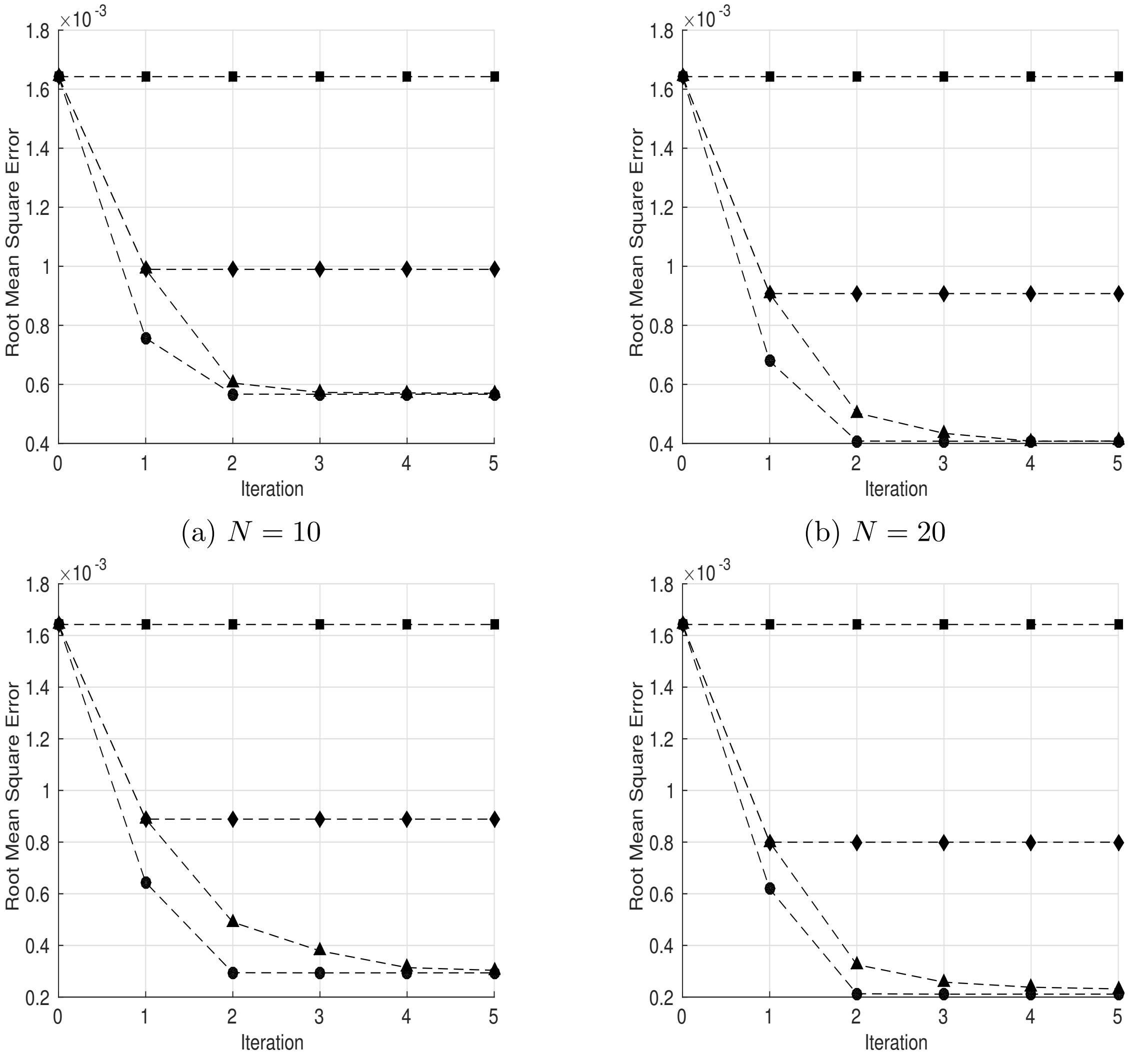


(a) $N = 10$ (b) $N = 20$



(c) $N = 40$ (d) $N = 80$

Figure 6.5: RMSE among observational times of the background state (■), POD-4D-EnKF (♦), ISM (▲) and TR-4D-EnKF (●) implementations for different ensemble sizes (N). The most accurate results are obtained by the iterative methods. The analyses reported for the iterative methods are obtained after five iterations.



