Computational Science Laboratory Technical Report CSL-TR-16-2015 February 3, 2015

Elias D. Nino and Adrian Sandu

"Ensemble Kalman Filter Implementations Based on Covariance Matrix Estimation"

Computational Science Laboratory
Computer Science Department
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060
Phone: (540)-231-2193
Fax: (540)-231-6075
Email: enino@vt.edu,sandu@cs.vt.edu
Web: http://csl.cs.vt.edu

Ensemble Kalman Filter Implementations Based on Covariance Matrix Estimation

Elias D. Nino-Ruiz and Adrian Sandu
Computational Science Laboratory,
Department of Computer Science,
Virginia Polytechnic Institute and State University,
Blacksburg, VA 24060, USA
enino@vt.edu, sandu@cs.vt.edu

February 3, 2015

Contents

Abstract

This paper 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 developed specifically developed to approximate high-dimensional covariance matrices using a small number of samples. 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 initial members are propagated. 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. Numerical experiments carried out with a quasi-geostrophic model show that the proposed implementations outperform current methods such as the traditional EnKF formulation, square root filters, and inflation-free EnKF implementations. The proposed implementations provide good results with small ensemble sizes (~ 10) and small percentages of observed components from the vector state. These results are similar (and in some cases better) to traditional methods using large ensemble sizes (~ 80) and large percentages of observed components. The computational times of the new implementations remain reasonably low.

Keywords: EnKF, shrinkage covariance estimation, background errors, square root filter

1   Introduction

Sequential data assimilation estimates the current unknown state $\mathbf{x}^{\text{true}} \in \mathbb{R}^{n \times 1}$ of a physical system as follows. The background (prior) state $\mathbf{x}^b \in \mathbb{R}^{n \times 1}$ is given by a physical model initialized with the best estimate of the state at

a previous time:

$$\mathbf{x}^b \equiv \mathbf{x}_{\text{current}} = \mathcal{M}_{t_{\text{previous}} \to t_{\text{current}}} \left(\mathbf{x}_{\text{previous}} \right) . \tag{1}$$

The background errors are assumed to be unbiased and normally distributed

$$\boldsymbol{\xi} = \mathbf{x}^b - \mathbf{x}^{\text{true}} \in \mathcal{N} \left(\mathbf{0}_n, \mathbf{B} \right) \tag{2}$$

where n is the dimension of the model state, $\mathbf{0}_q$ is the vector in the q-dimensional space whose components are all zeros, and $\mathbf{B} \in \mathbb{R}^{n \times n}$ is the unknown background error covariance matrix. Observations of the true state are taken

$$\mathbf{y} = \mathcal{H} \left(\mathbf{x}^{\text{true}} \right) + \boldsymbol{\epsilon} \in \mathbb{R}^{m \times 1} , \tag{3}$$

where $\mathcal{H} : \mathbb{R}^{n \times 1} \to \mathbb{R}^{m \times 1}$ is the observation operator, and m is the number of observed components. The observational errors are assumed to be normally distributed $\boldsymbol{\epsilon} \sim \mathcal{N} \left(\mathbf{0}_m, \mathbf{R} \right)$, where $\mathbf{R} \in \mathbb{R}^{m \times m}$ is the observation error covariance matrix that is assumed to possess a simple structure (e.g., block diagonal) and therefore its inverse can be easily computed.

Under the Gaussian assumption on data and model errors the negative logarithms of the a posteriori probability density is the 3D-Var cost function [31]:

$$\mathcal{J} \left(\mathbf{x} \right) = \frac{1}{2} \cdot \left\| \mathbf{x} - \mathbf{x}^b \right\|_{\mathbf{B}^{-1}}^2 + \frac{1}{2} \cdot \left\| \mathbf{y} - \mathcal{H} \left(\mathbf{x} \right) \right\|_{\mathbf{R}^{-1}}^2 . \tag{4}$$

The maximum likelihood estimate of the state is obtained by minimizing the cost function (4), i.e., the analysis state $\mathbf{x}^a \in \mathbb{R}^{n \times 1}$ is the solution of the following optimization problem:

$$\mathbf{x}^a = \arg \min_{\mathbf{x}} \mathcal{J} \left(\mathbf{x} \right) . \tag{5}$$

The solution of (5) over the subspace spanned by the ensemble members is:

$$\mathbf{x}^a = \mathbf{x}^b + \boldsymbol{\Psi} \cdot \boldsymbol{\alpha_B} , \tag{6}$$

where $\boldsymbol{\Psi} = \mathbf{B}^{1/2}$ is a set of basis vectors satisfying $\boldsymbol{\Psi} \boldsymbol{\Psi}^T = \mathbf{B}$, and the vector of weights is given by

$$\boldsymbol{\alpha_B} = \mathbf{V_B}^T \cdot \left(\mathbf{R} + \mathbf{V_B} \cdot \mathbf{V_B}^T \right)^{-1} \cdot \left(\mathbf{y} - \mathcal{H}(\mathbf{x}^b) \right) , \tag{7}$$

3

where $\mathbf{V_B} = \mathbf{H} \cdot \mathbf{\Psi} \in \mathbb{R}^{m \times n}$ and $\mathbf{H} = \mathcal{H}'$. By propagating in time the analysis (6) a new prior (background) state is obtained for the next assimilation cycle.

Since $\mathbf{B}$ is unknown the direct use of (6) is infeasible. Methods for estimating the background error covariance matrix have been developed [13, 21]. However, in the absence of prior information about the true structure of $\mathbf{B}$ biased estimators are often obtained. Buhener [7] discusses the impact of having biased estimators for the background errors in the assimilation process. AIn the context of sequential data assimilation an ensemble of model realizations [18] is built in order to represent the background error statistics:

$$\mathbf{X}^b = \left[\mathbf{x}_1^b, \, \mathbf{x}_2^b, \, \dots, \mathbf{x}_N^b \right] \in \mathbb{R}^{n \times N} , \tag{8}$$

where $\mathbf{x}_i^b \in \mathbb{R}^{n \times 1}$ is the i-th ensemble member (model run) in the *background stage*. Estimates of $\mathbf{x}^b$ and $\mathbf{B}$ are obtained via the empirical moments of the ensemble (8)

$$\mathbf{x}^b \approx \overline{\mathbf{x}}^b = \frac{1}{N} \cdot \sum_{i=1}^{N} \mathbf{x}_i^b \in \mathbb{R}^{n \times 1} , \tag{9}$$

and

$$\mathbf{B} \approx \mathbf{P}^b = \mathbf{S} \cdot \mathbf{S}^T \in \mathbb{R}^{n \times n} , \tag{10}$$

where the matrix of scaled member deviations is

$$\mathbf{S} = \frac{1}{\sqrt{N-1}} \cdot \left[\mathbf{X}^b - \overline{\mathbf{x}}^b \otimes \mathbf{1}_N^T \right] \in \mathbb{R}^{n \times N}, \tag{11}$$

where $\mathbf{1}_q$ is the vector in the q-th dimensional space whose components are all ones. The set of basis vectors (11) does not span the full space of model errors. By replacing the estimators (9) and (10) in (6), the analysis can be approximated as follows:

$$\mathbf{x}^a \approx \overline{\mathbf{x}}^a = \overline{\mathbf{x}}^b + \mathbf{S} \cdot \boldsymbol{\alpha} \in \mathbb{R}^{n \times 1} , \tag{12}$$

where

$$\boldsymbol{\alpha} = \mathbf{V} \cdot \left(\mathbf{R} + \mathbf{V} \cdot \mathbf{V}^T \right)^{-1} \cdot \left(\mathbf{y} - \mathcal{H}(\overline{\mathbf{x}}^b) \right) \in \mathbb{R}^{N \times 1} , \tag{13}$$

and $\mathbf{V} = \mathbf{H} \cdot \mathbf{S} \in \mathbb{R}^{m \times N}$. The analysis state (12) is the optimal solution in the space spanned by the ensemble members (8), but in general it is not

optimal in the model space. Since $n \gg N$ information contained in (7) is not represented by the set of basis vectors $\mathbf{S}$. From another point of view, the ensemble covariance matrix $\mathbf{P}^b$ is rank deficient and therefore there are insufficient degrees of freedom to explain the full error. This problem can be alleviate by making use of localization techniques [9, 14, 25]. However, since the structure of $\mathbf{B}$ remains unknown, the use of localization generally increases the bias in the background error estimation.

There is an opportunity to avoid the intrinsic need of inflation in ensemble based methods by replacing the covariance matrix (10) with a more accurate and well-conditioned estimate of $\mathbf{B}$. We do not want to impose any kind of structure on $\mathbf{B}$ since it will make our approach sensitive to problems faced by current implementations. Instead, we seek to capture most of the information contained in the ensemble. Shrinkage covariance estimators developed to estimate high-dimensional covariance matrices from a small number of samples [30] fit very well in the context of sequential data assimilation.

The remaining part of the paper is organized as follows. Section 2 reviews ensemble based data assimilation and shrinkage covariance estimation. In section 3 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. Section 5 summarizes the conclusions of this work.

2   Background

In this section we review relevant concepts with regard to shrinkage covariance estimation and ensemble based methods in sequential data assimilation.

2.1   Covariance 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 [14, 24] to sequential [9, 41] data assimilation rely on accurately estimated covariance matrices.

Let $\{s_1, s_2, \ldots, s_N\}$ be a sample of independent identical distributed n-dimensional Gaussian vectors

$$s_i \sim \mathcal{N}(\mathbf{0}_n, \mathbf{Q}) \in \mathbb{R}^{n \times 1}.$$

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

$$\mathbf{C}_s = \frac{1}{N-1} \cdot \sum_{i=1}^{N} s_i \otimes s_i^T \in \mathbb{R}^{n \times n}. \tag{14}$$

$\mathbf{C}_s$ is the maximum likelihood estimator when it is invertible [30]. 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 $\{s_i\}_{1 \leq i \leq N}$. In order to improve estimation of covariance matrices many methods have been proposed in the literature based on tapering procedures [8, 10], minimizing the log-determinant divergence [36], and greedy methods [27]. Another class of well-conditioned estimators is based on shrinkage approximations [4, 15, 16, 19, 20, 35]. 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 (14). To better understand this assume that the components of s_i are uncorrelated for $1 \leq i \leq N$. A simple estimate of $\mathbf{Q}$ is given by

$$\mathbf{T} = \frac{\text{tr}\,(\mathbf{C}_s)}{n} \cdot \mathbf{I}_{n \times n},$$

where $\mathbf{I}_{n \times n}$ 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

$$\widehat{\mathbf{C}} = \gamma \cdot \mathbf{T} + (1 - \gamma) \cdot \mathbf{C}_s \in \mathbb{R}^{n \times n}, \tag{15}$$

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

$$\mathbb{E}\left[\left\|\widehat{\mathbf{C}} - \mathbf{Q}\right\|_F^2\right] \tag{16}$$

is minimized, where $\|\bullet\|_F$ denotes the Frobenius norm. There are many shrinkage based estimators derived from the minimization of (16) subject to

(15). We restrict our exploration to three well-accepted methods: the Ledoit and Wolf estimator [30], the Rao-Blackwell Ledoit and Wolf estimator [11] and the oracle approximating shrinkage estimator [11, 12].

The distribution-free Ledoit and Wolf (LW) estimator [30] 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 [30]. The optimal γ value proposed by this estimator is

$$\gamma_{LW} = \min \left(\frac{\sum_{i=1}^{N} \left\| \mathbf{C}_s - s_i \otimes s_i^T \right\|_F^2}{N^2 \cdot \left[\operatorname{tr}\left(\mathbf{C}_s^2\right) - \frac{\operatorname{tr}^2(\mathbf{C}_s)}{n} \right]}, \ 1 \right) \tag{17}$$

and the LW estimator $\widehat{\mathbf{C}}_{LW}$ is obtained by using γ_{LW} in (15).

