Chapter 9

Efficient Matrix-Free Ensemble Kalman Filter Implementations: Accounting for Localization

Elias David Niño Ruiz, Rolando Beltrán Arrieta and Alfonso Manuel Mancilla Herrera

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.72465

Abstract

This chapter discusses efficient and practical matrix-free implementations of the ensemble Kalman filter (EnKF) in order to account for localization during the assimilation of observations. In the EnKF context, an ensemble of model realizations is utilized in order to estimate the moments of its underlying error distribution. Since ensemble members come at high computational costs (owing to current operational model resolutions) ensemble sizes are constrained by the hundreds while, typically, their error distributions range in the order of millions. This induces spurious correlations in estimates of prior error correlations when these are approximated via the ensemble covariance matrix. Localization methods are commonly utilized in order to counteract this effect. EnKF implementations in this context are based on a modified Cholesky decomposition. Different flavours of Cholesky-based filters are discussed in this chapter. Furthermore, the computational effort in all formulations is linear with regard to model resolutions. Experimental tests are performed making use of the Lorenz 96 model. The results reveal that, in terms of root-mean-square-errors, all formulations perform equivalently.

Keywords: ensemble Kalman filter, modified Cholesky decomposition, sampling methods

1. Introduction

The ensemble Kalman filter (EnKF) is a sequential Monte Carlo method for parameter and state estimation in highly nonlinear models. The popularity of the EnKF owes to its simple

formulation and relatively easy implementation. In the EnKF, an ensemble of model realizations is utilized in order to, among other things, estimate the moments of the prior error distribution (forecast distribution). During this estimation and the assimilation of observations, many challenges are present in practice. Some of them are addressed in this chapter: the first one is related to the proper estimation of background error correlations, which has a high impact on the quality of the analysis corrections while the last one is related to the proper estimation of the posterior ensemble when the observation operator is nonlinear. In all cases, background error correlations are estimated based on the Bickel and Levina estimator [1].

2. Preliminaries

We want to estimate the state of a dynamical system $\boldsymbol{x}^* \in \mathbb{R}^{n\times 1}$, which (approximately) evolves according to some (imperfect) numerical model operators $\boldsymbol{x}_{\boldsymbol{k}}^* = \boldsymbol{\mathcal{M}}_{t_{k-1}\to t_k}\left(\boldsymbol{x}_{\boldsymbol{k-1}}^*\right)$, for instance, a model which mimics the behavior of the atmosphere and/or the ocean where n is the model dimension (typically associated with the model resolution). The estimation process is based on two sources of information: a prior estimate of $\boldsymbol{x}^*$, $\boldsymbol{x}^b = \boldsymbol{x}^* + \boldsymbol{\xi}$ with $\boldsymbol{\xi} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{0}_n, \boldsymbol{B})$, and a noisy observation $\boldsymbol{y} = \boldsymbol{\mathcal{H}}(\boldsymbol{x}^*) + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{0}_m, \boldsymbol{R})$, where m is the number of observed model components, $\boldsymbol{R}^{m\times m}$ is the estimated data error covariance matrix, $\boldsymbol{\mathcal{H}} : \mathbb{R}^{n\times 1} \to \mathbb{R}^{m\times 1}$ is the (nonlinear) observation operator that maps the information from the model domain to the observation space, $\boldsymbol{0}_n$ is the nth dimensional vector whose components are all zeros. In practice, $m \ll n$ or $m < n$. The assimilation process can be performed by using sequential data assimilation methods, and one of the most widely used methods is the ensemble Kalman filter, which is detailed in the following paragraphs.

2.1. The ensemble Kalman filter

In the ensemble Kalman filter (EnKF) [2], an ensemble of model realizations

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

is utilized in order to estimate the moments of the background error distribution,

$$\boldsymbol{x}^{\boldsymbol{b}[\boldsymbol{i}]} \sim \boldsymbol{\mathcal{N}}\left(\boldsymbol{x}^{\boldsymbol{b}}, \boldsymbol{B}\right), \tag{2}$$

via the empirical moments of the ensemble (1) and therefore,

$$\boldsymbol{x}^{\boldsymbol{b}} \approx \overline{\boldsymbol{x}}^{\boldsymbol{b}} = \frac{1}{N} \cdot \sum_{\boldsymbol{i}=\boldsymbol{1}}^{N} \boldsymbol{x}^{\boldsymbol{b}[\boldsymbol{i}]} \in \mathbb{R}^{n\times 1}, \tag{3}$$

and

$$\boldsymbol{B} \approx \boldsymbol{P}^{\boldsymbol{b}} = \frac{1}{N-1} \cdot \Delta \boldsymbol{X}^{\boldsymbol{b}} \cdot \left[\Delta \boldsymbol{X}^{\boldsymbol{b}}\right]^{\boldsymbol{T}} \in \mathbb{R}^{n \times n}, \tag{4}$$

where N is the ensemble size, $\boldsymbol{x}^{\boldsymbol{b}[i]} \in \mathbb{R}^{n \times 1}$ is the ith ensemble member, for $1 \leq i \leq N$, $\boldsymbol{x}^{\boldsymbol{b}} \in \mathbb{R}^{n \times 1}$ is well known as the background state, $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ stands for the background error covariance matrix, $\overline{\boldsymbol{x}}^{\boldsymbol{b}}$ is the ensemble mean, and $\boldsymbol{P}^{\boldsymbol{b}}$ is the ensemble covariance matrix. Likewise, the matrix of member deviations $\Delta \boldsymbol{X}^{\boldsymbol{b}} \in \mathbb{R}^{n \times N}$ reads,

$$\Delta \boldsymbol{X}^{\boldsymbol{b}} = \boldsymbol{X}^{\boldsymbol{b}} - \overline{\boldsymbol{x}}^{\boldsymbol{b}} \cdot \boldsymbol{1}_{\boldsymbol{N}}^{\boldsymbol{T}} \in \mathbb{R}^{n \times N}. \tag{5}$$

A variety of EnKF implementations have been proposed in the current literature, most of them rely on basic assumptions over the moments of two probability distributions: the background (prior) and the analysis (posterior) distributions. In most of EnKF formulations, normal assumptions are done over both error distributions.

In general, depending on how the assimilation process is performed, the EnKF implementation will fall in one of two categories: stochastic filter [3] or deterministic filter [4]. In the context of stochastic filters, for instance, the assimilation of observations can be performed by using any of the next formulations,

$$\boldsymbol{X}^{\boldsymbol{a}} = \boldsymbol{A} \cdot \left[\left[\boldsymbol{P}^{\boldsymbol{b}}\right]^{-\boldsymbol{1}} \cdot \boldsymbol{X}^{\boldsymbol{b}} + \boldsymbol{H}^{\boldsymbol{T}} \cdot \boldsymbol{R}^{-\boldsymbol{1}} \cdot \boldsymbol{Y}^{\boldsymbol{s}}\right] \in R^{n \times N}, \tag{6}$$

$$\boldsymbol{X}^{\boldsymbol{a}} = \boldsymbol{X}^{\boldsymbol{b}} + \boldsymbol{A} \cdot \boldsymbol{H}^{\boldsymbol{T}} \cdot \boldsymbol{R}^{-\boldsymbol{1}} \cdot \Delta \boldsymbol{Y} \in \mathbb{R}^{n \times N}, \tag{7}$$

