This article can be cited as E. D. Nino-Ruiz, Hybrid Ensemble Kalman Filter and Markov Chain Monte Carlo Implementations for Non-Gaussian Data Assimilation, International Journal of Artificial Intelligence, vol. 18, no. 2, pp. 24-50, 2020.

Hybrid Ensemble Kalman Filter and Markov Chain Monte Carlo Implementations for Non-Gaussian Data Assimilation

Elias D. Nino-Ruiz[1]

[1]Applied Math and Computer Science Lab
Department of Computer Science
Universidad del Norte
Barranquilla 080001, Colombia
enino@uninorte.edu.co

ABSTRACT

This paper proposes two efficient matrix-free ensemble Kalman filter implementations for non-linear Data Assimilation (DA). Starting with an ensemble of model realizations, the proposed methods employ Markov chains and Gaussian kernels to sample from posterior error distributions. The first method employs a Random-Walk to propose candidates, while the second one does it via the pre-conditioned Crank-Nicholson (pCN) proposal distribution. For the pCN formulation, an iterative matrix-free method is employed to generate samples from the proposal distribution. The posterior ensemble can then be built similar to that of the posterior ensemble Kalman filter implementation. Experimental tests are performed by using the Lorenz-96 model. Two observational operators are employed: a non-smooth operator and an exponential one. For full observational networks, prior and posterior errors differ by order of magnitudes. In terms of Root-Mean-Square-Errors, prior errors are decreased by several order of magnitudes for observation coverages of 70%, 80%, and 90% model components.

Keywords: Non-Gaussian data assimilation, ensemble Kalman filter, Markov Chain Monte Carlo

2000 Mathematics Subject Classification: 13P25, 65C60, 97K80.

1 Introduction

Numerical models are built to mimic the behaviour of physical systems (Roman, Precup, Bojan-Dragos and Szedlak-Stinean, 2019; Gil, Johanyák and Kovács, 2018; Nowaková, Prílepok and Snášel, 2017), in practice, these are employed to perform forecast of real-life complex dynamics. However, since numerical models are imperfect, operational forecasts are valid for short time periods (i.e., 48 hours). Thus, imperfect numerical forecast must be corrected by employing, for instance, real-life observations (i.e., sensor data). This is commonly done in the context of sequential Data Assimilation (DA). In sequential DA, a numerical forecast $\mathbf{x}^b \in \mathbb{R}^{n\times 1}$ is modified to account for information brought by an observation $\mathbf{y} \in \mathbb{R}^{m\times 1}$ (Pham, 2001; Ruiz and Sandu, 2016), where n and m are the spatial domain resolution and the number

of observations, respectively. In this framework, the state to estimate $\mathbf{x}^* \in \mathbb{R}^{n\times 1}$ evolves according to some numerical model,

$$\mathbf{x}^*_{next} = \mathcal{M}_{t_{current}\to t_{next}}\left(\mathbf{x}^*_{current}\right) , \tag{1.1}$$

Typically, the estimation process rely on Gaussian assumptions on prior and observational errors. This mainly obeys to computational aspects. Thus, for any $\mathbf{x} \in \mathbb{R}^{n\times 1}$, Gaussian models are fitted for background errors,

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{x}^b,\, \mathbf{B}\right) , \tag{1.2}$$

and observational ones,

$$\mathbf{y} \sim \mathcal{N}\left(\mathcal{H}\left(\mathbf{x}\right),\, \mathbf{R}\right) , \tag{1.3}$$

where $\mathbf{B} \in \mathbb{R}^{n\times n}$ holds the background error covariances and $\mathbf{R} \in \mathbb{R}^{m\times m}$ is the estimated data error covariance matrix. Likewise, $\mathcal{H} : \mathbb{R}^{n\times 1} \to \mathbb{R}^{m\times 1}$ is the (non-linear) observation operator which projects vector states to the observation space. The kernel of the analysis distribution of errors can be estimated via Bayes' rule as follows:

$$\mathcal{P}\left(\mathbf{x}|\mathbf{y}\right) \propto \exp\left(-\mathcal{J}(\mathbf{x})\right) , \tag{1.4}$$

where $\mathcal{J}(\mathbf{x})$ is the well-known Three-Dimensional-Variational (3D-Var) cost function (Buehner, 2005):