The Rao-Blackwell Ledoit and Wolf (RBLW) estimator [11, 12] provably improves the LW method 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 γ is

$$\gamma_{RBLW} = \min \left(\frac{\frac{N-2}{n} \cdot \operatorname{tr}\left(\mathbf{C}_s^2\right) + \operatorname{tr}^2\left(\mathbf{C}_s\right)}{(N+2) \cdot \left[\operatorname{tr}\left(\mathbf{C}_s^2\right) - \frac{\operatorname{tr}^2(\mathbf{C}_s)}{n} \right]}, \ 1 \right) \tag{18}$$

and the corresponding estimator $\widehat{\mathbf{C}}_{RBLW}$ is obtained by replacing (18) in the equation (15). In addition, in [12, Theorem 2], it is proven that

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

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

The oracle approximating shrinkage (OAS) estimator [11] is an iterative approximation of the unimplementable oracle method [11, Section 3]. The optimal γ at each iteration j is given by

$$\widehat{\mathbf{C}}_j = \gamma_j \cdot \mathbf{T} + (1 - \gamma_j) \cdot \mathbf{C}_s, \tag{19a}$$

$$\gamma_{j+1} = \frac{\left(1 - \frac{2}{n}\right) \cdot \operatorname{tr}\left(\widehat{\mathbf{C}}_j \cdot \mathbf{C}_s\right) + \operatorname{tr}^2\left(\widehat{\mathbf{C}}_j\right)}{\left(N + 1 - \frac{2}{n}\right) \cdot \operatorname{tr}\left(\widehat{\mathbf{C}}_j \cdot \mathbf{C}_s\right) + \left(1 - \frac{N}{n}\right) \cdot \operatorname{tr}^2\left(\widehat{\mathbf{C}}_j\right)} \tag{19b}$$

where the initial estimator $\widehat{\mathbf{C}}_0$ can be any estimator (i.e., the LW, the RBLW, or even the sample covariance matrix).

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.

2.2 Sequential data assimilation methods

Ensemble data assimilation methods are widely used in applications to weather, oceanography, and climatology [42]. These methods represent the background error statistics by the empirical moments of the ensemble (9) and (10)[22]. The trajectory of each ensemble member and the dispersion of the overall ensemble around the background state provide meaningful information about the background error distribution. One of the most important advantages of ensemble DA is the flow-dependent forecast error covariance matrix [7]. When observations are available the analysis state (12) is computed. The next step is to generate an ensemble which describes the analysis uncertainty around this optimal state. We briefly discuss three implementations that achieve this, the ensemble square root filter, the ensemble transform Kalman filter, and the (basic) ensemble Kalman filter.

In ensemble square root filters [39], the new analysis ensemble in the analysis is built as follows:

$$\mathbf{X}^a = \overline{\mathbf{x}}^a \otimes \mathbf{1}_N^T + \mathbf{S} \cdot \left[\mathbf{I}_{N \times N} - \mathbf{V}^T \cdot \left(\mathbf{R} + \mathbf{V} \cdot \mathbf{V}^T \right)^{-1} \cdot \mathbf{V} \right]^{1/2} \in \mathbb{R}^{n \times N}. \quad (20)$$

As expected the analysis ensemble members live in the subspace space spanned by the background ensemble, this is, the space spanned by the columns of $\mathbf{S}$. All possible information that can be obtained from the model states is contained in this set of basis vectors.

The covariance matrix in the observation space is

$$\mathbf{W}_{\text{obs}} = \mathbf{R} + \mathbf{V} \cdot \mathbf{V}^T \in \mathbb{R}^{m \times m}, \quad (21)$$

the linear system $\mathbf{W}_{\text{obs}} \cdot \mathbf{Z}_{\mathbf{V}} = \mathbf{V} \in \mathbb{R}^{m \times N}$ can be solved via the iterative

Sherman Morrison formula (ISMF) [32]

$$\mathbf{h}^{(k)} = \left(1 + \mathbf{v}_k^T \cdot \mathbf{u}_k^{(k-1)}\right)^{-1} \cdot \mathbf{u}_k^{(k-1)} \in \mathbb{R}^{m \times 1} \tag{22a}$$

$$\mathbf{Z}^{(k)} = \mathbf{Z}^{(k-1)} - \mathbf{h}^{(k)} \cdot \left[\mathbf{v}_k^T \cdot \mathbf{Z}^{(k-1)}\right] \in \mathbb{R}^{m \times N} \tag{22b}$$

$$\mathbf{U}^{(k)} = \mathbf{U}^{(k-1)} - \mathbf{h}^{(k)} \cdot \left[\mathbf{v}_k^T \cdot \mathbf{U}^{(k-1)}\right] \in \mathbb{R}^{m \times N} \tag{22c}$$

for $1 \leq k \leq N$, where $\mathbf{Z}^{(0)} = \mathbf{R}^{-1} \cdot \left(\mathbf{y} - \mathbf{H} \cdot \overline{\mathbf{x}}^b\right)$ and $\mathbf{U}^{(0)} = \mathbf{R}^{-1} \cdot \mathbf{V}$. The implementation (22) requires no more than $\mathcal{O}\left(N^2 \cdot m\right)$ long computations, where $\mathbf{v}_k$ and $\mathbf{u}_k^{(k-1)}$ are the k-th column of the matrices $\mathbf{V}$ and $\mathbf{U}_k^{(k-1)}$, respectively. Moreover, by applying the singular value decomposition to

$$\mathbf{V}^T \cdot \mathbf{Z_V} = \mathbf{U_Z} \cdot \mathbf{\Sigma_Z} \cdot \mathbf{U_Z}^T \in \mathbb{R}^{N \times N} \in \mathbb{R}^{N \times N}, \tag{23}$$

the square root in (20)

$$\mathbf{\Gamma} = \mathbf{U_Z} \cdot [\mathbf{I}_{N \times N} - \mathbf{\Sigma_Z}]^{1/2} \cdot \mathbf{U_Z}^T,$$

can be efficiently computed with no more than $\mathcal{O}\left(N^3\right)$ long computations.

Another widely used square root filter implementation is the ensemble transform Kalman filter (EnTKF) [33]. By making use of the matrix identity

$$\mathbf{I} - \mathbf{V} \cdot \left[\mathbf{R} + \mathbf{V} \cdot \mathbf{V}^T\right]^{-1} \cdot \mathbf{V}^T = \left[\mathbf{I} + \mathbf{V} \cdot \mathbf{R} \cdot \mathbf{V}^T\right]^{-1} \in \mathbb{R}^{m \times m}, \tag{24}$$

and the singular value decomposition

$$\mathbf{V}^T = \mathbf{U_V} \cdot \widehat{\mathbf{\Sigma}}_{\mathbf{V}} \cdot \mathbf{V_V}^T \in \mathbb{R}^{N \times m}, \tag{25}$$

the analysis state (12) can be written as follows

$$\mathbf{x}^a = \overline{\mathbf{x}}^b + \mathbf{S} \cdot \boldsymbol{\beta} \tag{26}$$

with the optimal weights $\boldsymbol{\beta} \in \mathbb{R}^{N \times N}$ given by

$$\boldsymbol{\beta} = \mathbf{U_V} \cdot \widehat{\mathbf{\Sigma}}_{\mathbf{V}} \cdot \left(\mathbf{I}_{N \times N} + \widehat{\mathbf{\Sigma}}_{\mathbf{V}}^T \cdot \widehat{\mathbf{\Sigma}}_{\mathbf{V}}\right)^{-1} \cdot \mathbf{V_V}^T \cdot \sqrt{\mathbf{R}} \cdot \left(\mathbf{y} - \mathbf{H} \cdot \overline{\mathbf{x}}^b\right). \tag{27}$$

The new ensemble members are built as follows

$$\mathbf{X}^a = \overline{\mathbf{x}}^a \otimes \mathbf{1}_N^T + \mathbf{S} \cdot \mathbf{U_V} \cdot \left[\mathbf{I}_{N \times N} + \widehat{\mathbf{\Sigma}}_{\mathbf{V}} \cdot \widehat{\mathbf{\Sigma}}_{\mathbf{V}}^T\right]^{1/2} \cdot \mathbf{U_V}^T. \tag{28}$$

9

In the ensemble Kalman filter (EnKF) [17] each ensemble member (8) and the observations (3) are treated as random variables, and the i-th ensemble member is updated as follows:

$$\mathbf{x}_i^a = \mathbf{x}_i^b + \mathbf{S} \cdot \boldsymbol{\beta}_i \in \mathbb{R}^{n \times 1}, \quad i = 1, \ldots, N, \tag{29}$$

where the optimal weights for i-th ensemble member are given by

$$\boldsymbol{\beta}_i = \cdot \mathbf{V}^T \cdot \left(\mathbf{R} + \mathbf{V} \cdot \mathbf{V}^T \right)^{-1} \cdot \left[\mathbf{y}_i^s - \mathbf{H} \cdot \mathbf{x}_i^b \right] \in \mathbb{R}^{N \times 1} \tag{30}$$

and

$$\mathbf{y}_i^s \sim \mathcal{N}\left(\mathbf{y}, \ \mathbf{R} \right), \tag{31}$$

The addition of the perturbed observations (31) in the analysis provides asymptotically correct analysis-error covariance estimates for large ensemble sizes and makes the formulation of the EnKF statistical consistent [40]. However, it also has been proven that the inclusion of perturbed observations introduces sampling errors in the assimilation [2, 29].

One of the important problems faced by current ensemble based methods is filter divergence due to the insufficient degrees of freedom ($N \ll n$). To alleviate this deficiency localization is used to impose structure on the sample covariance matrix (10) according to the physics of the model. Intuitively, the correlations between individual model variable errors decays with distance, e.g., exponentially:

$$\boldsymbol{\rho}_{ij} = \exp\left(\frac{-d\left(i, j\right)^2}{2 \cdot L^2} \right), \tag{32}$$

where $d\left(i, j\right)$ is the physical distance between the locations of the i-th and j-th model components, and L is the localization radius. The localized covariance matrix is obtained as

$$\mathbf{P}_{\text{Loc}}^b = \boldsymbol{\rho} \circ \mathbf{P}^b \in \mathbb{R}^{n \times n}. \tag{33}$$

where $\circ$ is the Schur product. This method can be impractical since it relies on the explicit computation of $\mathbf{P}^b$. Moreover, there are no guaranties that this method captures the true structure of $\mathbf{B}$. More sophisticated covariance estimation methods have been proposed in the context of data assimilation. A classic approximation is the Hollingworth and Lonnberg method

[25] 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 spaces, requires uniform observing network (not the case in practice), and the resulting statistics are biased towards data-dense areas. Another method has been proposed by Benedetti and Fisher [3] 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. Since the structure of $\mathbf{B}$ remains unknown, assumptions made about its structure may increase the bias in the estimate. Furthermore, the balance of variables in the model with some physical meaning can be disturbed when localization is utilized [34].

A different approach is based on the 3D-Var cost function in the ensemble space. Any vector $\mathbf{x} \in \mathbb{R}^{n \times 1}$ in the ensemble subspace can be written as

$$\mathbf{x} = \overline{\mathbf{x}}^b + \mathbf{U} \cdot \boldsymbol{\alpha} \tag{34}$$

where $\mathbf{U}$ is the matrix of anomalies

$$\mathbf{U} = \left[\mathbf{x}_1^b - \overline{\mathbf{x}}^b, \, \mathbf{x}_2^b - \overline{\mathbf{x}}^b, \, \ldots, \, \mathbf{x}_N^b - \overline{\mathbf{x}}^b \right] \in \mathbb{R}^{n \times N} , \tag{35}$$

and $\boldsymbol{\alpha} \in \mathbb{R}^{N \times 1}$ is a vector to be determined. The columns of $\mathbf{U}$ and $\mathbf{S}$ span the same space. Using (34) the 3D-Var cost function (4) in the ensemble space reads

$$\mathcal{J}_{\mathrm{ens}} \left(\boldsymbol{\alpha} \right) = \frac{1}{2} \cdot \| \mathbf{U} \cdot \boldsymbol{\alpha} \|_{\mathbf{B}^{-1}}^2 + \frac{1}{2} \cdot \| \cdot \mathbf{Q} \cdot \boldsymbol{\alpha} \|_{\mathbf{R}^{-1}}^2 , \tag{36}$$

where the optimal value of the control variable $\boldsymbol{\alpha}$

$$\boldsymbol{\alpha}^* = \arg \min_{\boldsymbol{\alpha}} \mathcal{J}_{\mathrm{ens}} \left(\boldsymbol{\alpha} \right) , \tag{37}$$

provides the analysis state in (34)

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^b + \mathbf{U} \cdot \boldsymbol{\alpha}^* . \tag{38}$$

Two recent formulations based on this approximation are the finite size anddual ensemble Kalman filters [6]. These formulation avoid the intrinsic needed of inflation by choosing Jeffrey's prior for background errors:

$$\mathcal{P} \left(\mathbf{x}^b, \mathbf{B} \right) = \mathcal{P}_J \left(\mathbf{x}^b \right) \cdot \mathcal{P}_J \left(\mathbf{B} \right) ,$$

where the parameters $\mathbf{x}^b$ and $\mathbf{B}$ are assumed to be independent.