and

$$\boldsymbol{X}^{\boldsymbol{a}} = \boldsymbol{X}^{\boldsymbol{b}} + \boldsymbol{P}^{\boldsymbol{b}} \cdot \boldsymbol{H}^{\boldsymbol{T}} \cdot \left[\boldsymbol{R}^{-\boldsymbol{1}} + \boldsymbol{H} \cdot \boldsymbol{P}^{\boldsymbol{b}} \cdot \boldsymbol{H}^{\boldsymbol{T}}\right]^{-\boldsymbol{1}} \cdot \Delta \boldsymbol{Y} \in \mathbb{R}^{n \times N}, \tag{8}$$

where the analysis covariance matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is given by $\boldsymbol{A} = \left[\left[\boldsymbol{P}^{\boldsymbol{b}}\right]^{-\boldsymbol{1}} + \boldsymbol{H}^{\boldsymbol{T}} \cdot \boldsymbol{R}^{-\boldsymbol{1}} \cdot \boldsymbol{H}\right]^{-\boldsymbol{1}}$, the matrix of innovations (on observations) is denoted by $\Delta \boldsymbol{Y} = \boldsymbol{Y}^{\boldsymbol{s}} - \boldsymbol{H} \cdot \boldsymbol{X}^{\boldsymbol{b}} \in \mathbb{R}^{n \times N}$, and the matrix of perturbed observations reads $\boldsymbol{Y}^{\boldsymbol{s}} = \boldsymbol{y} \cdot \boldsymbol{1}_{\boldsymbol{N}}^{\boldsymbol{T}} + \boldsymbol{E} \in \mathbb{R}^{m \times N}$, where the columns of $\boldsymbol{E} \in \mathbb{R}^{m \times N}$ are formed by samples from a normal distribution with moments $\boldsymbol{\mathcal{N}}(\boldsymbol{0}, \boldsymbol{R})$. In practice, one of the most utilized formulations is given by (8).

2.2. Cholesky and modified Cholesky decomposition

The forms of the Cholesky and the modified Cholesky decomposition were discussed by Golub and Van Loan in [5]. If $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is symmetric positive definite, then there exist a unique lower triangular $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ with positive diagonal entries such that $\boldsymbol{A} = \boldsymbol{L} \cdot \boldsymbol{L}^{\boldsymbol{T}}$. If all the leading principal submatrices of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ are non-singular, then there exist unique lower triangular matrices $\boldsymbol{L}$ and $\boldsymbol{M}$ and a unique diagonal matrix $\boldsymbol{D} = \boldsymbol{diag}(d_1, d_2, \ldots, d_n)$ such that $\boldsymbol{A} = \boldsymbol{L} \cdot \boldsymbol{D} \cdot \boldsymbol{M}^{\boldsymbol{T}}$. If $\boldsymbol{A}$ is symmetric, then $\boldsymbol{L} = \boldsymbol{M}$ and $\boldsymbol{A} = \boldsymbol{L} \cdot \boldsymbol{D} \cdot \boldsymbol{L}^{\boldsymbol{T}}$. Golub and Van Loan provide proofs of those

decompositions, as well as various ways to compute them. We can use the Cholesky factor of a covariance matrix to quickly solve linear systems that involve a covariance matrix. Note that if we compute the Cholesky factorization and solve the triangular system $\boldsymbol{L}\cdot\boldsymbol{y}=\boldsymbol{b}$ and $\boldsymbol{L}^{\boldsymbol{T}}\cdot\boldsymbol{x}=\boldsymbol{y}$, then

$$\boldsymbol{A}\cdot\boldsymbol{x}=\left(\boldsymbol{L}\cdot\boldsymbol{L}^{\boldsymbol{T}}\right)\cdot\boldsymbol{x}=\boldsymbol{L}\cdot\left(\boldsymbol{L}^{\boldsymbol{T}}\cdot\boldsymbol{x}\right)=\boldsymbol{L}\cdot\boldsymbol{y}=\boldsymbol{b}. \tag{9}$$

For a symmetric matrix $\boldsymbol{A}\in\boldsymbol{R}^{n\times n}$, the modified Cholesky decomposition involves finding a non-negative diagonal matrix $\boldsymbol{E}$ such that $\boldsymbol{A}+\boldsymbol{E}$ is positive definite. In particular, $\boldsymbol{E}=\boldsymbol{0}$ whenever $\boldsymbol{A}$ is positive definite, otherwise $\|\boldsymbol{E}\|$ should be sufficiently small. This allows applying the usual Cholesky factorization to $\boldsymbol{A}+\boldsymbol{E}$, i.e. finding matrices $\boldsymbol{L},\boldsymbol{D}\in\mathbb{R}^{n\times n}$ such that $\boldsymbol{A}+\boldsymbol{E}=\boldsymbol{L}\cdot\boldsymbol{D}\cdot\boldsymbol{L}^{\boldsymbol{T}}$.

Recall $\Delta\boldsymbol{X}^{\boldsymbol{b}}=\boldsymbol{X}^{\boldsymbol{b}}-\overline{\boldsymbol{x}}^{\boldsymbol{b}}\otimes\boldsymbol{1}_{\boldsymbol{N}}^{\boldsymbol{T}}\in\mathbb{R}^{n\times N}$. Thus, $\Delta\boldsymbol{x}^{\boldsymbol{b}[\boldsymbol{i}]}\sim\boldsymbol{\mathcal{N}}(\boldsymbol{0}_{\boldsymbol{n}},\boldsymbol{B})$ for $\boldsymbol{1}\le\boldsymbol{i}\le\boldsymbol{n}$. Let $\boldsymbol{x}^{\boldsymbol{b}[\boldsymbol{i}]}\in\mathbb{R}^{N\times 1}$, the vector holding the ith row across all columns of $\Delta\boldsymbol{X}^{\boldsymbol{b}}$ for $\boldsymbol{1}\le\boldsymbol{i}\le\boldsymbol{n}$. Bickel and Levina in [1] discussed a modified Cholesky decomposition for the estimation of precision covariance matrix, and this allows in our context to estimate of $\boldsymbol{B}^{-\boldsymbol{1}}$ by fitting regressions of the form:

$$\boldsymbol{x}^{\boldsymbol{b}[\boldsymbol{i}]}=\sum_{\boldsymbol{j}=\boldsymbol{1}}^{\boldsymbol{i}-\boldsymbol{1}}\boldsymbol{x}^{\boldsymbol{b}[\boldsymbol{j}]}\cdot\beta_{i,j}+\boldsymbol{\xi}^{[\boldsymbol{i}]}\in\mathbb{R}^{N\times 1}, \tag{10}$$

then, $\{\boldsymbol{L}\}_{ij}=-\beta_{i,j}$, for $1\le i<j\le n$.