(a) $N = 10$ (b) $N = 20$

(c) $N = 40$ (d) $N = 80$

Figure 6.6: RMSE among iterations of the background state (■), POD-4D-EnKF (♦), ISM (▲) and TR-4D-EnKF (•) implementations for different ensemble sizes (N). Since the POD-4D-EnKF is equivalent to one ISM iteration, its RMSE holds constant after the first iteration (for comparison purposes). The background is constant over all the iterations since it is the best estimation prior any measurement.

6.4 Conclusions

This chapter discuss the TR-4D-EnKF formulation, an ensemble-based 4D-Var data assimilation method based on the trust region framework. The proposed implementation projects

Table 6.3: Assimilation times for the compared 4D-EnKF implementations.

N	**Method**	**Assimilation Time**
10	POD-4D-EnKF	$\sim$ 10 seconds
	ISM	$\sim$ 55 seconds
	TR-4D-EnKF	$\sim$ 1 minute
20	POD-4D-EnKF	$\sim$ 12 seconds
	ISM	$\sim$ 1 minute
	TR-4D-EnKF	$\sim$ 1 minute
40	POD-4D-EnKF	$\sim$ 20 seconds
	ISM	$\sim$ 1.5 minutes
	TR-4D-EnKF	$\sim$ 1.9 minutes
80	POD-4D-EnKF	$\sim$ 1 minute
	ISM	$\sim$ 4 minutes
	TR-4D-EnKF	$\sim$ 5 minutes

the model space onto the space spanned by the deviations of the ensemble members from the mean, as is typically done in 4D-EnKF implementations. A small optimization problem is solved in the ensemble space. At each iteration a new ensemble-based surrogate model of the 4D-Var cost function is constructed, and the convergence is controlled by the trust region method. The trust region radius connects the optimal solution found in the ensemble space with the corresponding solution in the full model space. Moreover, the evolution of error statistics throughout iterations are captured by an empirical relation that uses the changes in trust region radius as a proxy for uncertainty decrease. Experimental results shows that the proposed implementation provide more accurate results than some of the best 4D-EnKF implementations available in the literature within a reasonable computational effort and a lesser number of iterations.

Chapter 7

Conclusions and Future Research Directions

This dissertation proposes efficient implementations of ensemble-based methods in data assimilation. Our formulations exploit the flow-dependency and the rank-deficiency properties of the ensemble covariance matrix, features that encapsulate the most important information needed to estimate the true background error correlations.

The iterative Sherman Morrison formula (ISMF) uses the fact that the ensemble covariance matrix is of rank at most $\mathcal{O}\left(10^2\right)$ in order to obtain an efficient implementation of the ensemble Kalman filter (EnKF). The computational effort of this method is bounded linearly with respect to the number of observations and the model dimension. The numerical stability of the iterative Sherman Morrison formula is guaranteed if the data error covariance matrix is positive definite. The computational effort of the ISMF is equivalent to those of the best EnKF implementations found in the literature. A future direction is to solve the different linear systems (involved in a step of the method) simultaneously on different processors. If the number of ensemble members matches the number of processors only linear terms (including the ensemble size) will bound the computational complexity of the method.

We proposed efficient EnKF implementations based on well-conditioned shrinkage estimates of background error covariance matrices. The resulting estimators in this context can be represented by two scalars and the matrix of member deviations of ensemble members. The computational effort involved in the analysis step is equivalent to that of the traditional EnKF where background errors are estimated via the ensemble covariance matrix. Since we have a well-conditioned estimator, samples from the prior error distribution can be taken in order to reduce the impact of sampling errors during the assimilation process. Numerical experiments reveal that the use of synthetic samples improves the accuracy of the proposed filters. The assimilation of observed components can be performed in parallel, using domain decomposition, without degrading the accuracy of the proposed methods.