In the case of the finite size ensemble Kalman filter (EnKF-N) the cost function in the ensemble space reads

$$\mathcal{J}_{\text{ens}}^{\text{FN}}(\boldsymbol{\alpha}) = \frac{1}{2} \left\| \mathbf{y} - \mathcal{H}\left(\overline{\mathbf{x}}^b + \mathbf{U} \cdot \boldsymbol{\alpha}\right) \right\|_{\mathbf{R}^{-1}}^2 + \frac{N}{2} \cdot \log\left(1 + \frac{1}{N} + \|\boldsymbol{\alpha}\|^2\right). \quad (39)$$

Minimization of this cost function provides the optimal weights in the ensemble space

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \mathcal{J}_{\text{ens}}^{\text{FN}}(\boldsymbol{\alpha}), \quad (40)$$

and the analysis is computed via (38). The projection of the analysis covariance matrix on the ensemble space is approximated by the inverse of the Hessian of (39) at the optimal value (40). The Hessian reads:

$$\begin{aligned}
\nabla_{\boldsymbol{\alpha},\boldsymbol{\alpha}}^2 \mathcal{J}_{\text{ens}}^{\text{FN}}(\boldsymbol{\alpha}) &= [\mathbf{H} \cdot \mathbf{U}]^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H} \cdot \mathbf{U} \\
&+ N \cdot \frac{\left(1 + \frac{1}{N} + \|\boldsymbol{\alpha}\|^2\right) \cdot \mathbf{I}_{N \times N} - 2 \cdot \boldsymbol{\alpha} \cdot \boldsymbol{\alpha}^T}{\left(1 + \frac{1}{N} + \|\boldsymbol{\alpha}\|^2\right)^2} \in \mathbb{R}^{N \times N}.
\end{aligned} \quad (41)$$

The analysis ensemble is generated as follows:

$$\mathbf{X}^a = \overline{\mathbf{x}}^a \otimes \mathbf{1}_N^T + \mathbf{U} \cdot \left\{(N-1) \cdot \left[\nabla_{\boldsymbol{\alpha},\boldsymbol{\alpha}}^2 \mathcal{J}_{\text{ens}}^{\text{FN}}(\boldsymbol{\alpha}^*)\right]^{-1}\right\}^{1/2} \cdot \boldsymbol{\Phi} \in \mathbb{R}^{n \times N}, \quad (42)$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{N \times N}$ is an arbitrary orthogonal matrix which preserves the ensemble mean (9).

Another approach is based on the dual formulation of the cost function (41) [6]

$$\mathcal{D}_{\text{ens}}^{\text{DU}}(\zeta) = \left\| \mathbf{y} - \mathbf{H} \cdot \mathbf{x}^b \right\|_{\mathbf{W}_\zeta^{-1}}^2 + \zeta \cdot \left(1 + \frac{1}{N}\right) + N \cdot \log\left(\frac{N}{\zeta}\right) - N, \quad (43)$$

where the weighted covariance matrix $\mathbf{W}_\zeta$ reads

$$\mathbf{W}_\zeta = \mathbf{R} + \frac{1}{\zeta} \cdot \mathbf{V} \cdot \mathbf{V}^T \in \mathbb{R}^{m \times m}. \quad (44)$$

The dual optimization problem is one-dimansional

$$\zeta^* = \arg\min_{\zeta \in \left(0, \frac{N}{1+\frac{1}{N}}\right]} \mathcal{D}_{\text{ens}}^{\text{DU}}(\zeta) \qquad \text{subject to} \quad \zeta \in \left(0, \frac{N}{1+\frac{1}{N}}\right]. \quad (45)$$

The optimal state is computed as follows:

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^b + \mathbf{U} \cdot \left[\mathbf{V}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{V} + \zeta^* \cdot \mathbf{I}_{N \times N}\right]^{-1} \cdot \mathbf{V}^T \cdot \mathbf{R}^{-1} \cdot \left[\mathbf{y} - \mathbf{H} \cdot \overline{\mathbf{x}}^b\right] , \quad (46)$$

and the following analysis ensemble is built:

$$\mathbf{X}^a = \overline{\mathbf{x}}^a \otimes \mathbf{1}_N^T + \mathbf{U} \cdot \left\{(N-1) \cdot \left[\mathbf{V}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{V} + \zeta^* \cdot \mathbf{I}_{N \times N}\right]^{-1}\right\}^{1/2} \cdot \mathbf{\Phi}. \quad (47)$$

In this paper we consider a different representation of the background error statistics by making use of shrinkage covariance estimation. The idea is not to impose any structure on $\mathbf{P}^b$ but to obtain a well-conditioned estimator $\widehat{\mathbf{B}}$ of the background error covariance matrix $\mathbf{B}$ wherein using all the possible information brought from the ensemble members. Samples from the distribution $\mathcal{N}\left(\overline{\mathbf{x}}^b, \widehat{\mathbf{B}}\right)$ are taken in order to better represent the error statistics and to increase the number of degrees of freedom. Two novel EnKF implementations based on the Rao-Blackwell Ledoit and Wolf estimator are presented in the next section.

3 Ensemble Filters Based on Shrinkage Covariance Estimators

In this section, we propose two efficient implementations of the EnKF based on the RBLW estimator (18). As mentioned before, we do not impose any kind of structure on $\mathbf{P}^b$ since the information brought by the ensemble members is more than only background errors

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

where $\mathbf{Q}$ is the covariance of model errors and $\mathbf{C} \in \mathbb{R}^{n \times n}$ is the covariance matrix of additional errors whose sources are unknown for us. 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 (18) to build a covariance matrix that captures all error correlations. The standard form of this estimatordepends on the explicit representation of $\mathbf{P}^b$. The efficient implementation for high-dimensional covariance matrices presented in section 3.1 avoids the explicit computation of $\mathbf{P}^b$. Section 3.2 discusses two EnKF implementations based on the RBLW estimator. Section 3.3 develops an efficient sampling method in high dimensions for drawing

samples from the prior error distribution based on the RBLW estimate. Finally, section 3.4 discusses the similarities and differences between the two proposed implementations.

3.1 RBLW estimator for covariance matrices in high-dimensions

Consider the sample covariance matrix (10). In the context of data assimilation the RBLW estimator (15),(18) reads

$$\widehat{\mathbf{B}} = \gamma_{\widehat{\mathbf{B}}} \cdot \mu_{\mathbf{B}} \cdot \mathbf{I} + (1 - \gamma_{\widehat{\mathbf{B}}}) \cdot \mathbf{P}^b \in \mathbb{R}^{n \times n}, \tag{48a}$$

where

$$\mu_{\widehat{\mathbf{B}}} = \frac{\mathrm{tr}\left(\mathbf{P}^b\right)}{n}, \tag{48b}$$

$$\gamma_{\mathbf{B}} = \min\left(\frac{\frac{N-2}{n} \cdot \mathrm{tr}\left(\left[\mathbf{P}^b\right]^2\right) + \mathrm{tr}^2\left(\mathbf{P}^b\right)}{(N+2) \cdot \left[\mathrm{tr}\left(\left[\mathbf{P}^b\right]^2\right) - \frac{\mathrm{tr}^2\left(\mathbf{P}^b\right)}{n}\right]}, 1\right). \tag{48c}$$

Since the dimension of the model state is high ($n \sim \mathcal{O}\left(10^7\right)$), the direct computation of (52) is impractical as it requires the explicit representation of the sample covariance matrix $\mathbf{P}^b$. An alternative manner to compute $\mathrm{tr}\left(\mathbf{P}^b\right)$ and $\mathrm{tr}\left([\mathbf{P}^b]^2\right)$ is proposed. Consider the eigenvalue decomposition of $\mathbf{P}^b$

$$\mathbf{P}^b = \mathbf{U}_{\mathbf{P}^b} \cdot \mathbf{\Sigma}_{\mathbf{P}^b} \cdot \mathbf{U}_{\mathbf{P}^b}^T \in \mathbb{R}^{n \times n}, \tag{49}$$

where $\mathbf{\Sigma}_{\mathbf{P}^b} \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose diagonal components σ_i, for $1 \leq i \leq n$, 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 definitiont $\mathrm{tr}\left(\mathbf{P}^b\right) = \sum_{i=1}^n \sigma_i$ and $\mathrm{tr}\left(\left[\mathbf{P}^b\right]^2\right) = \sum_{i=1}^n \sigma_i^2$. Since there are only $N-1$ eigenvalues different from zero we obtain:

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

and the computations in the set of equations (48) can be efficiently performed whenever the first $N-1$ eigenvalues of $\mathbf{P}^b$ can be easily obtained. Consider the singular value decomposition (SVD) for the set of basis vectors (11)

$$\mathbf{S} = \mathbf{U_S} \cdot \widehat{\boldsymbol{\Sigma}}_{\mathbf{S}} \cdot \mathbf{V_S}^T \in \mathbb{R}^{n \times N}, \tag{50}$$

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

$$\mathrm{tr}\left(\mathbf{P}^b\right) = \sum_{i=1}^{N-1} \sigma_i = \sum_{i=1}^{N-1} \widehat{\sigma}_i^{\ 2},$$

$$\mathrm{tr}\left(\left[\mathbf{P}^b\right]^2\right) = \sum_{i=1}^{N-1} \sigma_i^2 = \sum_{i=1}^{N-1} \widehat{\sigma}_i^{\ 4}.$$

The computational effort of the SVD decomposition (50) is $\mathcal{O}\left(N^2 \cdot n\right)$. The traces in (48) 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 $1 \leq i \leq N-1$. The parameter values in (48) are computed as follows:

$$\mu_{\widehat{\mathbf{B}}} = \frac{\sum_{i=1}^{N-1} \widehat{\sigma}_i^{\ 2}}{n}, \tag{51a}$$

$$\gamma_{\mathbf{B}} = \min\left(\frac{\frac{N-2}{n} \cdot \sum_{i=1}^{N-1} \widehat{\sigma}_i^{\ 4} + \left[\sum_{i=1}^{N-1} \widehat{\sigma}_i^{\ 2}\right]^2}{(N+2) \cdot \left[\sum_{i=1}^{N-1} \widehat{\sigma}_i^{\ 4} - \frac{\left[\sum_{i=1}^{N-1} \widehat{\sigma}_i^{\ 2}\right]^2}{n}\right]}, 1\right). \tag{51b}$$

With $\varphi = \mu_{\widehat{\mathbf{B}}} \cdot \gamma_{\widehat{\mathbf{B}}}$ and $\delta = 1 - \gamma_{\widehat{\mathbf{B}}}$ the estimated covariance matrix (48) is

$$\widehat{\mathbf{B}} = \varphi \cdot \mathbf{I}_{n \times n} + \delta \cdot \mathbf{S} \cdot \mathbf{S}^T \in \mathbb{R}^{n \times n}. \tag{52}$$