3. Filters based on modified Cholesky decomposition

3.1. Ensemble Kalman filter based on modified Cholesky

In the ensemble Kalman filter based on a modified Cholesky decomposition (EnKF-MC) [6], the analysis ensemble is estimated by using the following equations,

$$\boldsymbol{X}^{\boldsymbol{a}}=\boldsymbol{X}^{\boldsymbol{b}}+\left[\boldsymbol{B}^{-\boldsymbol{1}}+\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{R}^{-\boldsymbol{1}}\cdot\boldsymbol{H}\right]^{-\boldsymbol{1}}\cdot\boldsymbol{H}^{\boldsymbol{T}}\cdot\boldsymbol{R}^{-\boldsymbol{1}}\cdot\Delta\boldsymbol{Y}, \tag{11}$$

and

$$\Delta\boldsymbol{Y}=\left[\boldsymbol{Y}^{\boldsymbol{S}}-\boldsymbol{H}.\boldsymbol{X}^{\boldsymbol{b}}\right]\in\mathbb{R}^{\boldsymbol{n}\times\boldsymbol{N}}. \tag{12}$$

In this context, error correlations are estimated via a modified Cholesky decomposition. This provides an estimate of the inverse background error covariance matrix of the form,

$$\boldsymbol{B}^{-\boldsymbol{1}}\approx\widehat{\boldsymbol{B}}^{-\boldsymbol{1}}=\boldsymbol{L}^{\boldsymbol{T}}\cdot\boldsymbol{D}^{-\boldsymbol{1}}\cdot\boldsymbol{L}\in\mathbb{R}^{\boldsymbol{n}\times\boldsymbol{n}}, \tag{13}$$

or equivalently

$$\boldsymbol{B} \approx \widehat{\boldsymbol{B}} = \boldsymbol{L}^{-1} \cdot \boldsymbol{D} \cdot \boldsymbol{L}^{-T} \in \mathbb{R}^{n \times n}, \tag{14}$$

where $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ is a lower-triangular matrix whose diagonal elements are all ones, and $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix. Even more, when only local effects are considered during the estimation of $\widehat{\boldsymbol{B}}^{-1}$, sparse estimators of the precision background can be obtained. Furthermore, the matrix $\boldsymbol{L}$ can be sparse with only a few non-zero elements per row. Typically, the number of non-zero elements depends on the radius of influence during the estimation of background error correlations. For instance, in the one-dimensional case, the radius of influence denotes the maximum number of non-zero elements, per row, in $\boldsymbol{L}$. The EnKF-MC is then obtained by plugging in the estimator (3) in (6). Given the structure of the Cholesky factors, the EnKF-MC can be seen as a matrix-free implementation of the EnKF.

Recall that the precision analysis covariance matrix reads,

$$\boldsymbol{A}^{-1} = \boldsymbol{B}^{-1} + \boldsymbol{H}^{T} \cdot \boldsymbol{R}^{-1} \cdot \boldsymbol{H} \cdot \mathbb{R}^{n \times n}, \tag{15}$$

moreover, since $\boldsymbol{H}^{T} \cdot \boldsymbol{R}^{-1} \cdot \boldsymbol{H} \in \mathbb{R}^{n \times n}$ can be written as a sum of m rank-one matrices, the factors (3) can be updated in order to obtain an estimate of the inverse analysis covariance matrix. The next section discussed such approximation in detail.

3.1.1. Posterior ensemble Kalman filter stochastic (PEnKF-S)

In the stochastic posterior ensemble Kalman filter (PEnKF-S) [7, 8], we want to estimate the moments of the analysis distribution,

$$\boldsymbol{x} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{x}^{a}, \boldsymbol{A}), \tag{16}$$

based on the background ensemble $\left(\boldsymbol{X}^{b}\right)$, where $\boldsymbol{x}^{a}$ is the analysis state and $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the analysis covariance matrix. Consider the estimate of the inverse background error covariance matrix Eq. (3), the precision analysis covariance matrix Eq. (4) can be approximated as follows:

$$\boldsymbol{A}^{-1} \approx \widehat{\boldsymbol{A}}^{-1} = \widehat{\boldsymbol{B}}^{-1} + \boldsymbol{X} \cdot \boldsymbol{X}^{T}, \tag{17}$$

where $\boldsymbol{X} = \boldsymbol{H}^{T} \cdot \boldsymbol{R}^{-\frac{1}{2}} \in \mathbb{R}^{n \times m}$. The matrix (17) can be written as follows:

$$\widehat{\boldsymbol{A}}^{-1} = \boldsymbol{L}^{T} \cdot \boldsymbol{D}^{-1} \cdot \boldsymbol{L} + \sum_{i=1}^{m} \boldsymbol{x}_{i} \cdot \boldsymbol{x}_{i}^{T},$$

where $\boldsymbol{x}_{i}$ denotes the ith column of matrix $\boldsymbol{X}$, for $1 \leq i \leq m$. Consider the sequence of factor updates,

$$\left[\boldsymbol{L^i}\right]^{\boldsymbol{T}} \cdot \boldsymbol{D^i} \cdot \boldsymbol{L^i} = \left[\boldsymbol{L^{(i-1)}}\right]^{\boldsymbol{T}} \cdot \boldsymbol{D^{i-1}} \cdot \boldsymbol{L^{i-1}} + \boldsymbol{x_i} \cdot \boldsymbol{x_i^T}$$

$$\left[\boldsymbol{L^i}\right]^{\boldsymbol{T}} \cdot \boldsymbol{D^i} \cdot \boldsymbol{L^i} = \left[\boldsymbol{L^{(i-1)}}\right]^{\boldsymbol{T}} \cdot \left[\boldsymbol{D^{(i-1)}} + \boldsymbol{p_i} \cdot \boldsymbol{p_i^T}\right] \cdot \boldsymbol{L^{(i-1)}}$$

$$\left[\boldsymbol{L^i}\right]^{\boldsymbol{T}} \cdot \boldsymbol{D^i} \cdot \boldsymbol{L^i} = \left[\tilde{\boldsymbol{L}}^{(i-1)} \cdot \boldsymbol{L^{(i-1)}}\right]^{\boldsymbol{T}} \cdot \tilde{\boldsymbol{D}}^{(i-1)} \cdot \left[\tilde{\boldsymbol{L}}^{(i-1)} \cdot \boldsymbol{L^{(i-1)}}\right],$$

where $\boldsymbol{L^{i-1}} \cdot \boldsymbol{p_i} = \boldsymbol{x_i} \in \mathbb{R}^{n \times 1}$, for $1 \leq i \leq m$, $\widehat{\boldsymbol{B}}^{-1} = \left[\boldsymbol{L^{(0)}}\right]^{\boldsymbol{T}} \cdot \boldsymbol{D^{(0)}} \cdot \boldsymbol{L^{(0)}}$ and

$$\boldsymbol{D^{(i)}} + \boldsymbol{p_i} \cdot \boldsymbol{p_i^T} = \left[\tilde{\boldsymbol{L}}^{(i)}\right]^{\boldsymbol{T}} \cdot \tilde{\boldsymbol{D}}^{(i)} \cdot \tilde{\boldsymbol{L}}^{(i)} \in \mathbb{R}^{n \times n}, \tag{18}$$