The EnKF based on modified Cholesky (EnKF-MC) decomposition for inverse covariance matrix estimation exploits the localized nature of error correlations in order to compute sparse estimates of the inverse background error covariance matrix. A rigorous proof of convergence is provided for this estimator in the context of ensemble-based data assimilation is given. The estimate of the inverse background covariance matrix will converge to the true covariance when the ratio between the logarithm of the number of model components and the ensemble size is bounded by a constant. Experimental results reveal that the use of modified Cholesky in data assimilation can reduce the impact of spurious correlations. Even more, domain decomposition can be used in order to speed up the analysis using parallel computations without impacting the accuracy of the method. In future work the factorization provided by the modified Cholesky decomposition will be updated in order to account for the information coming from the observations. This will be achieved using rank-one updates with the resulting matrix being an approximation of the square root of the inverse analysis covariance matrix. This square root can be used to compute the posterior model of the error distribution and then to generate a posterior ensemble.

Strong constraint four dimensional variational assimilation (4D-Var) uses adjoint models to solve an optimization problem that computes the mode of the initial posterior distribution such as to best fit the observations within an assimilation window. In practice adjoint models are difficult to implement and maintain for real forecast models, and implementations are highly sensitive to human errors. An ensemble of model trajectories encapsulates information about error correlations at different times. Snapshots of ensemble trajectories at observation times can be used in order to obtain approximations of the analysis trajectory in the ensemble space. Since the approximate analysis is sensitive to sampling errors (e.g., the size of the sub-spaces is several times smaller than the underlying background error distribution), we formulate an ensemble 4D-Var implementation based on the trust region method (TR-4D-EnKF). The proposed implementation links the uncertainty in the ensemble with the trust region size. Experimental results reveal that the use of our proposed implementation decreases the error in the analysis estimation faster than other efficient 4D-EnKF formulations found in the current literature.

Bibliography

[AA99] Jeffrey L. Anderson and Stephen L. Anderson. A Monte Carlo Implementation of the Nonlinear Filtering Problem to Produce Ensemble Assimilations and Forecasts. Monthly Weather Review, 127(12):2741–2758, 1999.

[ABD+90] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammerling, J. Demmel, C. Bischof, and D. Sorensen. Lapack: A portable linear algebra library for high-performance computers. In Proceedings of the 1990 ACM/IEEE conference on Supercomputing, Supercomputing 90, pages 2–11, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[AC07] Jeffrey L. Anderson and Nancy Collins. Scalable Implementations of Ensemble Filter Algorithms for Data Assimilation. Journal of Atmospheric & Oceanic Technology, 2007.

[And01] Jeffrey L Anderson. An Ensemble Adjustment Kalman Filter for Data Assimilation. Monthly Weather Review, 2001.

[And12] Jeffrey L. Anderson. Localization and Sampling Error Correction in Ensemble Kalman Filter Data Assimilation. Monthly Weather Review, 140(7):2359–2371, 2012.

[BF07] Angela Benedetti and Michael Fisher. Background Error Statistics for Aerosols. Quarterly Journal of the Royal Meteorological Society, 133(623):391–405, 2007.

[BKKM04] Annalisa Bracco, Fred Kucharski, Rameshan Kallummal, and Franco Molteni. Internal variability, external forcing and climate trends in multi-decadal agcm ensembles. Climate Dynamics, 23(6):659–678, 2004.

[BL08] Peter J. Bickel and Elizaveta Levina. Regularized estimation of large covariance matrices. Ann. Statist., 36(1):199–227, 02 2008.

[Bou74] William Bourke. A multi-level spectral model. i. formulation and hemispheric integrations. Monthly Weather Review, 102(10):687–701, 1974.

[BP14] David R. Bickel and Marta Padilla. A Prior-free Framework of Coherent Inference and Its Derivation of Simple Shrinkage Estimators. Journal of Statistical Planning and Inference, 145(0):204–221, 2014.

[Bra11] Pierre Brasseur. Ensemble-based data assimilation methods. In Andreas Schiller and Gary B. Brassington, editors, Operational Oceanography in the 21st Century, pages 381–393. Springer Netherlands, 2011.

[Bue11] Mark Buehner. Evaluation of a Spatial/Spectral Covariance Localization Approach for Atmospheric Data Assimilation. Monthly Weather Review, 140(2):617–636, 2011.