3.2 EnKF implementations based on the RBLW estimator

By replacing the estimated error covariance matrix (52) in (29), the EnKF analysis in matrix form becomes

$$\mathbf{X}^a = \mathbf{X}^b + \widehat{\mathbf{B}} \cdot \mathbf{H}^T \cdot \left(\mathbf{R} + \mathbf{H} \cdot \widehat{\mathbf{B}} \cdot \mathbf{H}^T\right)^{-1} \cdot \mathbf{D} \in \mathbb{R}^{n \times N}, \tag{53}$$

where the matrix of innovations $\mathbf{D} \in \mathbb{R}^{m \times N}$ is

$$\mathbf{D} = \left[\mathbf{y}_1^s - \mathbf{H} \cdot \mathbf{x}_1^b, \, \mathbf{y}_2^s - \mathbf{H} \cdot \mathbf{x}_2^b, \, \ldots, \, \mathbf{y}_N^s - \mathbf{H} \cdot \mathbf{x}_N^b \right] \tag{54}$$

and the data $\mathbf{y}_i^s$ for $1 \leq i \leq N$ is drawn from the distribution (31). We have

$$
\begin{aligned}
\mathbf{X}^a &= \mathbf{X}^b + \left(\varphi \cdot \mathbf{I}_{n \times n} + \delta \cdot \mathbf{S} \cdot \mathbf{S}^T \right) \cdot \mathbf{H}^T \\
&\quad \cdot \left(\mathbf{R} + \mathbf{H} \cdot \left(\varphi \cdot \mathbf{I}_{n \times n} + \delta \cdot \mathbf{S} \cdot \mathbf{S}^T \right) \cdot \mathbf{H}^T \right)^{-1} \cdot \mathbf{D} \\
\mathbf{X}^a &= \mathbf{X}^b + \mathbf{E} \cdot \mathbf{\Pi} \cdot \mathbf{Z}_{\widehat{\mathbf{B}}} + \varphi \cdot \mathbf{H}^T \cdot \mathbf{Z}_{\widehat{\mathbf{B}}},
\end{aligned} \tag{55}
$$

where $\mathbf{E} = \sqrt{\delta} \cdot \mathbf{S} \in \mathbb{R}^{n \times N}$, $\mathbf{\Pi} = \mathbf{H} \cdot \mathbf{E} \in \mathbb{R}^{m \times N}$, and $\mathbf{Z}_{\widehat{\mathbf{B}}} \in \mathbb{R}^{m \times N}$ is given by the solution of the linear system

$$
\begin{aligned}
\left(\mathbf{\Gamma} + \mathbf{\Pi} \cdot \mathbf{\Pi}^T \right) \cdot \mathbf{Z_B} &= \mathbf{D}, \tag{56} \\
\mathbf{\Gamma} &= \mathbf{R} + \varphi \cdot \mathbf{H} \cdot \mathbf{H}^T \in \mathbb{R}^{m \times m}.
\end{aligned}
$$

When $\mathbf{H}$ possesses a simple structure (e.g., indexes to observed components from vector states) the matrix $\mathbf{\Gamma}$ also has a simple structure (since in practice $\mathbf{R}$ is block diagonal). By letting $\mathbf{U}^{(0)} = \mathbf{\Gamma}^{-1} \cdot \mathbf{\Pi} \in \mathbb{R}^{m \times N}$ and $\mathbf{Z_B}^{(0)} = \mathbf{\Gamma}^{-1} \cdot \mathbf{D}$, the linear system (56) can be efficiently solved via the ISMF with no more than $\mathcal{O}\left(N^2 \cdot m \right)$ long computations. When $\mathbf{\Gamma}$ has no special structure its inverse can be calculated off-line.

In order to obtain a better representation of the background error statistics (uncertainty) about the background state (9) additional samples can be taken from the distribution

$$\widetilde{\mathbf{x}}_i^b \sim \mathcal{N}\left(\overline{\mathbf{x}}^b, \, \widehat{\mathbf{B}} \right) \in \mathbb{R}^{n \times 1}, \quad 1 \leq i \leq K. \tag{57}$$

This yields to 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 artificially built by taking samples from the distribution (57) and do not require additional model runs.

The artificial 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 (5). Figure 1 exemplifies the effect of additional members using two-dimensional projections of ensemble member states from the Lorenz 96 model. Figure 1a shows the spread of the real

16

ensemble members (for the background uncertainty around $\overline{\mathbf{x}}^b$). Figures 1b shows the distribution when artificial members are added to the background ensemble, resulting in a better representation of the background error and therefore a decrease in the sampling error.

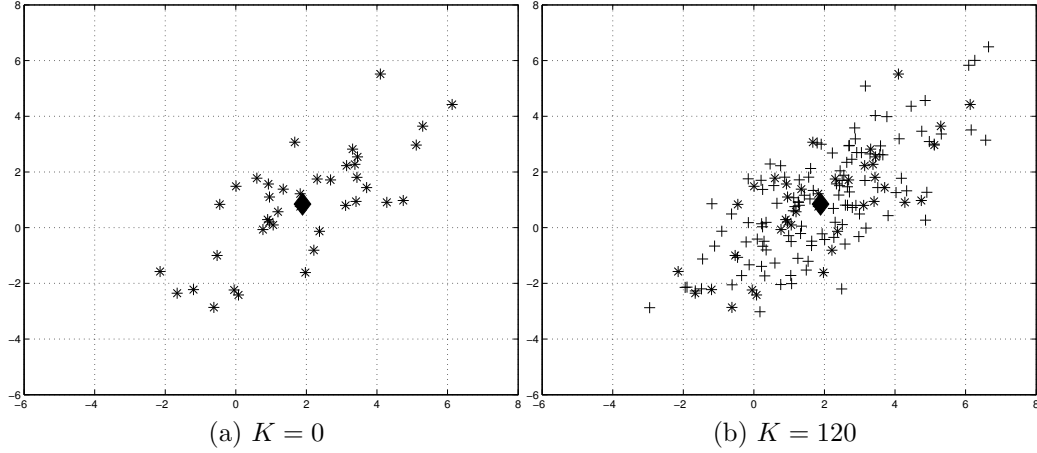
(a) $K = 0$ (b) $K = 120$

Figure 1: Error distribution for different values of K using two-dimensional projections of ensemble members from the Lorenz 96 model. The number of real members is $N = 40$. In the plots $*$ represents real ensemble members, $+$ artificial members, and $\diamond$ is the ensemble mean.

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

$$\overline{\mathbf{x}}^a \in \mathbf{span}\left\{\mathbf{x}_1^b,\, \mathbf{x}_2^b,\, \ldots,\, \mathbf{x}_N^b,\, \widetilde{\mathbf{x}}_1^b,\, \widetilde{\mathbf{x}}_2^b,\, \ldots,\, \widetilde{\mathbf{x}}_K^b\right\}. \tag{58}$$

The *extended ensemble* reads:

$$\widetilde{\mathbf{X}}^b = \left[\mathbf{x}_1^b,\, \mathbf{x}_2^b,\, \ldots,\, \mathbf{x}_N^b,\, \widetilde{\mathbf{x}}_1^b,\, \widetilde{\mathbf{x}}_2^b,\, \ldots,\, \widetilde{\mathbf{x}}_K^b\right] \in \mathbb{R}^{n \times N_k}. \tag{59}$$

where $N_k = N + K$. Similar to (55), the background ensemble (59) is updated using

$$\mathbf{X}^a = \mathbf{X}^b + \widetilde{\mathbf{E}} \cdot \widetilde{\mathbf{\Pi}} \cdot \widetilde{\mathbf{Z}}_{\widehat{\mathbf{B}}} + \varphi \cdot \mathbf{H}^T \cdot \widetilde{\mathbf{Z}}_{\mathbf{B}} \in \mathbb{R}^{n \times N}, \tag{60}$$

17

where

$$
\begin{align}
\widetilde{\mathbf{E}} &= \sqrt{\delta} \cdot \widetilde{\mathbf{S}} \in \mathbb{R}^{n \times N_k} , \tag{61a} \\
\widetilde{\mathbf{\Pi}} &= \mathbf{H} \cdot \widetilde{\mathbf{E}} \in \mathbb{R}^{m \times N_k} , \tag{61b} \\
\widetilde{\mathbf{Z}}_{\widehat{\mathbf{B}}} &= \left(\mathbf{\Gamma} + \widetilde{\mathbf{\Pi}} \cdot \widetilde{\mathbf{\Pi}}^T \right)^{-1} \cdot \mathbf{D} \in \mathbb{R}^{m \times N} , \tag{61c} \\
\widetilde{\mathbf{S}} &= \widetilde{\mathbf{X}} - \overline{\mathbf{x}}^b \otimes \mathbf{1}_{N_k}^T \in \mathbb{R}^{n \times N_k} . \tag{61d}
\end{align}
$$

The solution of (61c) can be efficiently obtained via the ISMF. Even if the artificial members (57) are used in the covariance approximation for the analysis step, according to (60) only the states of real members are adjusted. After the assimilation step the artificial 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 artificial 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 artificial members such as to refresh the ensemble directions, e.g., in order to prevent filter divergence.

To summarize the above discussion, the implementation of the ensemble Kalman filter based on the RBWL estimator with weighted covariance matrix in the observation space (EnKF-FS) consists of the following steps:

1. Estimate the background error covariance matrix (52) based on the samples $\left\{ \mathbf{x}^{b[i]} \right\}_{i=1}^N$, for $1 \leq i \leq N$.

2. Compute the innovation matrix $\mathbf{D}$ according to (54).

3. Draw K artificial members from the distribution (57).

4. Compute the set of matrices (61).

5. Perform the assimilation (60).

6. Propagate the ensemble members

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

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

18

Another efficient implementation of the filter based on the RBLW estimator can be obtained via the 3D-Var cost function in the ensemble space (36). This involves projecting the weighted covariance matrix (21) onto the ensemble space (58) rather than onto the observation space.

After taking the samples (57) and building the ensemble (59) a new set of basis vectors can be built as follows

$$\widetilde{\mathbf{U}} = \left[\mathbf{x}_1^b - \overline{\mathbf{x}}^b, \ldots, \mathbf{x}_N^b - \overline{\mathbf{x}}^b, \widetilde{\mathbf{x}}_1^b - \overline{\mathbf{x}}^b, \ldots, \widetilde{\mathbf{x}}_K^b - \overline{\mathbf{x}}^b\right] \in \mathbb{R}^{n \times N_k}, \qquad (62)$$

where $\overline{\mathbf{x}}^b$ is given in (9). By replacing (62) and the RBLW estimator of the background error covariance matrix (52) in the 3D-Var cost function (36) we obtain

$$\mathcal{J}_{\text{ens}}(\boldsymbol{\lambda}) = \frac{1}{2} \cdot \left\|\widetilde{\mathbf{U}} \cdot \boldsymbol{\lambda}\right\|_{\widehat{\mathbf{B}}^{-1}}^2 + \frac{1}{2} \cdot \left\|\mathbf{D} - \widetilde{\mathbf{Q}} \cdot \boldsymbol{\lambda}\right\|_{\mathbf{R}^{-1}}^2 \qquad (63)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{N_k \times N}$ is the matrix of weights whose i-th column represent the coordinates of the i-th ensemble member in the space (58), for $1 \leq i \leq N_k$, and $\widetilde{\mathbf{Q}} = \mathbf{H} \cdot \widetilde{\mathbf{U}} \in \mathbb{R}^{m \times N_k}$. The resulting 3D-Var optimization problem is

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \mathcal{J}_{\text{ens}}(\boldsymbol{\lambda}), \qquad (64)$$

and has the solution

$$\boldsymbol{\lambda}^* = \left[\widetilde{\mathbf{U}}^T \cdot \mathbf{Z}_{\widehat{\mathbf{B}}\mathbf{U}} + \widetilde{\mathbf{Q}}^T \cdot \mathbf{R}^{-1} \cdot \widetilde{\mathbf{Q}}\right]^{-1} \cdot \widetilde{\mathbf{Q}}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{D} \in \mathbb{R}^{N_k \times N}, \qquad (65)$$

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

$$\mathbf{X}^a = \mathbf{X}^b + \widetilde{\mathbf{U}} \cdot \boldsymbol{\lambda}^* \in \mathbb{R}^{n \times N}. \qquad (66)$$

To summarize, the implementation of the ensemble Kalman filter based on covariance estimation with weighted covariance matrix in the ensemble space (EnKF-FS) consists of the following steps:

1. Estimate the background error covariance matrix (52) based on the samples $\left\{\mathbf{x}^{b[i]}\right\}_{i=1}^N$, for $1 \leq i \leq N$.