We can use of Dolittle's method in order to compute the factors $\tilde{\boldsymbol{D}}^{(i)}$ and $\tilde{\boldsymbol{L}}^{(i)}$ in (9), and it is enough to note that,

$$\underbrace{\begin{bmatrix} 1 & \tilde{l}_{21} & \tilde{l}_{31} & \cdots & \tilde{l}_{n1} \\ 0 & 1 & \tilde{l}_{32} & \cdots & \tilde{l}_{n2} \\ 0 & 0 & 1 & \cdots & \tilde{l}_{n3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{\left[\tilde{\boldsymbol{L}}^{(i)}\right]^{\boldsymbol{T}}} \underbrace{\begin{bmatrix} \tilde{d}_1 & 0 & 0 & 0 & 0 \\ 0 & \tilde{d}_2 & 0 & 0 & 0 \\ 0 & 0 & \tilde{d}_3 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \tilde{d}_n \end{bmatrix}}_{\left[\tilde{\boldsymbol{D}}^{(i)}\right]} \underbrace{\begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ \tilde{l}_{21} & 1 & 0 & \ldots & 0 \\ \tilde{l}_{31} & \tilde{l}_{21} & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \tilde{l}_{n1} & \tilde{l}_{n2} & \tilde{l}_{n3} & \ldots & 1 \end{bmatrix}}_{\left[\tilde{\boldsymbol{L}}^{(i)}\right]}$$

$$= \underbrace{\begin{bmatrix} d_1 + p_1^2 & p_1 \cdot p_2 & p_1 \cdot p_3 & \cdots & p_1 \cdot p_n \\ p_2 \cdot p_1 & d_2 + p_2^2 & p_2 \cdot p_3 & \cdots & p_2 \cdot p_n \\ p_3 \cdot p_1 & p_3 \cdot p_2 & d_3 + p_3^2 & \cdots & p_3 \cdot p_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_n \cdot p_1 & p_n \cdot p_2 & p_n \cdot p_3 & \ldots & d_n + p_n^2 \end{bmatrix}}_{\left[\boldsymbol{D^{(i)}} + \boldsymbol{p_i} \cdot \boldsymbol{p_i^T}\right]}$$

After some math simplifications, the next equations are obtained,

$$\tilde{d}_k = p_k^2 + d_k - \sum_{q=k+1}^{n} \tilde{d}_q \cdot \tilde{l}_{qi}^2, \tag{19}$$

and

$$\tilde{l}_{kj} = \frac{1}{\tilde{d}_k} \cdot \left[p_k \cdot p_j - \sum_{q=k+1}^{n} \tilde{d}_q \cdot \tilde{l}_{qi} \cdot \tilde{l}_{qj}\right], \tag{20}$$

for $1 \leq k < n$ and $1 \leq j \leq k-1$. The set of Eqs. (19) and (20) can be used in order to derive an algorithm for rank-one update of Cholesky factors, and the updating process is shown in Algorithm 1.

Algorithm 1. Rank-one update for the factors $\boldsymbol{L}^{(i-1)}$ and $\boldsymbol{D}^{(i-1)}$

1. **function** UPD_CHOLESKY_FACTORS $\left(\boldsymbol{L}^{(i-1)}, \boldsymbol{D}^{(i-1)}, \boldsymbol{x}_i\right)$
2. Compute $\boldsymbol{p}_i$ from $\left[\boldsymbol{L}^{\boldsymbol{i}}\right]^{\boldsymbol{T}} \cdot \boldsymbol{p}_{\boldsymbol{i}} = \boldsymbol{x}_{\boldsymbol{i}}$
3. **for** $k = n \sim 1$ **do**
4. Compute $\tilde{d}_k$ via Eq. (10).
5. Set $l_{kk} \leftarrow 1$.
6. **for** $j = 1 \rightarrow k-1$ **do**
7. Compute $\tilde{l}_{kk}$ according to (11).
8. end for
9. end for
10. Set $\boldsymbol{L}^{(\boldsymbol{i})} \leftarrow \boldsymbol{L}^{(\boldsymbol{i-1})} \cdot \boldsymbol{L}^{(\boldsymbol{i-1})}$ and $\boldsymbol{D}^{(\boldsymbol{i})} \leftarrow \tilde{\boldsymbol{D}}^{(\boldsymbol{i})}$.
11. return $\boldsymbol{L}^{(\boldsymbol{i})}$, $\boldsymbol{D}^{(\boldsymbol{i})}$
12. end function

Algorithm 1 can be used in order to update the factors of $\widehat{\boldsymbol{B}}^{-\mathbf{1}}$ for all column vectors in $\boldsymbol{X}$, and this process is detailed in Algorithm 2.

Algorithm 2. Computing the factors $\boldsymbol{L}^{(m)}$ and $\boldsymbol{D}^{(m)}$ of $\left[\boldsymbol{L}^{(m)}\right]^{T} \cdot \boldsymbol{D}^{(m)} \cdot \boldsymbol{L}^{(m)}$

13. **function** COMPUTE_ANALYSIS_FACTOR $\left(\boldsymbol{L}^{(\mathbf{0})}, \boldsymbol{D}^{(\mathbf{0})}, \boldsymbol{H}, \boldsymbol{R}\right)$
14. Set $\boldsymbol{X} \leftarrow \boldsymbol{H}^{\boldsymbol{T}} \cdot \boldsymbol{R}^{-\frac{1}{2}}$.
15. **for** $i = 1 \rightarrow m$ do
16. Set $\left[\boldsymbol{L}^{(i)}, \boldsymbol{D}^{(i)}\right] \leftarrow$ UPD_CHOLESKY_FACTORS $\left(\boldsymbol{L}^{(i-1)}, \boldsymbol{D}^{(i-1)}, \boldsymbol{x}_i\right)$
17. **end for**
18. **return** $\boldsymbol{L}^{(m)}$, $\boldsymbol{D}^{(m)}$
19. **end function**

Once the updating process has been performed, the resulting factors form an estimate of the inverse analysis covariance matrix,

$$\widehat{\boldsymbol{A}}^{-1} = \left[\boldsymbol{L}^{(m)}\right]^{T} \cdot \boldsymbol{D}^{(m)} \cdot \boldsymbol{L}^{(m)} \in \mathbb{R}^{n \times n}, \tag{21}$$

From this covariance matrix, the posterior mode of the distribution can be approximated as follows:

$$\overline{\boldsymbol{x}}^{a} = \overline{\boldsymbol{x}}^{b} + \boldsymbol{z} \in \mathbb{R}^{n \times 1}, \tag{22}$$

where

$$\left[\boldsymbol{L}^{(m)}\right]^{T} \cdot \boldsymbol{D}^{(m)} \cdot \boldsymbol{L}^{(m)} \cdot \boldsymbol{z} = \boldsymbol{q}, \tag{23}$$

with $\boldsymbol{q} = \boldsymbol{H}^{T} \cdot \boldsymbol{R}^{-1} \cdot \left[\boldsymbol{y} - \boldsymbol{H} \cdot \overline{\boldsymbol{x}}^{b}\right] \in \mathbb{R}^{n \times 1}$. Note that the linear system (23) involves lower and upper triangular matrices, and therefore, $\overline{\boldsymbol{x}}^{a}$ can be estimated without the need of matrix inversion. Once the posterior mode is computed, the analysis ensemble is built about it. Note that $\widehat{\boldsymbol{A}}$ reads $\widehat{\boldsymbol{A}} = \left[\boldsymbol{L}^{(m)}\right]^{-1} \cdot \left[\boldsymbol{D}^{(m)}\right]^{-1} \cdot \left[\boldsymbol{L}^{(m)}\right]^{-T}$, and therefore, a square root of $\widehat{\boldsymbol{A}}$ can be approximated as follows:

$$\widehat{\boldsymbol{A}}^{\frac{1}{2}} = \left[\boldsymbol{L}^{(m)}\right]^{-1} \cdot \left[\boldsymbol{D}^{(m)}\right]^{-\frac{1}{2}} \in \mathbb{R}^{n \times n}, \tag{24}$$

which can be utilized in order to build the analysis ensemble,

$$\boldsymbol{X}^{a} = \overline{\boldsymbol{x}}^{a} \cdot \boldsymbol{1}_{N}^{T} + \Delta \boldsymbol{X}^{a}, \tag{25}$$

where $\Delta \boldsymbol{X}^{a} \in \mathbb{R}^{n \times N}$ is given by the solution of the linear system,

$$\boldsymbol{L}^{(m)} \cdot \left[\boldsymbol{D}^{(m)}\right]^{-\frac{1}{2}} \cdot \Delta \boldsymbol{X}^{a} = \boldsymbol{W} \in \mathbb{R}^{n \times N}, \tag{26}$$

In addition, the columns of $\boldsymbol{W} \in \mathbb{R}^{n \times N}$ are formed by samples from a multivariate standard normal distribution. Again, since $\boldsymbol{L}^{(m)}$ is lower triangular, the solution of (26) can be obtained readily.