[CB94] X. Carton and R. Baraille. Data Assimilation in Quasi-geostrophic Ocean Models. In OCEANS 94. Oceans Engineering for Today's Technology and Tomorrow's Preservation. Proceedings, volume 3, pages III/337 –III/346 vol.3, sep 1994.

[CCF$^+$13] G. Candiani, C. Carnevale, G. Finzi, E. Pisoni, and M. Volta. A Comparison of Reanalysis Techniques: Applying Optimal Interpolation and Ensemble Kalman Filtering to Improve Air Quality Monitoring at Mesoscale. Science of the Total Environment, 458:7–14, AUG 1 2013.

[CCFT13] K. S. Chung, W. Chang, L. Fillion, and M. Tanguay. Examination of Situation-Dependent Background Error Covariances at the Convective Scale in the Context of the Ensemble Kalman Filter. Monthly Weather Review, 141(10):3369–3387, OCT 2013.

[CEK$^+$13] Abhishek Chatterjee, Richard J. Engelen, Stephan R. Kawa, Colm Sweeney, and Anna M. Michalak. Background Error Covariance Estimation for Atmospheric CO2 Data Assimilation. Journal of Geophysical Research: Atmospheres, 118(17):10,140–10,154, 2013.

[CJAS10a] Haiyan Cheng, Mohamed Jardak, Mihai Alexe, and Adrian Sandu. A Hybrid Approach to Estimating Error Covariances in Variational Data Assimilation. Tellus A, 62(3):288–297, 2010.

[CJAS10b] Haiyan Cheng, Mohamed Jardak, Mihai Alexe, and Adrian Sandu. A Hybrid Approach to Estimating Error Covariances in Variational Data Assimilation. Tellus A, 62(3):288–297, March 2010.

[CM14] Romain Couillet and Matthew McKay. Large Dimensional Analysis and Optimization of Robust Shrinkage Covariance Matrix Estimators. Journal of Multivariate Analysis, 131(0):99–120, 2014.

[CMMS10] H. Cheng, Jardak M., Alexe M., and A. Sandu. A Hybrid Approach to Estimating Error Covariances in Variational Data Assimilation. Tellus A, 62(A):288–297, 2010.

[CNF11] X. Chen, I. M. Navon, and F. Fang. A Dual-weighted Trust-region Adaptive POD 4D-Var Applied to a Finite-element Shallow-water Equations Model. International Journal for Numerical Methods in Fluids, 65(5):520–541, 2011.

[CRV10] A. L. Custodio, H. Rocha, and L.N. Vicente. Incorporating Minimum Frobenius Norm Models in Direct Search. Computational Optimization and Applications, 46(2):265–278, 2010.

[CSV09] A.R. Conn, K. Scheinberg, and L. N. Vicente. Introduction to Derivative-Free Optimization. MPS-SIAM Book Series on Optimization, Philadelphia, Pennsylvania, 2009.

[CV07] A. L. Custodio and L. N. Vicente. Using Sampling and Simplex Derivatives in Pattern Search Methods. SIAM J. on Optimization, 18(2):537–555, May 2007.

[CWEH10] Yilun Chen, A Wiesel, Y.C. Eldar, and AO. Hero. Shrinkage Algorithms for MMSE Covariance Estimation. Signal Processing, IEEE Transactions on, 58(10):5016–5029, Oct 2010.

[CWH11] Yilun Chen, A Wiesel, and AO. Hero. Robust Shrinkage Estimation of High-Dimensional Covariance Matrices. Signal Processing, IEEE Transactions on, 59(9):4097–4107, Sept 2011.

[CWM12] Xiaohui Chen, Z.J. Wang, and M.J. McKeown. Shrinkage-to-Tapering Estimation of Large Covariance Matrices. Signal Processing, IEEE Transactions on, 60(11):5640–5656, Nov 2012.

[CZZ10] T. Tony Cai, Cun-Hui Zhang, and Harrison H. Zhou. Optimal Rates of Convergence for Covariance Matrix Estimation. The Annals of Statistics, 38(4):2118–2144, 08 2010.