2. Compute the innovation matrix $\mathbf{D}$ according to (54).

3. Draw K artificial members according to (57).

19

4. Compute the matrix of optimal weights (65).

5. Perform the assimilation (66).

6. Propagate the ensemble members

$$\mathbf{x}^b_{\text{next}} = \mathcal{M}_{t_{\text{current}} \to t_{\text{next}}} \left(\mathbf{x}^a_{\text{current}} \right),$$

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

3.3 Sampling in high-dimensions based on the RBLW estimator

Both implementations discussed in Section 3.2 use samples from the distribution (57). Such samples can be generated as follows:

$$\widetilde{\mathbf{x}}^b_i = \overline{\mathbf{x}}^b + \widehat{\mathbf{B}}^{1/2} \cdot \boldsymbol{\xi}_i = \overline{\mathbf{x}}^b + \left(\varphi \cdot \mathbf{I}_{n \times n} + \delta \cdot \mathbf{S} \cdot \mathbf{S}^T \right)^{1/2} \cdot \boldsymbol{\xi}_i \tag{67}$$

for $1 \leq i \leq K$, where $\boldsymbol{\xi}_i \sim \mathcal{N}\left(\mathbf{0}_n, \mathbf{I}_{n \times n} \right)$. 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 (67) impractical.

We need an equivalent strategy to obtain the samples (57) 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

$$\begin{aligned} \boldsymbol{\xi}^1_i &\sim \mathcal{N}\left(\mathbf{0}_n, \mathbf{I}_{n \times n} \right) \in \mathbb{R}^{n \times 1}, \\ \boldsymbol{\xi}^2_i &\sim \mathcal{N}\left(\mathbf{0}_N, \mathbf{I}_{N \times N} \right) \in \mathbb{R}^{N \times 1}, \end{aligned}$$

and let

$$\begin{aligned} \text{Cov}\left(\boldsymbol{\xi}^1_i, \boldsymbol{\xi}^2_i \right) &= \boldsymbol{\xi}^1_i \otimes \boldsymbol{\xi}^{2T}_i = \mathbf{0}_{n \times N}, \\ \text{Cov}\left(\boldsymbol{\xi}_2, \boldsymbol{\xi}_1 \right) &= \boldsymbol{\xi}^2_i \otimes \boldsymbol{\xi}^{1T}_i = \mathbf{0}_{N \times n}. \end{aligned}$$

We make the following substitution in (67)

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

20

This does not change the statistics since

$$
\mathbb{E}\left[\left(\sqrt{\varphi}\cdot\mathbf{I}_{n\times n}\cdot\boldsymbol{\xi}_i^1+\sqrt{\delta}\cdot\mathbf{S}\cdot\boldsymbol{\xi}_i^2\right)\cdot\left(\sqrt{\varphi}\cdot\mathbf{I}_{n\times n}\cdot\boldsymbol{\xi}_i^1+\sqrt{\delta}\cdot\mathbf{S}\cdot\boldsymbol{\xi}_i^2\right)^T\right]
$$

$$
=\varphi\cdot\underbrace{\boldsymbol{\xi}_i^1\otimes\boldsymbol{\xi}_i^{1T}}_{\mathrm{Cov}\left(\boldsymbol{\xi}_i^1,\boldsymbol{\xi}_i^1\right)=\mathbf{I}_{n\times n}}+\sqrt{\varphi\cdot\delta}\cdot\underbrace{\boldsymbol{\xi}_i^1\otimes\boldsymbol{\xi}_i^{2T}}_{\mathrm{Cov}\left(\boldsymbol{\xi}_i^1,\boldsymbol{\xi}_i^2\right)=\mathbf{0}_{n\times N}}+\sqrt{\varphi\cdot\delta}\cdot\underbrace{\boldsymbol{\xi}_i^2\otimes\boldsymbol{\xi}_i^{1T}}_{\mathrm{Cov}\left(\boldsymbol{\xi}_i^2,\boldsymbol{\xi}_i^1\right)=\mathbf{0}_{N\times n}}
$$

$$
+\delta\cdot\mathbf{S}\cdot\underbrace{\boldsymbol{\xi}_i^2\otimes\boldsymbol{\xi}_i^{2T}}_{\mathrm{Cov}\left(\boldsymbol{\xi}_i^2,\boldsymbol{\xi}_i^2\right)=\mathbf{I}_{N\times N}}\cdot\mathbf{S}^T=\varphi\cdot\mathbf{I}_{n\times n}+\delta\cdot\mathbf{S}\cdot\mathbf{S}^T
$$

$$
=\widehat{\mathbf{B}}.
$$

The artificial ensemble members are obtained as follows:

$$
\widetilde{\mathbf{x}}_i^b=\overline{\mathbf{x}}^b+\sqrt{\varphi}\cdot\mathbf{I}_{n\times n}\cdot\boldsymbol{\xi}_i^1+\sqrt{\delta}\cdot\mathbf{S}\cdot\boldsymbol{\xi}_i^2,\quad i=1,\ldots,K. \tag{68}
$$

The components of the random variables $\boldsymbol{\xi}_i^1$ and $\boldsymbol{\xi}_i^2$ 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 $\boldsymbol{\xi}_i^1$ and $\boldsymbol{\xi}_i^2$ 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 correct sampling from the distribution (57).

3.4 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 (60) to the form (6):

$$
\mathbf{X}^a=\mathbf{X}^b+\widehat{\mathbf{B}}^{1/2}\cdot\boldsymbol{\alpha}_{\widehat{\mathbf{B}}}\in\mathbb{R}^{n\times N}, \tag{69}
$$

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

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

It is readily apparent from equations (69) and (66) 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 (46) is searched for in the full space. 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 (69) reveal that the weighted covariance matrix of the EnKF-FS implementation

$$\mathbf{W} = \widehat{\mathbf{B}}^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H} \in \mathbb{R}^{n \times n} , \tag{70}$$

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

$$\mathbf{W}_{\mathrm{ens}} = \widetilde{\mathbf{U}}^T \cdot \mathbf{W} \cdot \widetilde{\mathbf{U}} \in \mathbb{R}^{N_k \times N_k} . \tag{71}$$

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

4   Experimental Results

This section tests the new EnKF implementations on a data assimilation problem using the quasi-geostrophic model presented in [38]. A comparison is done in two steps: first, the proposed implementations are compared against the well-known EnKF implementations presented in section 2, and next the quality of the results for the EnKF-FS and EnKF-RS based on different values of N and K are assessed.

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 behavior of the ocean. It is defined by the following partial differential equation:

$$\omega_t + r \cdot J\left(\omega,\, \psi\right) + \beta \cdot \psi_x - v \cdot \nabla^4 \psi = -\mu \cdot \nabla^2 \psi + \tau \cdot \sin\left(\frac{2 \cdot \pi \cdot y}{L_y}\right) \quad (72)$$

in $\Omega \in [0,\, L_x] \times [0,\, L_y]$, where x and y represent the horizontal and vertical space components, ω is the vorticity, ψ is the stream function, $J\left(\psi, \omega\right)$ is the Jacobian of two fields

$$J\left(\psi, \omega\right) = \psi_x \cdot \omega_y - \psi_y \cdot \omega_x\,, \quad (73)$$

and ∇^2 is the Laplacian operator. The coefficients β, v, μ and τ are associated with the horizontal vorticity, the horizontal friction, the biharmonic horizontal friction, and the horizontal wind stress at the surface of the ocean, respectively. Moreover, the vorticity is related to the stream function by the elliptical equation:

$$\nabla^2 \psi = \omega\,. \quad (74)$$

The spatial domain for our experiments is $\Omega = [0,\, 1] \times [0,\, 1]$. The interior is covered by computational grids of different resolutions, and we denote by D_1 and D_2 the number of horizontal and vertical grid points, respectively. The different model resolutions are presented in the table 1.

Instance	D_1	D_2	$n = D_1 \cdot D_2$
$QG_{33 \times 33}$	31	31	961
$QG_{65 \times 65}$	63	63	3,969
$QG_{129 \times 129}$	127	127	16,129

Table 1: Quasi-geostrophic instances for the computational tests in terms of the number of horizontal D_1 and vertical D_2 grid points. D_1 and D_2 do not consider boundary points.

The numerical data assimilation experiments are characterized by the following settings:

- Initial vorticities have the form

$$\omega_0 = \sin(4\,x_i\,y_j) \cdot \cos(2\,x_i\,y_j) + \sin(2\,x_i\,y_j) + \cos(4\,x_i\,y_j)\,,$$

  for $1 \leq i \leq D_1$ and $1 \leq j \leq D_2$.

- The initial background error covariance is

$$\mathbf{B} = \left[\sigma^{\mathbf{B}}\right]^2 \cdot \mathbf{I}_{n \times n},$$

  where the standard deviation $\sigma^{\mathbf{B}}$ is chosen to be 0.05 or 0.15 (times the true vorticity) in different experiments .

- The observational errors are uncorrelated with variances $[0.01]^2$.

- The number of observed components from the vector state is given by

$$m = p \cdot n\,, \tag{75}$$

  where p is the percentage of observed components. We consider two values for p, 0.7 and 0.9, corresponding to a sparser and a denser network, respectively.

Other aspects of the numerical simulation are described below:

- The EnKF methods are implemented in C++ while the forward model (QG) [37] is implemented making use of FORTRAN.

- The partial derivatives are discretized by central finite differences.

- The matrix and vector computations are efficiently carried out using the BLAS library [5].

- Matrix decompositions as well as eigenvalue computations are performed using the LAPACK library [1].

- The Arakawa method [26] is utilized in order to compute the Jacobian (73).

- The time discretization of the model (72) uses of a fourth order Runge-Kutta method. The time step size 1.27 (units) which represent one hour in the ocean. The integration is performed for 1000 hours.

- The NLOPT library [28] is utilized to numerically solve the optimization problems (39) and (43).

- The GSL-GNU scientific library [23] is utilized to generate the synthetic background and data errors.

4.1   Comparison with current EnKF implementations

We compare the EnKF-RS and EnKF-FS methods against current well-known EnKF implementations. The root mean square errors (RMSE) and CPU times for different methods making use of the $QG_{33\times33}$, $QG_{65\times65}$ and $QG_{129\times129}$ model instances are reported inTables 2, 3, and 4, respectively. For a given number of snapshots $\{\mathbf{x}_i^a\}_{i=1}^M$ and a reference trajectory $\{\mathbf{x}_i^{\text{true}}\}_{i=1}^M$, where M is the number of assimilation times, the RMSE is defined as follows

$$\text{RMSE} = \sqrt{\frac{1}{M} \cdot \sum_{i=1}^M \|\mathbf{x}_i^a - \mathbf{x}_i^{\text{true}}\|^2}.$$

We vary the size of the ensemble N, the percentage of observed components p, and the initial background error $\sigma^{\mathbf{B}}$. As expected the traditional ensemble implementations EnKF, EnSRF, and EnTKF provide accurate analyses for different model resolutions and number of observed components. Moreover it can be seen in figures 2 and 3 that the RMSE decreases as the simulation progresses. The traditional EnKF implementations provide the lowest elapsed time among the compared methods (i.e., EnKF and EnSRF implementations) which explains why they are attractive for use in real applications. The RMSE values for the EnSRF and EnTKF implementations are identical since both filters are deterministic and EnTKF is just an efficient implementation of the EnSRF. The inflation-free methods such as the EnKF-FN and EnK-DU implementations provide slightly better accurate results than other methods. For example, for the largest instance $QG_{129\times129}$, table 4 shows the EnKF-DU to perform better than current implementations. In most of the cases, strong duality holds: $\mathcal{J}_{ens}^{\text{FN}}(\boldsymbol{\alpha}^*) = \mathcal{D}_{ens}^{\text{DU}}(\zeta^*)$ and the slight differences between the optimal cost function values (39) and (43) are consistent with the numerical approximation errors in the solution of the optimization problems (40) and (45), respectively.