3.1.2. Posterior ensemble Kalman filter deterministic (PEnKF-D)

The deterministic posterior ensemble Kalman filter (PEnKF-D) is a square root formulation of the PEnKF-S. The main difference between them lies on the use of synthetic data. In both methods, the matrices $\boldsymbol{A}^{-1}$ and $\boldsymbol{B}^{-1}$ are computed similarly (i.e., by using Dolittle's method and the algorithms described in the previous section).

In the PEnKF-D, the computation of $\widehat{\boldsymbol{B}}^{-1}$ is performed (13) based on the empirical moments of the ensemble. The analysis update is performed by using the perturbations

$$\widehat{\boldsymbol{B}}^{-\frac{1}{2}} \cdot \left(\boldsymbol{X}^{b} - \overline{\boldsymbol{x}}^{b} \cdot \mathbf{1}_{N}^{T}\right) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}, \boldsymbol{I}), \tag{27}$$

which are consistent with the dynamics of the numerical model, for instance, samples from (27) are driven by the physics and the dynamics of the numerical model. Finally, the analysis equation of the PEnKF-D

$$\boldsymbol{X}^{a} = \overline{\boldsymbol{x}}^{a} \cdot \mathbf{1}_{N}^{T} + \widehat{\boldsymbol{A}}^{\frac{1}{2}} \cdot \widehat{\boldsymbol{B}}^{-\frac{1}{2}} \cdot \Delta \boldsymbol{X}^{b} \in \mathbb{R}^{n \times N}, \tag{28}$$

where

$$\overline{\boldsymbol{x}}^{a} = \overline{\boldsymbol{x}}^{b} + \boldsymbol{A} \cdot \boldsymbol{H}^{T} \cdot \boldsymbol{R}^{-1} \cdot \left[\boldsymbol{y} - \boldsymbol{H} \cdot \overline{\boldsymbol{x}}^{b}\right], \tag{29}$$

$$\boldsymbol{A}^{\frac{1}{2}} \approx \widehat{\boldsymbol{A}}^{\frac{1}{2}} = \left[\tilde{\boldsymbol{L}}^{T} \cdot \tilde{\boldsymbol{D}}^{\frac{1}{2}}\right]^{-1}, \tag{30}$$

and

$$\boldsymbol{B}^{-\frac{1}{2}} \approx \widehat{\boldsymbol{B}}^{-\frac{1}{2}} = \boldsymbol{L} \cdot \boldsymbol{D}^{\frac{1}{2}}, \tag{31}$$

Since the moments of the posterior distribution are unchanged, we expect both methods PEnKF-S and PEnKF-D to perform equivalently in terms of errors during the estimation process.

3.2. Markov Chain Monte Carlo-based filters

Definitely, the EnKF represents a breakthrough in the data assimilation context, perhaps its biggest appeal is that we can obtain a closed form expression for the analysis members. Nevertheless, as can be noted, during the derivation of the analysis equations, some constrains are imposed, for instance, the observation operator is assumed linear, and therefore, the likelihood $\mathcal{P}(\boldsymbol{y}|\boldsymbol{x})$ obeys a Gaussian distribution. Of course, in practice, this assumption can be easily broken, and therefore, bias can be introduced on the posterior prediction; this is notoriously significant if one considers the fields in which data assimilation lives are submerged. From a statistical point of view, more specifically, under the Bayesian framework, this situation can be summarized as follows:

- The prediction is obtained from the posterior distribution: $\mathcal{P}(\boldsymbol{x}|\boldsymbol{y})$.
- The information from numerical model is known like the prior of background: $\mathcal{P}^{b}(\boldsymbol{x})$.
- The information from the observations is incorporated through the likelihood: $\mathcal{P}(\boldsymbol{y}|\boldsymbol{x})$.
- The posterior distribution is calculated: $\mathcal{P}(\boldsymbol{x}|\boldsymbol{y}) \propto \mathcal{P}^{b}(\boldsymbol{x}) \cdot \mathcal{P}(\boldsymbol{y}|\boldsymbol{x})$.
- If $\mathcal{P}^{b}(\boldsymbol{x})$ and $\mathcal{P}(\boldsymbol{y}|\boldsymbol{x})$ are Gaussian, $\mathcal{P}(\boldsymbol{y}|\boldsymbol{x})$ is Gaussian too, and the analysis is equal to the mean of $\mathcal{P}(\boldsymbol{y}|\boldsymbol{x})$.

A significant number of approaches have been proposed in the current literature in order to deal with these constraints; for instance, the particle filters (PFs) and the maximum likelihood ensemble filter (MLEF) are ensemble-based methods, which deal with nonlinear observation operators. Unfortunately, in the context of PF, its use is questionable under realistic weather forecast scenarios since the number of particles (ensemble members) increases exponentially regarding the number of components in the model state. Anyways, an extended analysis of these methods exceeds the scope of this document, but its exploration is highly recommended.

Taking samples directly from the posterior distribution is a strategy that can help to remove the bias induced by wrong assumptions on the posterior distribution (i.e., normal assumptions). We do not put the sights on finding the mode of the posterior distribution; instead of this, we want a set of state vectors that allow us to create a satisfactory representation of the posterior error distribution, then, based on these samples, it is possible to estimate moments of the posterior distribution from which the analysis ensemble can be obtained [9].

Hereafter, we will construct a modification of the sequential scheme of data assimilation; first, we will describe how to compute the analysis members by using variations in Markov Chain Monte Carlo (MCMC)-based methods, and then, we will include the modified Cholesky decomposition in order to estimate a precision background covariance.

An overview of the proposed method is as follows:

1. The forecast step is unchanged; the forecast ensemble is obtained by applying the model to the ensemble of states of the previous assimilation cycle.

2. The analysis step is modified, so that the analysis is not obtained anymore by, for instance (28), but we perform $\boldsymbol{k}$ iterations of an algorithm from the Markov Chain Monte Carlo (MCMC) family in order to obtain samples from the posterior distribution.