[DMUN13] Victor DeMiguel, Alberto Martin-Utrera, and Francisco J. Nogales. Size Matters: Optimal Calibration of Shrinkage Estimators for Portfolio Selection. Journal of Banking & Finance, 37(8):3018 – 3034, 2013.

[DPW04] J. David, S. Philip, and A. Walter. Reduced-order Modeling: New Approaches for Computational Physics. Progress in Aerospace Sciences, 40(12):51–117, 2004.

[Eve03] Geir Evensen. The Ensemble Kalman Filter: Theoretical Formulation and Practical Implementation. Ocean Dynamics, 53(4):343–367, 2003.

[Eve06] Geir Evensen. Data Assimilation: The Ensemble Kalman Filter. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[EYO15] Takeshi ENOMOTO, Shozo YAMANE, and Wataru OHFUCHI. Simple sensitivity analysis using ensemble forecasts. Journal of the Meteorological Society of Japan. Ser. II, 93(2):199–213, 2015.

[Far78] R. W. Farebrother. A Class of Shrinkage Estimators. Journal of the Royal Statistical Society. Series B (Methodological), 40(1):pp. 47–49, 1978.

[Fra08] Andrew M. Fraser. Appendix A: Formulas for Matrices and Gaussians, chapter 7, pages 117–120. SIAM, 2008.

[FS11] Thomas J. Fisher and Xiaoqian Sun. Improved Stein-type Shrinkage Estimators for the High-dimensional Multivariate Normal Covariance Matrix. Computational Statistics & Data Analysis, 55(5):1909–1918, 2011.

[GHO99] Gene H. Golub, Per Christian Hansen, and Dianne P. O'Leary. Tikhonov regularization and total least squares. SIAM Journal on Matrix Analysis and Applications, 21(1):185–194, 1999.

[GLS14] S. Gratton, P. Laloyaux, and A. Sartenaer. Derivative-free optimization for large-scale nonlinear data assimilation problems. Quarterly Journal of the Royal Meteorological Society, 140(680):943–957, 2014.

[GM12] HumbertoC. Godinez and J.David Moulton. An Efficient Matrix-free Algorithm for the Ensemble Kalman Filter. Computational Geosciences, 16(3):565–575, 2012.

[GMC$^+$06] S. Gillijns, O.B. Mendoza, J. Chandrasekar, B. L R De Moor, D.S. Bernstein, and A Ridley. What is the Ensemble Kalman Filter and How Well Does It Work? In American Control Conference, 2006, pages 6 pp.–, June 2006.

[Han90] Per Christian Hansen. Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank. SIAM Journal on Scientific and Statistical Computing, 11(3):503–518, 1990.

[HDM$^+$13] P. L. Houtekamer, Xingxiu Deng, Herschel L. Mitchell, Seung-Jong Baek, and Normand Gagnon. Higher resolution in an operational ensemble kalman filter. Monthly Weather Review, 142(3):1143–1162, Sep 2013.

[HEE01] J. J. Hoelzemann, H. Elbern, and A. Ebel. PSAS and 4D-Var Data Assimilation for Chemical State Analysis by Urban and Rural Observation Sites. Physics and Chemistry of the Earth,Part B: Hydrology,Oceans and Atmosphere, 26(10):807–812, 2001.

[HH01] V. Hung and T. Hien. Modeling and Control of Physical Processes using Proper Orthogonal Decomposition. Mathematical and Computer Modelling, 33(13):223–236, 2001.

[HH02] V. Hung and T Hien. Proper Orthogonal Decomposition for Flow Calculations and Optimal Control in a Horizontal CVD Reactor. Quarterly of Applied Mathematics, 60(4):631–656, 2002.

[HL86] A. Hollingsworth and P. Lonnberg. The Statistical Structure of Short-range Forecast Errors as Determined from Radiosonde Data. Part I: The Wind Field. Tellus A, 38A(2):111–136, 1986.

[HS94] Isaac M. Held and Max J. Suarez. A proposal for the intercomparison of the dynamical cores of atmospheric general circulation models. Bulletin of the American Meteorological Society, 75(10):1825–1830, 1994.