Since the analysis state in the EnKF-DU formulation is obtained via the solution of the one-dimensional optimization problem (45), we expect this

25

method to be faster than the EnKF-FN implementation where the analysis requires the solution of the N-dimensional optimization problem (39). This fact is also pointed out by Boquet in [6, Section 2.2], where the cost of computing the inverse (44) is assumed to be negligible in the dual formulation. However, this statement seems to be true for small model resolutions (i.e., Bocquet makes use of the Lorenz 96 model with 40 variables) and it holds for the smallest QG instance $QG_{33\times33}$. Nevertheless, for the $QG_{65\times65}$ case, the difference between the CPU times for the primal and dual implementations is almost negligible and even more, for the largest instance $QG_{129\times129}$, the EnKF-FN performs better than its dual approach for $\sigma^{\mathbf{B}} = 0.15$. Although the cost function (43) depends only on ζ, every step in the optimization process requires the solution of the linear system (44) whose computational cost is not negligible in practice. Furthermore, when the initial background error is large the EnKF-DU computes many times the inverse of (44) and therefore its performance decreases considerably.

Even if the traditional implementations perform very well in terms of RMSE and elapsed time, the most accurate results are obtained by the proposed new EnKF implementations. The results presented in figures 2a and 3a show that the EnKF-RS method performs much better than traditional and inflation-free methods for the small instance $QG_{33\times33}$, and are slightly better in the larger instances. The most accurate results among all the compared methods are the ones obtained by the EnKF-FS implementation. The results reported in figures 2 and 3 show that, for all the instances and configurations, the RMSE obtained via the EnKF-FS outperforms the other methods by at least 60%. The CPU times for the proposed implementations are just slightly larger than those from the compared EnKF implementations. Since most of the computational time is spent in propagating the ensemble members, a modest increase in analysis time retains the potential of the new methods to perform well in practical applications.

			RMSE		CPU Time	
			$\sigma^{\mathbf{B}}$		$\sigma^{\mathbf{B}}$	
N	p	Method	0.05	0.15	0.05	0.15
40	0.7	EnKF	1.287	3.858	0.013	0.009
		EnSRF	1.289	3.864	0.013	0.013
		EnTKF	1.289	3.864	0.026	0.028
		EnKF-FN	1.286	3.855	0.125	0.116
		EnKF-DU	1.285	3.854	0.045	0.036
		EnKF-FS	0.659	1.974	0.024	0.036
		EnKF-RS	1.16	3.49	0.421	0.252
	0.9	EnKF	1.279	3.836	0.019	0.021
		EnSRF	1.282	3.841	0.01	0.017
		EnTKF	1.282	3.841	0.028	0.017
		EnKF-FN	1.279	3.831	0.115	0.115
		EnKF-DU	1.276	3.828	0.072	0.087
		EnKF-FS	0.371	1.116	0.048	0.028
		EnKF-RS	1.07	3.209	0.395	0.222
80	0.7	EnKF	1.268	3.803	0.065	0.061
		EnSRF	1.275	3.82	0.034	0.052
		EnTKF	1.275	3.82	0.143	0.076
		EnKF-FN	1.264	3.79	0.608	0.681
		EnKF-DU	1.263	3.79	0.162	0.181
		EnKF-FS	0.646	1.927	0.119	0.105
		EnKF-RS	1.069	3.144	1.303	1.285
	0.9	EnKF	1.252	3.756	0.074	0.077
		EnSRF	1.26	3.773	0.058	0.051
		EnTKF	1.26	3.773	0.168	0.12
		EnKF-FN	1.249	3.737	0.894	0.766
		EnKF-DU	1.246	3.737	0.155	0.181
		EnKF-FS	0.358	1.074	0.192	0.14
		EnKF-RS	0.879	2.588	1.269	1.26

Table 2: RMSE and CPU-time (TIME) for the EnKF, EnSRF, EnTKF, EnKF-FN, EnKF-DU, EnKF-RS and EnKF-FS implementations applied to the $QG_{33\times33}$ instance ($n = 961$).

			RMSE		CPU Time	
			$\sigma^{\mathbf{B}}$		$\sigma^{\mathbf{B}}$	
N	p	Method	0.05	0.15	0.05	0.15
40	0.7	EnKF	1.67	5.011	0.072	0.076
		EnSRF	1.675	5.025	0.059	0.047
		EnTKF	1.675	5.025	0.056	0.098
		EnKF-FN	1.671	4.987	0.185	0.261
		EnKF-DU	1.662	4.987	0.17	0.245
		EnKF-FS	0.893	2.668	0.108	0.205
		EnKF-RS	1.61	4.818	1.869	1.801
	0.9	EnKF	1.667	5.001	0.095	0.061
		EnSRF	1.671	5.015	0.06	0.035
		EnTKF	1.671	5.015	0.067	0.117
		EnKF-FN	1.667	4.975	0.262	0.269
		EnKF-DU	1.658	4.975	0.273	0.244
		EnKF-FS	0.52	1.533	0.209	0.139
		EnKF-RS	1.589	4.761	1.975	2.011
80	0.7	EnKF	1.653	4.958	0.189	0.189
		EnSRF	1.661	4.982	0.174	0.183
		EnTKF	1.661	4.982	0.265	0.377
		EnKF-FN	1.644	4.915	0.934	0.919
		EnKF-DU	1.638	4.916	0.567	0.787
		EnKF-FS	0.866	2.592	0.634	0.495
		EnKF-RS	1.546	4.612	7.627	7.542
	0.9	EnKF	1.65	4.949	0.271	0.302
		EnSRF	1.657	4.97	0.227	0.181
		EnTKF	1.657	4.97	0.428	0.429
		EnKF-FN	1.639	4.888	1.099	1.01
		EnKF-DU	1.63	4.888	0.85	0.936
		EnKF-FS	0.494	1.486	0.607	0.502
		EnKF-RS	1.507	4.509	7.845	8.012

Table 3: RMSE and CPU-time (TIME) for the EnKF, EnSRF, EnTKF, EnKF-FN, EnKF-DU, EnKF-RS and EnKF-FS implementations applied to the $QG_{65\times65}$ instance ($n = 3969$).

N	p	Method	RMSE $\sigma^{\mathbf{B}}$ 0.05	0.15	CPU Time $\sigma^{\mathbf{B}}$ 0.05	0.15
40	0.7	EnKF	1.708	5.125	0.307	0.186
		EnSRF	1.712	5.136	0.174	0.153
		EnTKF	1.712	5.136	0.237	0.223
		EnKF-FN	1.707	5.1	0.793	0.589
		EnKF-DU	1.7	5.1	0.788	0.746
		EnKF-FS	0.963	2.847	0.572	0.611
		EnKF-RS	1.682	5.042	5.607	5.832
	0.9	EnKF	1.709	5.127	0.263	0.277
		EnSRF	1.712	5.134	0.166	0.192
		EnTKF	1.712	5.134	0.294	0.296
		EnKF-FN	1.704	5.089	0.888	0.849
		EnKF-DU	1.696	5.089	1.047	1.041
		EnKF-FS	0.582	1.661	0.627	0.745
		EnKF-RS	1.676	5.02	6.368	6.292
80	0.7	EnKF	1.702	5.104	0.721	0.693
		EnSRF	1.709	5.125	0.554	0.576
		EnTKF	1.709	5.125	0.838	0.734
		EnKF-FN	1.69	5.063	3.085	2.596
		EnKF-DU	1.687	5.063	2.826	2.931
		EnKF-FS	0.941	2.81	2.191	2.373
		EnKF-RS	1.654	4.94	26.384	26.355
	0.9	EnKF	1.7	5.099	0.834	0.991
		EnSRF	1.708	5.123	0.667	0.729
		EnTKF	1.708	5.123	0.924	0.963
		EnKF-FN	1.683	5.034	4.07	3.485
		EnKF-DU	1.678	5.034	3.565	3.501
		EnKF-FS	0.558	1.622	2.745	2.442
		EnKF-RS	1.634	4.908	27.672	26.208

Table 4: RMSE and CPU-time for for the EnKF, EnSRF, EnTKF, EnKF-FN, EnKF-DU, EnKF-RS and EnKF-FS implementations applied to the $QG_{129\times129}$ instance ($n = 16129$).
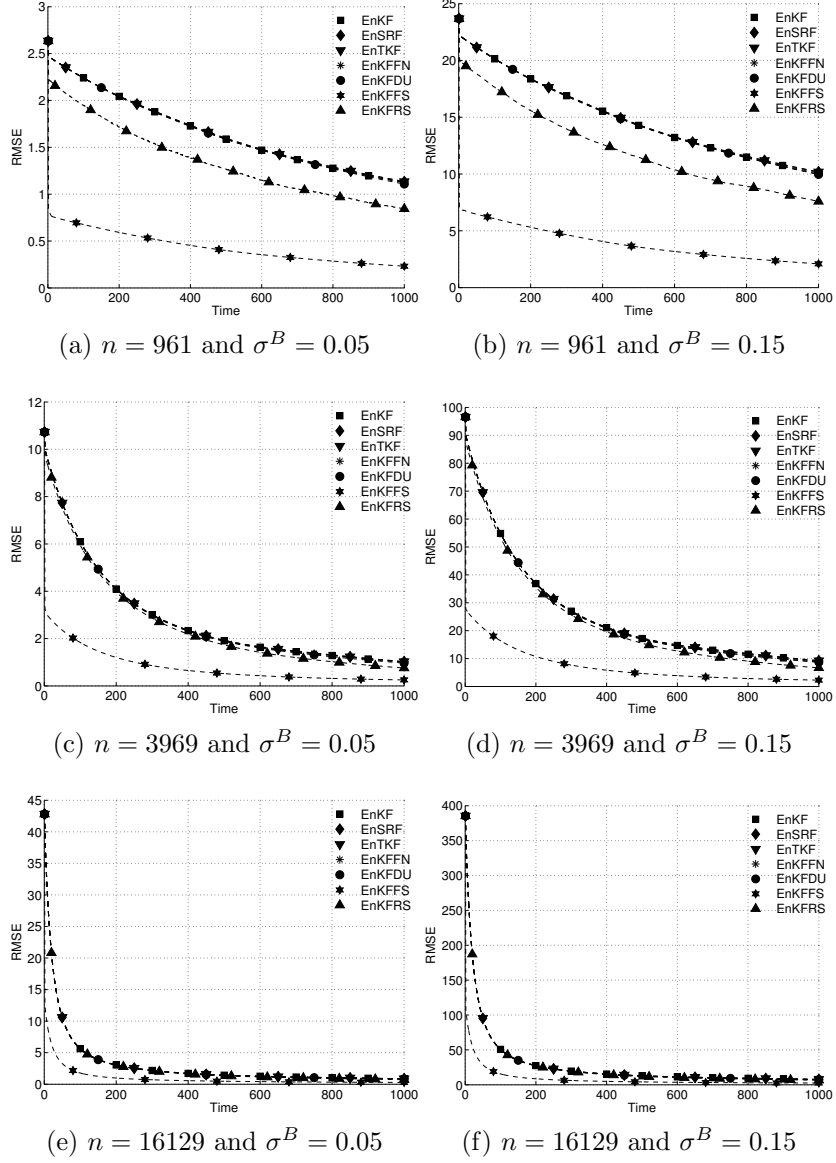
(a) $n = 961$ and $\sigma^B = 0.05$

(b) $n = 961$ and $\sigma^B = 0.15$

(c) $n = 3969$ and $\sigma^B = 0.05$

(d) $n = 3969$ and $\sigma^B = 0.15$

(e) $n = 16129$ and $\sigma^B = 0.05$

(f) $n = 16129$ and $\sigma^B = 0.15$

Figure 2: Plots of RMSE values for the EnKF, EnSRF, EnTKF, EnKF-FN, EnKF-DU, EnKF-FS and EnKF-RS for $N = 40$ and $p = 0.7$.