In order to be more specific in the explanation, let us define Metropolis-Hastings (MH) as the selected algorithm from the MCMC family. Now let us focus on MCMC in the most intuitive way possible. Let us define $\boldsymbol{J}(\boldsymbol{x},\boldsymbol{y})$ equal to the posterior pdf obtained from the expression $\boldsymbol{\mathcal{P}}(\boldsymbol{x}|\boldsymbol{y}) \propto \boldsymbol{\mathcal{P}}^{\boldsymbol{b}}(\boldsymbol{x}) \cdot \boldsymbol{\mathcal{P}}(\boldsymbol{y}|\boldsymbol{x})$. MH explores the state space in order to include model states in a so-called Markov Chain. The selection of candidates is based on the condition $\left(\boldsymbol{x}_{(c)},\boldsymbol{y}\right) > \boldsymbol{J}\left(\boldsymbol{x}_{(t-1)},\boldsymbol{y}\right)$, that is, if the value of $\boldsymbol{J}(\cdot)$ for a candidate $\boldsymbol{x}_{(c)}$ is greater than that for a previous vector $\boldsymbol{x}_{(t-1)}$. Candidates are generated from a pre-defined proposal distribution $\boldsymbol{Q}(\cdot)$; generally, a Gaussian multivariate distribution with mean $\boldsymbol{x}_{(t-1)}$ is chosen, and a covariance matrix is defined in order to handle the step sizes. Concisely, MCMC methods proceed as follows: at first iteration, the chain is started with $\boldsymbol{x}^{\boldsymbol{b}}$ vector. At end, the sample is obtained by extracting the last t vectors on the chain, and the other vectors are dismissed or "burned". In (32), we can see the expression to calculate the acceptance probability that relates the target pdf, $\boldsymbol{J}(\cdot)$, with the proposal $\boldsymbol{Q}(\cdot)$

$$\alpha = \frac{\boldsymbol{J}\left(\boldsymbol{x}_{(c)}|\boldsymbol{y}\right)\boldsymbol{Q}\left(\boldsymbol{x}_{(t-1)}|\boldsymbol{x}_{(c)}\right)}{\boldsymbol{J}\left(\boldsymbol{x}_{(t-1)}|\boldsymbol{y}\right)\boldsymbol{Q}\left(\boldsymbol{x}_{(c)}|\boldsymbol{x}_{(t-1)}\right)}, \tag{32}$$

The terms that involve $\boldsymbol{Q}(\cdot)$ can be canceled if a symmetric distribution is chosen as the proposal one, for instance, in the Gaussian distribution case. The procedure for the calculation of the analysis applying MH is described below:

1. Initialize the Markov Chain, $\boldsymbol{C}$, assigning the background value $\boldsymbol{x}^{\boldsymbol{b}}$ to $\boldsymbol{C}_{(0)}$.
2. Generate a candidate vector state $\boldsymbol{x}_{(c)}$, from$\boldsymbol{Q}$.
3. Obtain$\mathcal{U}$ from a uniform (0, 1) distribution.
4. If $\mathcal{U} \leq \alpha$ Then: $\boldsymbol{C}_{(t+1)} = \boldsymbol{x}_{(c)}$ Else: $\boldsymbol{C}_{(t+1)} = \boldsymbol{C}_{(t)}$
5. Repeat Steps 2 through 4, for $t = 1$ until $t = k - 1$
6. Remove the first p vectors of the chain, the burned ones.

The analysis is computed over the sample:

$$\boldsymbol{x}^{\boldsymbol{a}} = \frac{1}{(k-p)} \cdot \sum_{i=k-p}^{k} C_{(i)},$$

MCMC methods are straightforward to implement; when an enough number of iterations is performed, the behavior of the target function is captured on the chain [10]. This is true for even, complex density functions such as those with multiple modes. Briefly, let us focus on the fact that, generally, simulation-based methods such as MCMC explore a discretized grid, and as the mesh is refined, a huge number of iterations are needed before a high probability zone of the posterior distribution is reached [11]. Concretely, Hu et al. [12] proposed a family of modified MCMC dimension-independent algorithms under the name of preconditioned Crank-Nicolson (pCN) MCMC. These methods are robust regarding the curse of dimensionality in the statistical context. Initially, the Crank-Nicolson discretization is applied to a stochastic partial differential equation (SPDE) in order to obtain a new expression for the proposal distribution:

$$\boldsymbol{x}_{(c)} = \boldsymbol{x}_{(t-1)} - \frac{1}{2}\delta\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{L}}\left(\boldsymbol{x}_{(t-1)} + \boldsymbol{x}_{(c)}\right) + \sqrt{2\boldsymbol{\mathcal{K}}\delta}\boldsymbol{\epsilon}_0, \tag{33}$$

where $\boldsymbol{\mathcal{L}}$ is the precision matrix of $\boldsymbol{B}$, $\boldsymbol{\mathcal{K}}$ is a preconditioning matrix, $\boldsymbol{\epsilon}_0$ is white noise, if $\boldsymbol{\mathcal{K}} = \boldsymbol{B}$ and $\delta \in [0, 2]$ in (33), we get the pCN proposal described in (34):

$$\boldsymbol{x}_{(c)} = \boldsymbol{x}_{(t-1)} - \frac{1}{2}\delta\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{L}}\left(\boldsymbol{x}_{(t-1)} + \boldsymbol{x}_{(c)}\right) + \sqrt{2\boldsymbol{\mathcal{K}}\delta}, \tag{34}$$

where $\boldsymbol{\omega}\tilde{}\mathcal{N}(\boldsymbol{0}, \boldsymbol{B})$ and $\beta = 8\delta/(2+\delta)^2$. Of course, we use $\boldsymbol{P}^{\boldsymbol{b}}$ like a computationally efficient estimation of $\boldsymbol{B}$. The acceptance probability is defined in (35):

$$\alpha = min\{1, \exp(\boldsymbol{J}(\boldsymbol{x}_{t-1}, \boldsymbol{y}) - \boldsymbol{J}(\boldsymbol{x}_c, \boldsymbol{y}))\}, \tag{35}$$

The procedure for the calculation of the analysis applying pCN Metropolis-Hastings is described as follows:

1. Initialize the Markov Chain, $\boldsymbol{C}$, assigning the background value $\boldsymbol{x^b}$ to $\boldsymbol{C}_{(0)}$.
2. Generate a candidate vector state using (3): $\boldsymbol{x}_{(c)} = \left(\mathbf{1} - \beta^2\right)^{\frac{1}{2}} \boldsymbol{x}_{(t-1)} + \beta\boldsymbol{\omega}$.
3. Obtain $\mathcal{U}$ from a uniform (0, 1) distribution.
4. If $\mathcal{U} \leq \alpha$, then $\boldsymbol{C}_{(t+1)} = \boldsymbol{x}_{(c)}$. Else: $\boldsymbol{C}_{(t+1)} = \boldsymbol{C}_{(t)}$.
5. Repeat Steps 2 through 4, for $t = 1$ until $t = k - 1$.
6. Remove the first p vectors of the chain, the burned ones.
7. The analysis is calculated over the sample: $\boldsymbol{x^a} = \frac{1}{(k-p)} \cdot \sum_{i=k-p}^{k} C_{(i)}$.

Finally, in the data assimilation context, by using the modified Cholesky decomposition described in the earlier sections, the pCN-MH filter reads:

1. The forecast step remains unchanged; the forecast ensemble is obtained by applying the model to the ensemble of states of the previous iteration, but the estimation of background covariance matrix is calculated by modified Cholesky.
2. The update step is modified, so that the analysis is obtained by run k iterations of pCN-MH, to obtain a sample of the posterior error distribution.

Now we are ready to numerically test the methods discussed in this chapter.

4. Experimental results

4.1. Stochastic filter PEnKF-S

We assess the accuracy of the PEnKF-S and compare it against that of the LETFK implementation proposed by Hunt [13]. The numerical model is the Lorenz 96 model [11], which mimics the behavior of the atmosphere:

$$\frac{dx_k}{dt} = -x_{k-1} \cdot (x_{k-2} - x_{k+1}) - x_k + F, for\ 1 \leq k \leq n, \tag{36}$$

where n is the number of components in the model and F is an external force; with the value of the parameter, $F = 8$, the model presents a great entropy.

The experimental settings are described below:

1. An initial random solution $\boldsymbol{x}_{-3}^{+}$ is propagated for a while in time by using Lorenz 96 model and a fourth-order Runge Kutta method in order to obtain a vector state $\boldsymbol{x}_{-2}^{+}$ whose physics are consistent with the dynamics of such numerical model. This vector state serves as our reference solution.