[HS00] T. Hamill and C. Snyder. A Hybrid Ensemble Kalman Filter–3D Variational Analysis Scheme. Monthly Weather Review, 128(8):2905–2919, October 2000.

[Huf91] Sabine Van Huffel. Iterative algorithms for computing the singular subspace of a matrix associated with its smallest singular values. Linear Algebra and its Applications, 154:675 – 709, 1991.

[JB00] Eric P. Jiang and Michael W. Berry. Solving total least-squares problems in information retrieval. Linear Algebra and its Applications, 316(13):137 – 156, 2000. Special Issue: Conference celebrating the 60th birthday of Robert J. Plemmons.

[JFW14] Poterjoy Jonathan, Zhang Fuqing, and Yonghui Weng. The Effects of Sampling Errors on the EnKF Assimilation of Inner-Core Hurricane Observations. Monthly Weather Review, 142(4):1609–1630, 2014.

[JJR12] Christopher C. Johnson, Ali Jalali, and Pradeep D. Ravikumar. High-dimensional Sparse Inverse Covariance Estimation using Greedy Methods. In Neil D. Lawrence and Mark A. Girolami, editors, Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS-12), volume 22, pages 574–582, 2012.

[Kep00] Christian L. Keppenne. Data Assimilation into a Primitive-Equation Model with a Parallel Ensemble Kalman Filter. Monthly Weather Review, 128(6):1971–1981, 2000.

[KHE07] Misha E. Kilmer, Per Christian Hansen, and Malena I. Espaol. A projectionbased approach to generalform tikhonov regularization. SIAM Journal on Scientific Computing, 29(1):315–330, 2007.

[KM03] F. Kucharski and F. Molteni. On non-linearities in a forced north atlantic oscillation. Climate Dynamics, 21(7-8):677–687, 2003.

[KMB06a] Fred Kucharski, Franco Molteni, and Annalisa Bracco. Decadal interactions between the western tropical pacific and the north atlantic oscillation. Climate Dynamics, 26(1):79–91, 2006.

[KMB06b] Fred Kucharski, Franco Molteni, and Annalisa Bracco. Decadal interactions between the western tropical pacific and the north atlantic oscillation. Climate Dynamics, 26(1):79–91, 2006.

[KMK$^+$12] Fred Kucharski, Franco Molteni, Martin P. King, Riccardo Farneti, In-Sik Kang, and Laura Feudale. On the need of intermediate complexity general circulation models: A speedy example. Bulletin of the American Meteorological Society, 94(1):25–30, 2012.

[KMN11] Andrey Kovalenko, Trond Mannseth, and Geir Nvdal. Error estimate for the ensemble kalman filter analysis step. SIAM Journal on Matrix Analysis and Applications, 32(4):1275–1287, 2011.

[Lor86] A. C. Lorenc. Analysis methods for numerical weather prediction. Quarterly Journal of the Royal Meteorological Society, 112(474):1177–1194, 1986.

[Lor03] A. Lorenc. Modelling of error covariances by 4D-Var data assimilation. Quarterly Journal of the Royal Meteorological Society, 129(595):3167–3182, 2003.

[Lor05] Edward N. Lorenz. Designing chaotic models. Journal of the Atmospheric Sciences, 62(5):1574–1587, 2005.

[LS12] K. J. H. Law and A. M. Stuart. Evaluating data assimilation algorithms. Monthly Weather Review, 140(11):3757–3782, Jun 2012.

[LS14] William Albert Lahoz and Philipp Schneider. Data assimilation: Making sense of earth observation. Frontiers in Environmental Science, 2(16), 2014.

[LW04] Olivier Ledoit and Michael Wolf. A Well-conditioned Estimator for Large-dimensional Covariance Matrices. Journal of Multivariate Analysis, 88(2):365 – 411, 2004.

[Mai94] B. A. Mair. Tikhonov regularization for finitely and infinitely smoothing operators. SIAM Journal on Mathematical Analysis, 25(1):135–147, 1994.

[Man06] Jan Mandel. Efficient Implementation of the Ensemble Kalman Filter . Technical report, University of Colorado at Denver and Health Sciences Center, 2006.