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}\cdot\left\|\mathbf{x}-\mathbf{x}^b\right\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\cdot\left\|\mathbf{y}-\mathcal{H}\left(\mathbf{x}\right)\right\|^2_{\mathbf{R}^{-1}} . \tag{1.5}$$

The posterior mode of (1.4) provides the best approximation to $\mathbf{x}^*$ once observations have been digested, it can be approximated by solving the 3D-Var optimization problem,

$$\mathbf{x}^a = \arg\min_{\mathbf{x}} \mathcal{J}(\mathbf{x}) , \tag{1.6}$$

where the posterior mode $\mathbf{x}^a \in \mathbb{R}^{n\times 1}$ is better known as the analysis state. When $\mathcal{H}$ is (nearly) linear, ensemble based methods such as the ensemble Kalman filter (EnKF) (Evensen, 2009; Lorenc, 2003) and the local ensemble transform Kalman filter (Hunt, Kostelich and Szunyogh, 2007; Ott, Hunt, Szunyogh, Zimin, Kostelich, Corazza, Kalnay, Patil and Yorke, 2008) (LETKF) can be successfully utilized in order to, among other things, estimate the posterior mode (1.6). Nevertheless, for non-linear (and non-smooth) observation operators as those found in practice, the Gaussian assumption on (1.3) can be broken and therefore, the EnKF and the LETKF methods would likely fail. To account for non-linear observation operators during assimilation steps, alternatives to EnKF formulations such as, for instance, Particle Filters (PF) (van Leeuwen, 2010) are proposed in the current literature. However, in practice, the model dimension n ranges in $\mathcal{O}\left(10^8\right)$ and since the number of particles (model realizations) increases exponentially regarding n, the use of PF methods under realistic weather forecast scenarios can be questionable (Snyder, Bengtsson, Bickel and Anderson, 2008). In general,

robust methods have been proposed in order to draw samples from non-Gaussian (or even more non-known) probability distributions (Attia, Rao and Sandu, 2015). Nevertheless, an open question is how those methods can be implemented under practical scenarios such as those found in the DA context. For instance, there is no doubt that Markov-Chain-Monte-Carlo (MCMC) methods are potential candidates to be utilized for that purpose but, how to speed-up the MCMC converge under high-dimensional probability spaces is one of the current challenges in many fields of science (Attia, Ştefănescu and Sandu, 2017; Attia, Rao and Sandu, 2017).

To support the development of MCMC methods in the context of DA, we think that gradient approximations of (1.5) can be employed to speed-up their convergence. For instance, the observation operator can be linearized about partial (approximated) solutions of the optimization problem (1.6) in order to obtain candidates of $\mathbf{x}^a$ along a descent direction approximation of $\mathcal{J}(\mathbf{x})$. Likewise, MCMC methods can afford the decision of whether or not to accept solutions.

This paper is organized as follows: in Section 2 ensemble-based methods as well as MCMC methods from the current literature are discussed. Section 3 presents two novel matrix-free DA methods for non-linear observation operators based on MCMC and convex approximations of the $\mathcal{J}(\mathbf{x})$ function. In Section 4 experimental tests are performed by using the Lorenz 96 model. Finally, conclusions are stated in Section 5.

2 Preliminaries

In this section, we briefly describe ensemble-based methods, their limitations, and how those can be overcome via sampling methods. We restrict ourselves here to specific topics that are closely related to our proposed EnKF formulations, but readers are properly addressed to papers with further and rigorous discussions of tangential concepts.