30

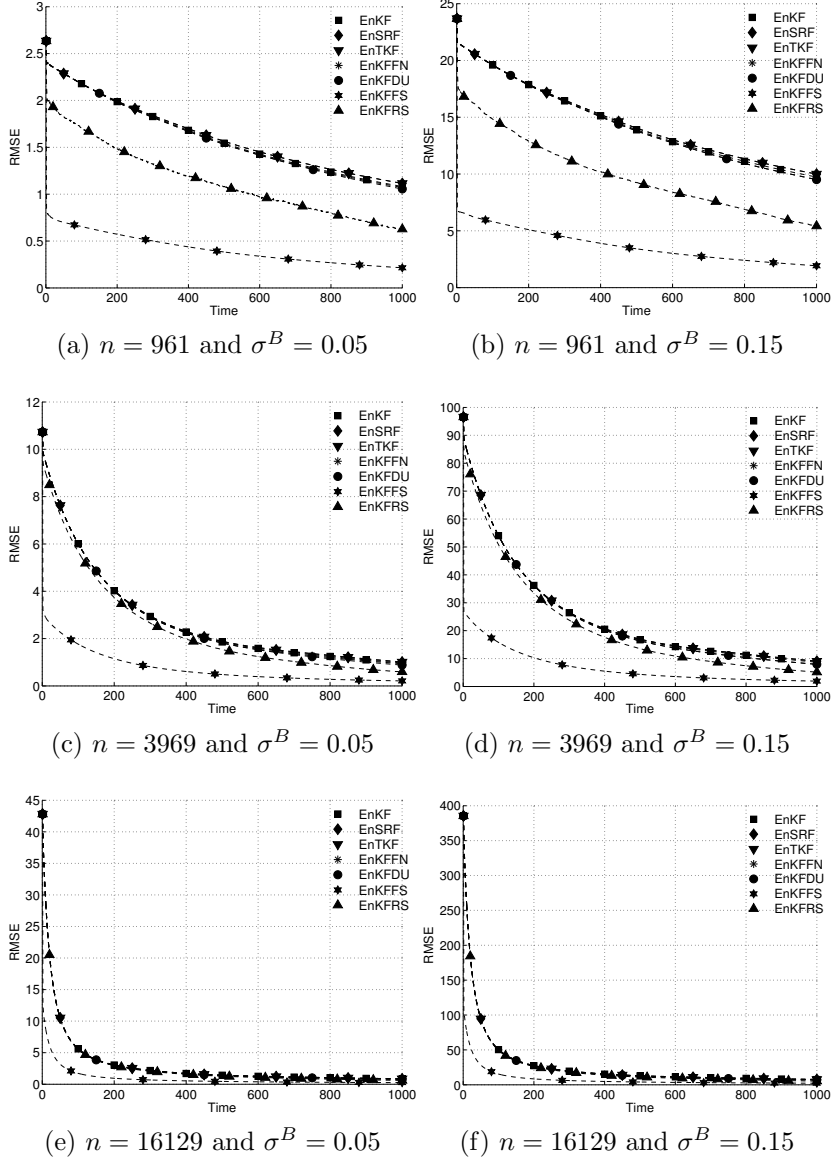(a) $n = 961$ and $\sigma^B = 0.05$

(b) $n = 961$ and $\sigma^B = 0.15$

(c) $n = 3969$ and $\sigma^B = 0.05$

(d) $n = 3969$ and $\sigma^B = 0.15$

(e) $n = 16129$ and $\sigma^B = 0.05$

(f) $n = 16129$ and $\sigma^B = 0.15$

Figure 3: Plots of RMSE values of the EnKF, EnSRF, EnTKF, EnKF-FN, EnKF-DU, EnKF-FS and EnKF-RS for $N = 80$ and $p = 0.7$.

31

4.2 The impact of the number of ensemble members on performance

The EnKF-RS and EnKF-FS implementations depend on the K samples taken from the distribution (57). We now study how the performance of the proposed methods varies for different values of N (number of real members) and K (number of artificial members). For this we let K and N to be related by

$$K = C \cdot N ,$$

where C is a constant. Practical ensemble sizes range in $40 \leq N \leq 80$ [22]. For each ensemble size we use several values of C between 0 and 10, e.g., $N = 40$ and $C = 10$ lead to $K = 400$. When $C = 0$ no artificial members are added, but the error covariance matrix $\widehat{\mathbf{B}}$ is estimated. In the numerical experiments, the variances of the initial background error are set to $\sigma^{\mathbf{B}} = 0.15$. The analyses RMSE values and the compute times for the proposed implementations using the $QG_{33 \times 33}$ instance are reported in figures 4 and 5, respectively. The analysis times for both implementations are small. Moreover, as expected, EnKF-RS is sensitive to changes in any of the parameters N and K. The RMSE is decreased when the values of those parameters are high as shown in figures 4b and 4d. The RMSE of the EnKF-FS analysis decreases only with increasing N as can be seen in figures 4a and 4c.

An important question is how well do the proposed implementations perform with a small number of real members. Hopefully the inexpensive addition of artificial members can compensate for a small number of real ones. To this end we consider a small number of real members $10 \leq N \leq 30$ and a large number of artificial members with $10 \leq C \leq 60$. The results for the EnKF-RS are shown in figures 6a and 6c for p equal to 0.7 and 0.9, respectively. The EnKF-RS implementation improves the estimated analysis state whenever N or K are increased. Moreover, the quality of the analyses obtained with small real ensembles ($10 \leq N \leq 30$) is comparable to those obtained with large real ensemble sizes ($40 \leq N \leq 80$). This justifies the addition of inexpensive artificial members in order to increase the degrees of freedom of the ensemble. The EnKF-FS analysis improves only when the number of real members is increased. This can be seen in figures 6a and 6c. For the smallest ensemble size ($N = 10$) the results obtained by the

EnKF-FS are better than those of any other implementation, including the traditional EnKF implementations with the large ensemble size $N = 80$.

Figures 6b and 6d show that the EnKF-RS analyses for a small number of observed components (70%) and a large number of artificial members (K) are equivalent to those obtained with a large number of real members (N) and many observed components (90%). This is another indication of the positive impact obtained by increasing the number of degrees of freedom with samples from the distribution (57). This is computationally less expensive than adding real members via running the model.

(a) EnKF-FS $p = 0.7$ (b) EnKF-RS $p = 0.7$

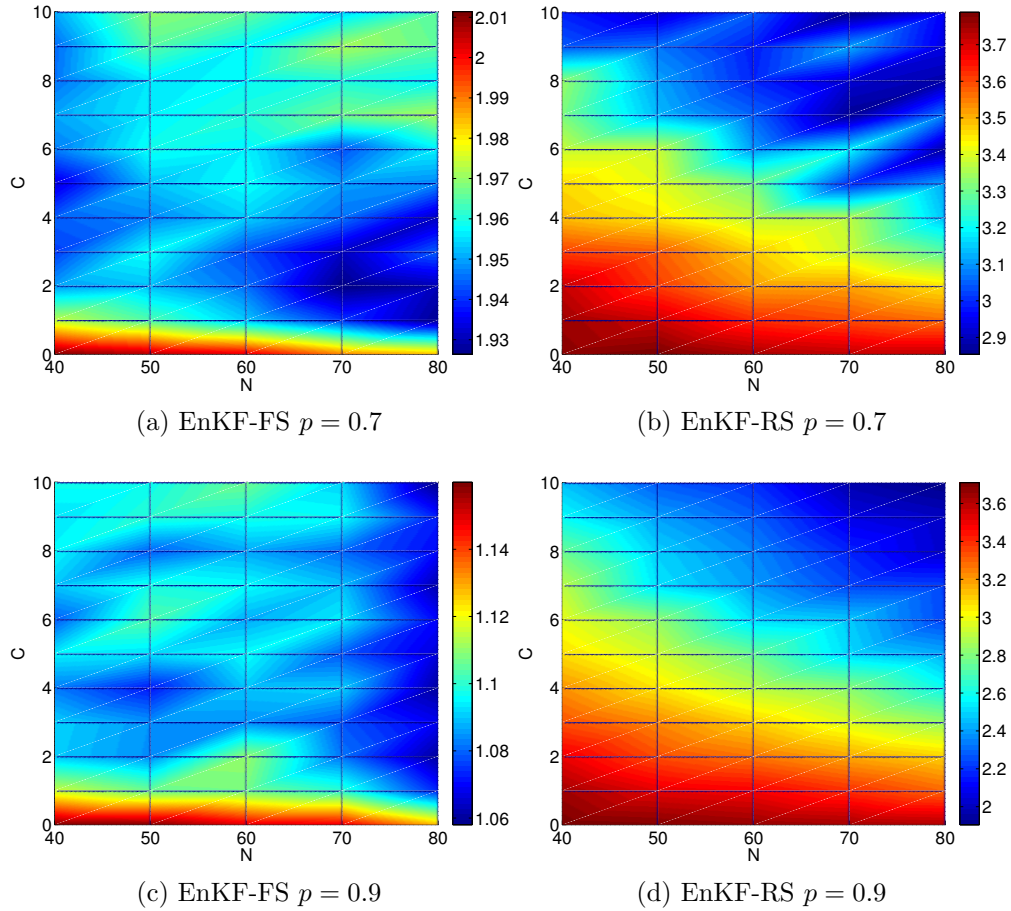(c) EnKF-FS $p = 0.9$ (d) EnKF-RS $p = 0.9$

Figure 4: RMSE of the EnKF-FS and EnKF-RS implementations for different values of $0 \leq C \leq 10$ and $40 \leq N \leq 80$ making use of the $QG_{33 \times 33}$ instance.