[Map07] P. Maponi. The solution of linear systems by using the sherman–morrison formula. Linear Algebra and its Applications, 420(2–3):276 – 294, 2007.

[MNBZ15] Francesco Di Maio, Giancarlo Nicola, Emanuele Borgonovo, and Enrico Zio. Invariant methods for an ensemble-based sensitivity analysis of a passive containment cooling system of an {AP1000} nuclear power plant. Reliability Engineering & System Safety, pages –, 2015.

[Mol03a] F. Molteni. Atmospheric simulations using a gcm with simplified physical parametrizations. i: model climatology and variability in multi-decadal experiments. Climate Dynamics, 20(2-3):175–191, 2003.

[Mol03b] F. Molteni. Atmospheric simulations using a gcm with simplified physical parametrizations. i: model climatology and variability in multi-decadal experiments. Climate Dynamics, 20(2-3):175–191, 2003.

[Neu88] Andreas Neubauer. An a posteriori parameter choice for tikhonov regularization in hilbert scales leading to optimal convergence rates. SIAM Journal on Numerical Analysis, 25(6):1313–1326, 1988.

[NRS15] EliasD. Nino-Ruiz and Adrian Sandu. Ensemble kalman filter implementations based on shrinkage covariance matrix estimation. Ocean Dynamics, 65(11):1423–1439, 2015.

[NRS16] Elias D. Nino Ruiz and Adrian Sandu. A derivative-free trust region framework for variational data assimilation. J. Comput. Appl. Math., 293(C):164–179, February 2016.

[NRSA14] EliasD. Nino Ruiz, Adrian Sandu, and Jeffrey Anderson. An Efficient Implementation of the Ensemble Kalman Filter Based on an Iterative ShermanMorrison Formula. Statistics and Computing, pages 1–17, 2014.

[NRSD] Elias D. Nino-Ruiz, Adrian Sandu, and Xinwei Deng. A parallel ensemble kalman filter implementation based on modified cholesky decomposition.

[NS15] EliasD. Nino and Adrian Sandu. Variational data assimilation based on derivative-free optimization. In Sai Ravela and Adrian Sandu, editors, Dynamic Data-Driven Environmental Systems Science, volume 8964 of Lecture Notes in Computer Science, pages 239–250. Springer International Publishing, 2015.

[NSA12] Elias D. Niño, Adrian Sandu, and Jeffrey L. Anderson. An effcient implementation of the ensemble kalman filter based on iterative sherman morrison formula. Procedia Computer Science, 9:1064–1072, 2012.

[OHS+04a] Edward Ott, Brian R. Hunt, Istvan Szunyogh, Aleksey V. Zimin, Eric J. Kostelich, Matteo Corazza, Eugenia Kalnay, D. J. Patil, and James A. Yorke. A Local Ensemble Kalman Filter for Atmospheric Data Assimilation. Tellus A, 56(5):415–428, 2004.

[OHS+04b] Edward Ott, Brian R. Hunt, Istvan Szunyogh, Aleksey V. Zimin, Eric J. Kostelich, Matteo Corazza, Eugenia Kalnay, D. J. Patil, and James A. Yorke. A local ensemble kalman filter for atmospheric data assimilation. Tellus A, 56(5):415–428, 2004.

[OHS$^+$08] Edward Ott, Brian Hunt, Istvan Szunyogh, Aleksey V Zimin, Eic J. Kostelich, Matteo Corazza, Eugenia Kalnay, D. J. Patil, and James A. Yorke. A Local Ensemble Transform Kalman Filter Data Assimilation System for the NCEP Global Model. Tellus A, 60(1):113–130, 2008.

[OL99] K. Otto and E. Larsson. Iterative Solution of the Helmholtz Equation by a Second-Order Method. SIAM Journal on Matrix Analysis and Applications, 21(1):209–229, 1999.

[Oli09] P.D. Olivier. A Comparison Of Reduced Order Model Techniques. In System Theory, 2009. SSST 2009. 41st Southeastern Symposium on, pages 240–243, 2009.

[Par14] Junyong Park. Shrinkage Estimator in Normal Mean Vector Estimation Based on Conditional Maximum Likelihood Estimators . Statistics & Probability Letters, 93(0):1–6, 2014.