2.1 The Ensemble Kalman Filter

Monte Carlo methods are commonly employed to estimate population parameters. In this context, the ensemble Kalman filter employs an ensemble of model runs (Gillijns, Mendoza, Chandrasekar, De Moor, Bernstein and Ridley, 2006)

$$\mathbf{X}^b = \left[\mathbf{x}^{b[1]},\, \mathbf{x}^{b[2]},\, \ldots,\, \mathbf{x}^{b[N]}\right] \in \mathbb{R}^{n\times N}\,. \tag{2.1a}$$

to estimate the moments of the background error distribution (1.2) (Evensen, 2006):

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{x}^b,\, \mathbf{B}\right)\,.$$

Thus:

$$\mathbf{x}^b \approx \overline{\mathbf{x}}^b = \frac{1}{N}\cdot\sum_{e=1}^{N}\mathbf{x}^{b[e]} \in \mathbb{R}^{n\times 1}\,, \tag{2.1b}$$

and

$$\mathbf{B} \approx \mathbf{P}^b = \frac{1}{N-1} \cdot \mathbf{\Delta X} \cdot \mathbf{\Delta X}^T \in \mathbb{R}^{n \times n}\,, \tag{2.1c}$$

where $\mathbf{x}^{b[e]} \in \mathbb{R}^{n \times 1}$ stands for the e-th model realization, for $1 \leq e \leq N$, $\overline{\mathbf{x}}^b$ and $\mathbf{P}^b$ are the ensemble mean and the ensemble covariance matrix, respectively. Likewise, $\mathbf{\Delta X} \in \mathbb{R}^{n \times N}$ is the matrix of member deviations,

$$\mathbf{\Delta X} = \mathbf{X}^b - \overline{\mathbf{x}}^b \cdot \mathbf{1}^T\,, \tag{2.1d}$$

where $\mathbf{1}$ is a vector of consistent dimension whose components are all ones. The assimilation process, for instance, can be performed stochastically as follows,

$$\mathbf{X}^a = \mathbf{X}^b + \left[\left[\mathbf{P}^b\right]^{-1} + \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{H}\right]^{-1} \cdot \mathbf{\Delta Y} \in \mathbb{R}^{n \times N}\,, \tag{2.2}$$

where $\mathcal{H}'(\mathbf{x}) \approx \mathbf{H}^T \in \mathbb{R}^{n \times m}$ is the Jacobian of the non-linear observation operator (whose expansion point is chosen as the background state), and the innovation matrix $\mathbf{\Delta Y} \in \mathbb{R}^{n \times m}$ reads,

$$\mathbf{\Delta Y} = \mathbf{H}^T \cdot \mathbf{R}^{-1} \cdot \left[\mathbf{y} \cdot \mathbf{1}^T + \mathbf{R}^{1/2} \cdot \mathbf{E} - \mathcal{H}\left(\mathbf{X}^b\right)\right]\,,$$

and the columns of matrix $\mathbf{E} \in \mathbb{R}^{m \times N}$ are samples from a multivariate standard Normal distribution. Since current model resolutions range in the order of the millions while ensemble sizes does it in the hundreds, the ensemble covariance (2.1c) is typically rank-deficient. Across the years, localization methods have been proposed (Keppenne, 2000; Nino-Ruiz and Morales-Retat, 2018) in order to overcome this situation. This has triggered the formulation and the implementation of efficient EnKF based methods (Bishop and Toth, 1999; Ott, Hunt, Szunyogh, Zimin, Kostelich, Corazza, Kalnay, Patil and Yorke, 2004; Nino Ruiz, Sandu and Anderson, 2014). A recent EnKF implementation relies on the Bickel and Levina estimator (Bickel and Padilla, 2014) in order to estimate background error correlations: the EnKF based on a modified Cholesky decomposition (EnKF-MC) (Nino-Ruiz, Sandu and Deng, 2015; Nino, Sandu and Deng, 2016; Nino-Ruiz, Sandu and Deng, 2017; Nino-Ruiz, Mancilla and Calabria, 2017). In this method, for each model component $1 \leq i \leq n$, a neighbourhood $P(i\,, r)$ is defined based on its predecessors according to some labelling of model components and a radius of influence $r \in \mathbb{R}^+$, therefore (Nino-Ruiz, Sandu and Deng, 2018),

$$j \in P(i,\, r) \Leftrightarrow d\left(x_i,\, x_j\right)^2 \leq r^2, \text{ and } j < i\,,$$

where $d(\bullet,\, \bullet)$ denotes a consistent distance function. Notice, each model component is conditionally correlated only with its predecessors given all components, from here, a sparse precision matrix for the background distribution can be estimated as follows,