2. The reference solution is perturbed by using samples from a normal distribution with parameters $\mathcal{N}(0_n, \sigma_B \cdot I)$. Three different values for σ_B are considered during the numerical experiments $\sigma_B \in \{0.05, 0.10, 0.15\}$. This perturbed state is propagated in time in order to make it consistent with the physics and dynamics of the numerical model. From here, an initial background state $\boldsymbol{x}_{-1}^{\boldsymbol{b}}$ is obtained.

3. A similar procedure is performed in order to build a perturbed ensemble about $\boldsymbol{x}_{-1}^{\boldsymbol{b}}$. The ensemble members are propagated in time from where an ensemble of model realizations $\left\{\boldsymbol{x}_{\boldsymbol{0}}^{\boldsymbol{b}[i]}\right\}_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{N}}$ of model is obtained.

4. The assimilation windows consist of 15 equidistant observations. The frequency of observations is 0.5 time units, which represents 3.5 days in the atmosphere.

5. The dimension of the vector state is $n = 40$. The external force of the numerical model is set to $\boldsymbol{F} = \mathbf{80}$.

6. The number of observed components is 50% of the dimension of the vector state.

7. Three ensemble sizes are tried during the experiments $\in \{20, 40, 60\}$.

8. As a measure of quality, the L_2 norm of the analysis state and the reference solution are computed across assimilation steps.

9. A total of 100 runs are performed for each pair (N, σ_B). For each run, a different initial random vector is utilized in order to build the initial perturbed reference solution $\boldsymbol{x}_{-3}^{+}$ (before the model is applied x_{-2}^{*}). This yields to different initial ensembles as well as synthetic data for the different runs of each configuration (pair).

The average of the error norms of each pair (N, σ_B) for the LETKF and the PEnKF implementations is shown in **Table 1**. As can be seen, on average across 100 runs, the

σ_B	N	LETKF	PEnKF-S
0.05	20	22,6166	21,2591
	40	20,5671	18,2548
	60	20,0567	17,8824
0.10	20	23,1742	21,0725
	40	20,9513	18,3542
	60	18,5048	17,8240
0.15	20	24,8201	20,9059
	40	21,1314	18,1731
	60	20,8487	17,7590

Table 1. Average of L-2 norm of errors for 100 runs of each configuration (σ_B, N) for the compared filter implementations.

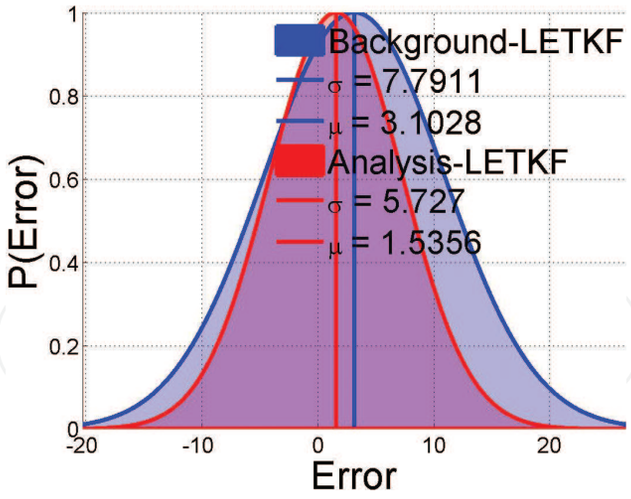

Figure 1. Local ensemble transform Kalmar filter (LETKF).

performance of the proposed EnKF implementation outperforms that of the LETKF in terms of L_2 norm of the error. Even more, the PEnKF-S seems to be invariant to the initial background error σ_B since, in all cases, when the ensemble size is increased a better estimation of the reference state $\boldsymbol{x}^*$ at different observation times is obtained. This can also obey to the estimation of background error correlations via the modified Cholesky decomposition since it is drastically improved whenever the ensemble size is increased as pointed out by Bickel and Levina in [1]. In such case, the error decreases by $\mathcal{O}(\log(n)/N)$. This is crucial in the PEnKF-S formulation since estimates of the precision analysis covariance matrix are obtained by rank-one updates on the inverse background error covariance matrix. On the other hand, in the LETKF context, increasing the ensemble size can improve the accuracy of the method but that is not better than the one shown by the PEnKF.

Some plots of the L_2 norm of error for the PEnKF and the LETKF across different configurations and runs are shown in **Figure 1**. Note that the error of the PEnKF decreases aggressively since the earlier iterations. In the LETKF context, the accuracy is similar to that of the PEnKF only at the end of the assimilation window.

4.2. Deterministic filter PEnKF-D

Holding the settings from the previous section, experiments are performed by using the PEnKF-D. The results have similar behavior to that of the LETFK and the PEnKF-S implementation proposed by Niño et al. in [8]. This can be appreciated in **Figures 1**–**3**. As can be seen, the error distributions reveal similar behavior across all compared filters.

4.3. MCMC filter based on modified Cholesky decomposition

We will describe experiments of our proposed filter based on a modified Cholesky decomposition and the preconditioned Crank-Nicolson Metropolis-Hastings. Again, the numerical

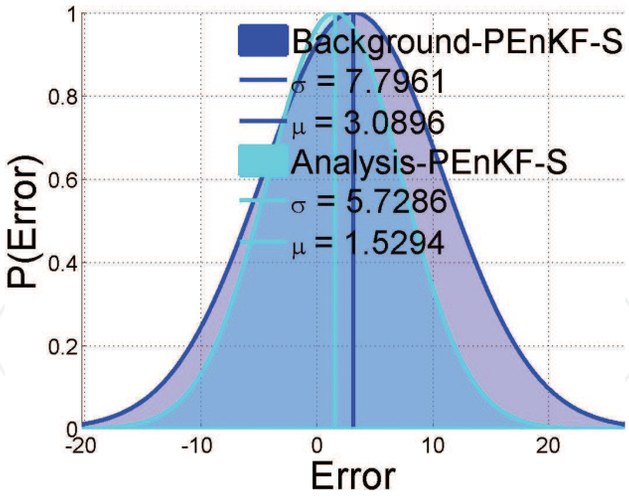

Figure 2. Posterior ensemble Kalmar filter stochastic (PEnKF-S).

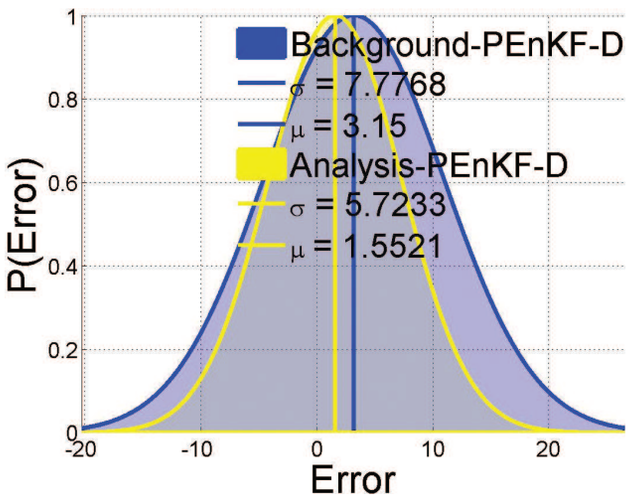

Figure 3. Posterior ensemble Kalmar filter deterministic (PEnKF-D).

model is the Lorenz 96 model. In this section, we are mainly interested on describing the behavior of the method, especially when the observational operator is nonlinear.