[RWRY11] Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional Covariance Estimation by Minimizing L1-penalized Log-determinant Divergence. Electronic Journal of Statistics, 5:935–980, 2011.

[SB08] William Sacher and Peter Bartello. Sampling errors in ensemble kalman filtering. part i: Theory. Monthly Weather Review, 136(8):3035–3049, Aug 2008.

[SC11] A. Sandu and T.F. Chai. Chemical Data Assimilation—An Overview. Atmosphere, 2(3):426–463, 2011.

[SH13] A. Sandu and Cheng H. A Subspace Approach to Data Assimilation and New Opportunities for Hybridization. International Journal for Uncertainty Quantification, submitted, 2013.

[SHC$^+$13] H. Song, I. Hoteit, B. Cornuelle, X. Luo, and A. Subramanian. An Adjoint-Based Adaptive Ensemble Kalman Filter. Monthly Weather Review, 141(10):3343–3359, October 2013.

[SHCS10] Hajoon Song, Ibrahim Hoteit, Bruce D. Cornuelle, and Aneesh C. Subramanian. An Adaptive Approach to Mitigate Background Covariance Limitations in the Ensemble Kalman Filter. Monthly Weather Review, 138(7):2825–2845, 2010.

[SO08a] Pavel Sakov and Peter Oke. A deterministic formulation of the ensemble kalman filter: an alternative to ensemble square root filters. Tellus A, 60(2), 2008.

[SO08b] Pavel Sakov and Peter R. Oke. A Deterministic Formulation of the Ensemble Kalman Filter: An Alternative to Ensemble Square Root Filters. Tellus A, 60(2):361–371, 2008.

[SO08c] PAVEL SAKOV and PETER R. OKE. A deterministic formulation of the ensemble kalman filter: an alternative to ensemble square root filters. Tellus A, 60(2):361–371, 2008.

[TAB+03] Michael K. Tippett, Jeffrey L. Anderson, Craig H. Bishop, Thomas M. Hamill, and Jeffrey S. Whitaker. Ensemble square root filters*. Monthly Weather Review, 131(7):1485–1490, Jul 2003.

[THL+13] G. Triantafyllou, I. Hoteit, X. Luo, K. Tsiaras, and G. Petihakis. Assessing a Robust Ensemble-based Kalman Filter for Efficient Ecosystem Data Assimilation of the Cretan Sea. Journal of Marine Systems, 125(SI):90–100, SEP 2013.

[Tor10] Ryan D. Torn. Ensemble-based sensitivity analysis applied to african easterly waves. Weather and Forecasting, 25(1):61–78, Feb 2010.

[TXD08] X. Tian, Z. Xie, and A. Dai. An Ensemble-based Explicit Four-dimensional Variational Assimilation Method. Journal of Geophysical Research: Atmospheres, 113(D21):n/a–n/a, 2008.

[WH12] Jeffrey S. Whitaker and Thomas M. Hamill. Evaluating methods to account for system errors in ensemble data assimilation. Monthly Weather Review, 140(9):3078–3089, Apr 2012.

[WT02] Jeffrey S. Whitaker and M. Hamill Thomas. Ensemble Data Assimilation without Perturbed Observations. Monthly Weather Review, 16(3):1913–1924, 2002.

[YMW+13] N. Yussouf, E. Mansell, L. Wicker, D. Wheatley, and D. Stensrud. The Ensemble Kalman Filter Analyses and Forecasts of the 8 May 2003 Oklahoma City Tornadic Supercell Storm Using Single- and Double-Moment Microphysics Schemes. Monthly Weather Review, 141(10):3388–3412, OCT 2013.

[Zup09] Milija Zupanski. Theoretical and practical issues of ensemble data assimilation in weather and climate. In SeonK. Park and Liang Xu, editors, Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications, pages 67–84. Springer Berlin Heidelberg, 2009.

[ZZ11] M. Zhang and F. Zhang. E4DVar: Coupling an Ensemble Kalman Filter with Four-Dimensional Variational Data Assimilation in a Limited-Area Weather Prediction Model. Monthly Weather Review, 140(2):587–600, 2011.