$$\widehat{\mathbf{B}}^{-1} = \widehat{\mathbf{L}}^T \cdot \widehat{\mathbf{D}}^{-1} \cdot \widehat{\mathbf{L}} \in \mathbb{R}^{n \times n}\,, \tag{2.3}$$

where the diagonal entries of the factor $\widehat{\mathbf{L}} \in \mathbb{R}^{n\times n}$ are all ones while its non-zero elements of row i are given by fitting models of the form,

$$\left[\mathbf{x}_{[i]}\right]^T = \sum_{j\in P(i,r)} -\left\{\widehat{\mathbf{L}}\right\}_{i,j} \cdot \left[\mathbf{x}_{[j]}\right]^T + \boldsymbol{\xi}^{[i]}\,, \text{ for } 1 \leq i \leq n\,,$$

where $\mathbf{x}_{[i]} \in \mathbb{R}^{N\times 1}$ denotes the i-th row of matrix, $\{\widehat{\mathbf{L}}\}_{i,j} \in \mathbb{R}$ is the $(i,\, j)$-th element of matrix $\widehat{\mathbf{L}}$, and the components of $\boldsymbol{\xi}^{[i]} \in \mathbb{R}^{N\times 1}$ are described by a zero-mean Normal distribution with unknown variance σ^2. Likewise, $\mathbf{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose diagonal entries are the empirical variances $\widehat{\mathbf{var}}(\bullet)$ of the residuals,

$$\left\{\widehat{\mathbf{D}}\right\}_{i,i} = \widehat{\mathbf{var}}\left(\left[\mathbf{x}_{[i]}\right]^T - \sum_{j\in P(i,r)} \left\{-\widehat{\mathbf{L}}\right\}_{i,j} \cdot \left[\mathbf{x}_{[j]}\right]^T\right) \approx \sigma^2\,, \tag{2.4}$$

for $2 \leq i \leq n$, with $\left\{\widehat{\mathbf{D}}\right\}_{1,1} = \widehat{\mathbf{var}}\left(\mathbf{x}_{[1]}\right)$. By replacing (2.3) in (2.2) the EnKF-MC is obtained. Another efficient EnKF implementation which exploits the structure of $\widehat{\mathbf{B}}^{-1}$ is the Posterior EnKF (P-EnKF) (Nino-Ruiz, 2017). On its square root formulation, the P-EnKF approximates the posterior covariance matrix by a sequence of rank-one updates over the prior factors in (2.3),

$$\widehat{\mathbf{A}}^{(j)} \;\; = \;\; \widehat{\mathbf{A}}^{(j-1)} + \mathbf{z}^{[j]} \cdot \left[\mathbf{z}^{[j]}\right]^T = \left[\mathbf{L}^{(j)}\right]^T \cdot \mathbf{D}^{(j)} \cdot \left[\mathbf{L}^{(j)}\right]\,, \text{for } 1 \leq j \leq m\,,$$

where $\widehat{\mathbf{B}}^{-1} = \left[\mathbf{L}^{(0)}\right]^T \cdot \mathbf{D}^{(0)} \cdot \mathbf{L}^{(0)} = \widehat{\mathbf{L}}^T \cdot \widehat{\mathbf{D}}^{-1} \cdot \widehat{\mathbf{L}}$, and $\mathbf{z}^{[j]}$ is the j-th column of a square root approximation of the information matrix $\mathbf{Z} = \mathbf{H}^T \cdot \mathbf{R}^{-1/2}$, for $1 \leq j \leq m$. We build the posterior ensemble as follows:

$$\mathbf{X}^a = \overline{\mathbf{x}}^a \cdot \mathbf{1}_N^T + \left[\widetilde{\mathbf{L}}^T \cdot \widetilde{\mathbf{D}}^{-1/2}\right]^{-1} \cdot \mathbf{E} \in \mathbb{R}^{n\times N}\,, \tag{2.5}$$