The experimental settings are as follows:

1. The observational operator is quadratic: $\{\mathcal{H}(x)\}_i = x_i^2$, where x_i denotes the ith component of the vector state x.
2. The vector of states has $n = 40$ components.
3. The ensemble size is $N = 20$.
4. The length of the Markov Chain is 1000.
5. The proposal is pCN with $\beta = 0.09$.

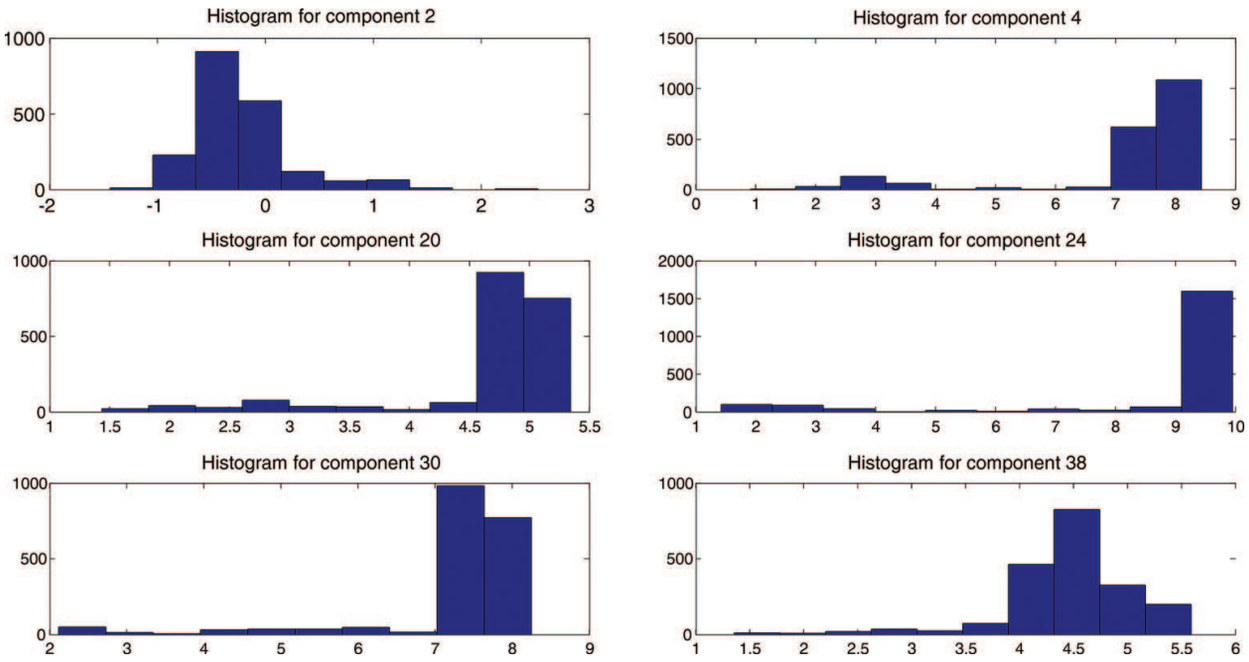

Figure 4. Histogram of six components of the state vector after 1000 iterations of the algorithm, the dashed lines indicate the position of the true value for the respective component.

Figure 4 describes the distribution of six of the 40 components of the state vector after 1000 iterations of the algorithm, attempting to visualize if the actual value is within or near the region of the highest probability density described by the sample contained in the Markov Chain.

Note that, not in all cases, the actual values of the model components were located at the peaks of the histogram, but most of them were within or near to zones of high probability described by the sample. This result is important if we take into account that probably the posterior distribution is not normal as is the case of quadratic observation operators.

5. Conclusions

In this chapter, efficient EnKF implementations were discussed. All of them are based on a modified Cholesky decomposition wherein a precision background covariance is obtained in terms of Cholesky factors. In the first filter, the PEnKF-S, synthetic data are utilized in order to compute the posterior members, as done in stochastic formulations of the filter. Even more, a sequence of rank-one updates can be applied over the factors of the prior precision matrix in order to estimate those of the posterior precision. In the second filter, the PEnKF-D, synthetic data are avoided by using perturbations obtained from the physics and the dynamics of the numerical model. Finally, a MCMC-based filter is obtained in order to reduce the impact of bias when Gaussian assumptions are broken during the assimilation of observations, for instance, when the observation operator is nonlinear. Numerical experiments with the Lorenz 96 model reveal that the proposed filters are comparable to filters from the specialized literature.

Acknowledgements

This work was supported by the Applied Math and Computational Science Lab (AML-CS) at Universidad del Norte in Barranquilla, Colombia.

Author details

Elias David Niño Ruiz*, Rolando Beltrán Arrieta and Alfonso Manuel Mancilla Herrera

*Address all correspondence to: enino@uninorte.edu.co

Applied Math and Computer Science Laboratory, Department of Computer Science, Universidad Del Norte, Barranquilla, Colombia

References

[1] Bickel PJ, Levina E. Covariance regularization by thresholding. The Annals of Statistics. 2008;**36**(6):2577-2604. DOI: 10.1214/08-AOS600

[2] Evensen G. The ensemble Kalman filter: Theoretical formulation and practical implementation. Ocean Dynamics. 2003;**53**(4):343-367

[3] Burgers G, Jan van Leeuwen P, Evensen G. Analysis scheme in the ensemble Kalman filter. Monthly Weather Review 1998;**126**(6):1719-1724

[4] Sakov P, Oke PR. A deterministic formulation of the ensemble Kalman filter: An alternative to ensemble square root filters. Tellus A. 2008;**60**(2):361-371

[5] Coleman TF, Van Loan C. Handbook for matrix computations. Society for Industrial and Applied Mathematics, 1988

[6] Nino-Ruiz ED, Sandu A, Deng X. A parallel implementation of the ensemble Kalman filter based on modified Cholesky decomposition. Journal of Computational Science. 2017

[7] Nino-Ruiz ED, Mancilla A, Calabria JC. A posterior ensemble Kalman filter based on a modified Cholesky decomposition. Procedia Computer Science. 2017;**108**:2049-2058

[8] Nino-Ruiz ED. A matrix-free posterior ensemble Kalman filter implementation based on a modified Cholesky decomposition. Atmosphere. 2017;**8**(7):125

[9] Attia A, Sandu A. A hybrid Monte Carlo sampling filter for non-gaussian data assimilation. AIMS Geosciences. 2015;**1**:41-78

[10] David H, Shane Reese C, David Moulton J, Vrugt JA, Colin F. Posterior exploration for computationally intensive. In: Steve B, Andrew G, Jones GL, Meng X-L, editors. Handbook of Markov Chain Monte Carlo. Chapman & Hall; 2011. p. 401-418

[11] Cotter SL, Roberts GO, Stuart AM, White D, et al. MCMC methods for functions: Modifying old algorithms to make them faster. Statistical Science. 2013;**28**(3):424-446

[12] Hu Z, Yao Z, Li J. On an adaptive preconditioned Crank–Nicolson MCMC algorithm for infinite dimensional Bayesian inference. Journal of Computational Physics. 2017;**332**:492-503

[13] Fatkullin I, Vanden-Eijnden E. A computational strategy for multiscale systems with applications to Lorenz 96 model. Journal of Computational Physics. 2004;**200**(2):605

IntechOpen