(a) EnKF-FS, $p = 0.7$ (b) EnKF-RS, $p = 0.7$

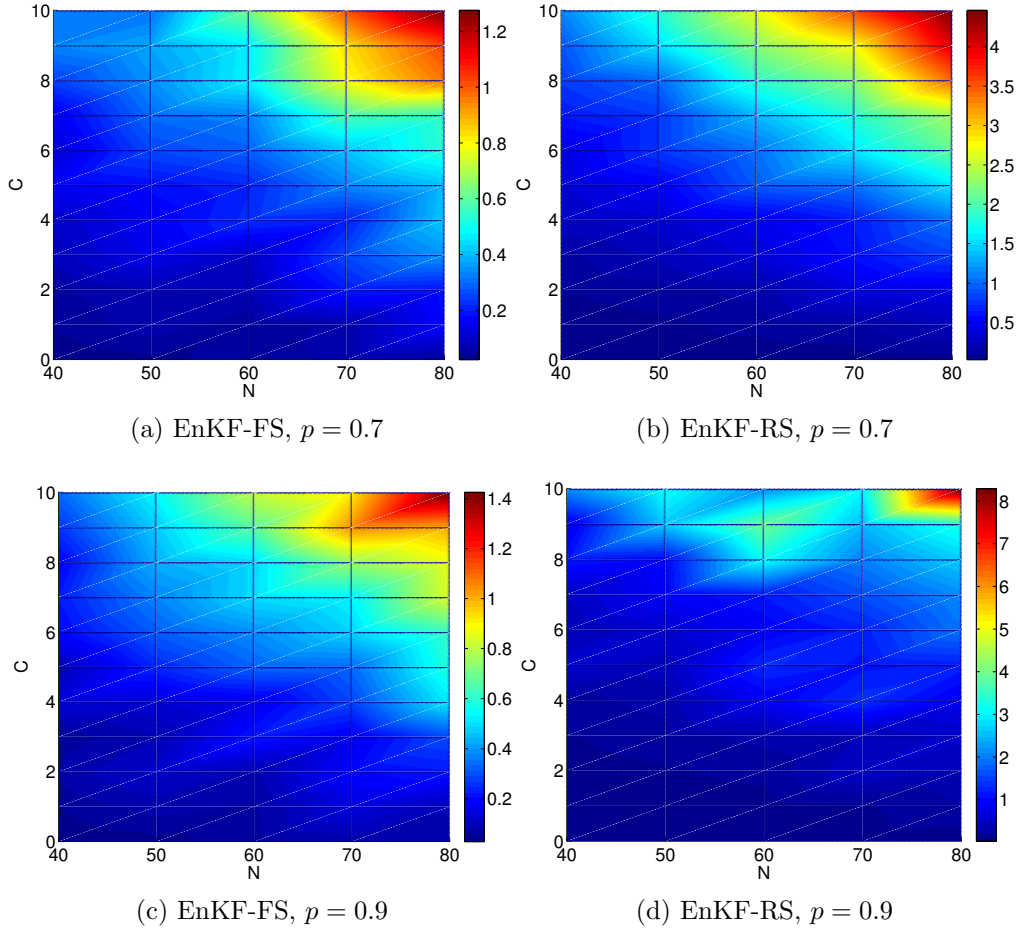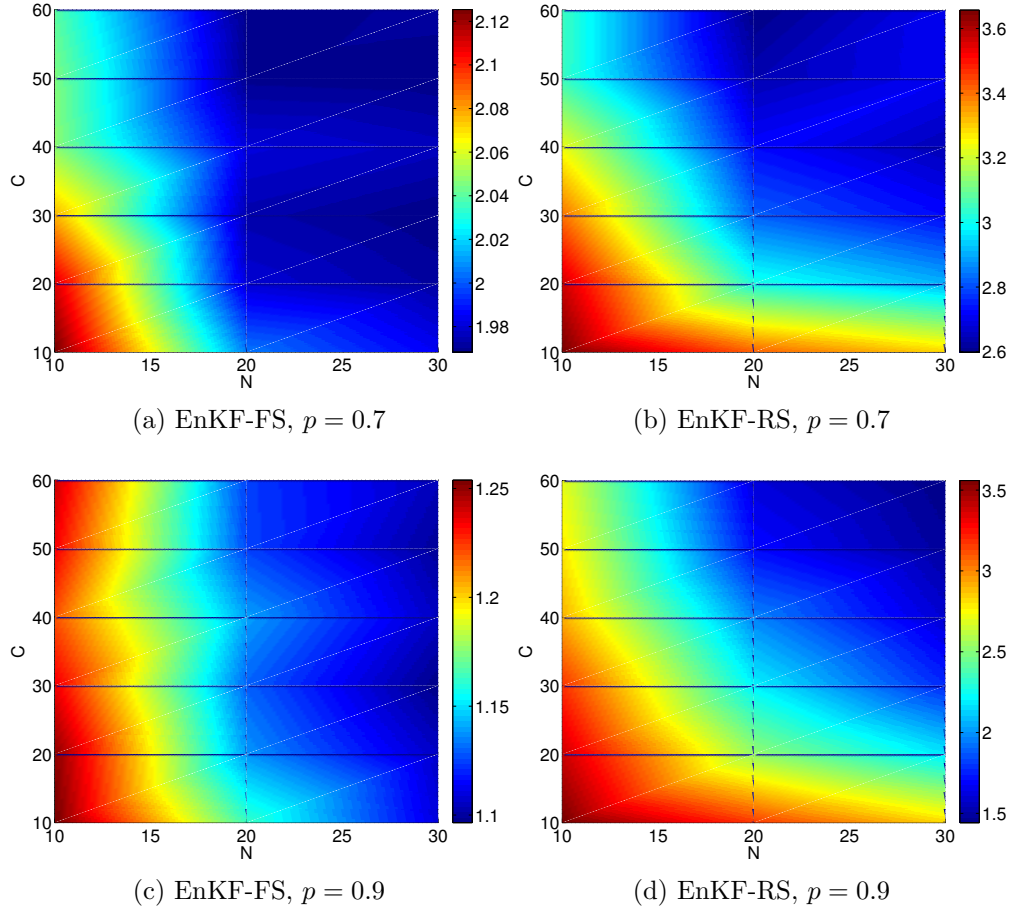(c) EnKF-FS, $p = 0.9$ (d) EnKF-RS, $p = 0.9$

Figure 5: Assimilation times of the EnKF-FS and EnKF-RS implementations for different values of $0 \leq C \leq 10$ and $40 \leq N \leq 80$ making use of the $QG_{33 \times 33}$ instance

(a) EnKF-FS, $p = 0.7$ (b) EnKF-RS, $p = 0.7$

(c) EnKF-FS, $p = 0.9$ (d) EnKF-RS, $p = 0.9$

Figure 6: RMSEs of the EnKF-FS and EnKF-RS implementations for different values of $10 \leq C \leq 60$ and $10 \leq N \leq 30$ making use of the $QG_{33 \times 33}$ instance.
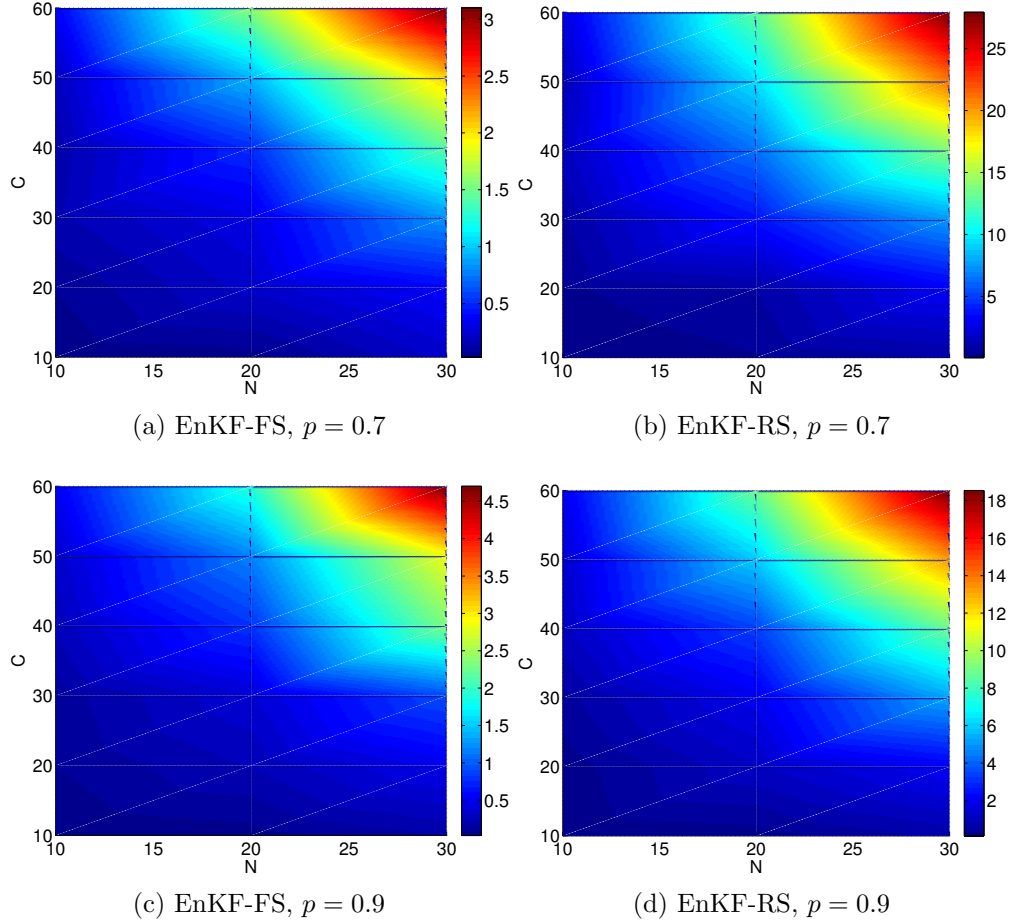
(a) EnKF-FS, $p = 0.7$ (b) EnKF-RS, $p = 0.7$

(c) EnKF-FS, $p = 0.9$ (d) EnKF-RS, $p = 0.9$

Figure 7: Assimilation times of the EnKF-FS and EnKF-RS implementations for different values of $10 \leq C \leq 60$ and $10 \leq N \leq 30$ making use of the $QG_{33 \times 33}$ instance.

5   Conclusions

This paper 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 and the background state (ensemble mean) serve as parameters of the normal error distribution associated with the ensemble members. Samples from this distribution are taken in order to increase the number of ensemble members, and therefore, to decrease the sampling error in representing the background error distribution, and to increase the number of degrees of freedom in the assimilation. The two proposed implementations differ in the space where the assimilation process is performed: 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. Numerical experiments are carried out using a quasi-geostrophic model. They show that the two new implementations perform better than current EnKF implementations such as the traditional EnKF, square root filters, and inflation-free EnKF methods. For all the scenarios and experimental settings, the EnKF-FS outperforms the other implementations by at least 60% in terms of accuracy (root mean square error). Moreover, for a small number of ensemble members (~ 10) and a moderate percentage of observed components from the vector state ($\sim 70\%$), the solutions obtained by the proposed methods are similar to those obtained by large ensemble sizes (~ 80) and large percentage of observed components ($\sim 90\%$). The computational time for analysis of the proposed implementations is reasonably low. 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.

Acknowledgements

References

[1] 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, IEEE Computer Society Press, Los Alamitos, CA, USA, 1990.

[2] Jeffrey L. Anderson. Localization and Sampling Error Correction in

Ensemble Kalman Filter Data Assimilation. *Monthly Weather Review*, 140(7):2359–2371, 2012.

[3] Angela Benedetti and Michael Fisher. Background Error Statistics for Aerosols. *Quarterly Journal of the Royal Meteorological Society*, 133(623):391–405, 2007.

[4] 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.

[5] L. S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. C. Whaley. An Updated Set of Basic Linear Algebra Subprograms (BLAS). *ACM Transactions on Mathematical Software*, 28:135–151, 2001.

[6] M. Bocquet and P. Sakov. Combining Inflation-free and Iterative Ensemble Kalman Filters for Strongly Nonlinear Systems. *Nonlinear Processes in Geophysics*, 19(3):383–399, 2012.

[7] Mark Buehner. Ensemble-derived Stationary and Flow-dependent Background-error Covariances: Evaluation in a Quasi-operational NWP Setting. *Quarterly Journal of the Royal Meteorological Society*, 131(607):1013–1043, 2005.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] Geir Evensen. The Ensemble Kalman Filter: Theoretical Formulation and Practical Implementation. *Ocean Dynamics*, 53(4):343–367, 2003.

[18] Geir Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[19] R. W. Farebrother. A Class of Shrinkage Estimators. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):pp. 47–49, 1978.

[20] 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.

[21] Reinhard Furrer and Thomas Bengtsson. Estimation of High-dimensional Prior and Posterior Covariance Matrices in Kalman Filter Variants. *Journal of Multivariate Analysis*, 98(2):227 – 255, 2007.

[22] 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.

[23] Brian Gough. *GNU Scientific Library Reference Manual - Third Edition*. Network Theory Ltd., 3rd edition, 2009.

[24] 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.

[25] 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.

[26] Dennis C Jespersen. Arakawa's Method is a Finite-element Method. *Journal of Computational Physics*, 16(4):383 – 390, 1974.

[27] 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.

[28] Steven G. Johnson. The NLopt Nonlinear-optimization Package. *10*, 10(10):10, 2012

[29] 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.

[30] Olivier Ledoit and Michael Wolf. A Well-conditioned Estimator for Large-dimensional Covariance Matrices. *Journal of Multivariate Analysis*, 88(2):365 – 411, 2004.

[31] A. C. Lorenc. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474):1177–1194, 1986.

[32] 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.

[33] Md. Nurujjaman, A. Apte, and P. Vinayachandran. Data Assimilation Using Ensemble Transform Kalman Filter (ETKF) in ROMS Model for Indian Ocean. *The European Physical Journal Special Topics*, 222(3-4):875–883, 2013.

[34] 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.

[35] Junyong Park. Shrinkage Estimator in Normal Mean Vector Estimation Based on Conditional Maximum Likelihood Estimators . *Statistics & Probability Letters*, 93(0):1–6, 2014.

[36] 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.

[37] 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.

[38] 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.

[39] Michael K. Tippett, Jeffrey L. Anderson, Craig H. Bishop, Thomas M. Hamill, and Jeffrey S. Whitaker. Square Root Filters*. *Monthly Weather Review*, 131(7):1485–1490, 2003.

[40] Jeffrey S. Whitaker and M. Hamill Thomas. Ensemble Data Assimilation without Perturbed Observations. *Monthly Weather Review*, 16(3):1913–1924, 2002.

[41] 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.

[42] 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.