where the columns of $\mathbf{E} \in \mathbb{R}^{n\times N}$ are formed by samples from a multivariate standard Normal distribution, and the precision matrix $\widehat{\mathbf{A}}^{-1} = \widetilde{\mathbf{L}}^T \cdot \widetilde{\mathbf{D}}^{-1} \cdot \widetilde{\mathbf{L}} \in \mathbb{R}^{n\times n}$ is an estimate of the posterior error covariance matrix $\mathbf{A}^{-1}$. Note that, the direct inversion of matrix $\widetilde{\mathbf{L}}^T \cdot \widetilde{\mathbf{D}}^{-1/2} \in \mathbb{R}^{n\times n}$ is not actually performed since $\widetilde{\mathbf{L}}^T$ is a upper triangular matrix while $\widetilde{\mathbf{D}}^{-1/2}$ is diagonal and therefore, forward substitutions are sufficient in order to solve the subjacent linear system.
There are many other methods in the context of ensemble-based formulations for (nearly) linear observation operators, which we do not discuss further here. Recently, a complete survey of those is detailed by Bannister in (Bannister, 2016).

2.2 Non-Linear Observation Operators

For non-linear observation operators, EnKF based formulations can fail to obtain reasonable estimates of posterior error distributions. At this point, it is convenient to make use of, for instance, sampling methods (Sarma, 2009; Preitl, Precup, Preitl, Vaivoda, Kilyeni and Tar, 2007) and as we mentioned before, Markov-Chain-Monte-Carlo (MCMC) methods (Plummer, Best,

Cowles and Vines, 2006) can fit within this context. Roughly speaking, the idea behind MCMC methods is to implicitly draw samples from a target distribution $\pi(x)$ (from which, potentially, it is not clear how to do that) via samples from a proposal distribution $\phi(x)$ (i.e., Normal distribution). Interestingly, such samples, which form a chain, converge to high-probability zones of $\pi(x)$. This fact has been theoretically proven (Asmussen and Glynn, 2011). For instance, in the Metropolis algorithm, MCMC methods with symmetric proposal distributions work as is shown in the Algorithm 1. The main concern in this context is how to speed-up MCMC methods in large-dimensional probability spaces. Recently, this issue have been addressed by Cotter in (Cotter, Roberts, Stuart, White et al., 2013), a novel idea is based on the Crank-Nicholson proposal which, in the context of DA, has the form:

$$\left[\mathbf{I}+\frac{1}{2}\cdot\mathbf{T}\cdot\mathbf{B}^{-1}\right]\cdot\mathbf{z}_{CN}=\left[\mathbf{I}-\frac{1}{2}\cdot\mathbf{T}\cdot\mathbf{B}^{-1}\right]\cdot\mathbf{x}+\left[2\cdot\mathbf{T}\right]^{1/2}\boldsymbol{\xi}\,, \tag{2.6}$$

where $\mathbf{T}\in\mathbb{R}^{n\times n}$ is a pre-conditioner matrix, and $\boldsymbol{\xi}\in\mathbb{R}^{n\times 1}$ follows a multivariate standard Normal distribution. The idea behind this method is to avoid the Random-Walk provided by the traditional proposal (Lombardi, 2007; Sengupta, Friston and Penny, 2016),

$$\mathbf{z}_B=\mathbf{x}+\mathbf{B}^{1/2}\cdot\boldsymbol{\xi}\,, \tag{2.7}$$

and as a potential consequence, the number of burn-in steps in MCMC methods can be minimized. Thus, high-probability zones of the target distribution can be reached faster (Hu, Yao and Li, 2017; Beskos, Girolami, Lan, Farrell and Stuart, 2017).

There are many other families of methods that can be employed to compute a posterior mode of (1.6) in the general case. Some of them are based on optimization methods (Precup and Preitl, 2004; Zupanski, 2005; Fan, Huang, Baetz, Li, Huang, Li, Chen and Xiong, 2016), which exploit the use of gradient information. However, how those methods can be employed in DA wherein the number of variables ranges in the order of millions is an open question, and further research is needed in order to extrapolate those to practical contexts (Apte, Hairer, Stuart and Voss, 2007; Bannister, 2016). Besides, in most of the cases, gradient-driven optimization methods rely on smooth properties of gradient approximations in order to guaranty their convergence (Nino-Ruiz, 2018), and in DA, for instance, rough observation operators can be found.

The literature proposes sampling methods (Hadfield et al., 2010; Hoang, Schwab and Stuart, 2013) which can be contextualized in the DA framework but, how those can be implemented under, for instance, operational weather forecast scenarios is a current issue. We center our attention on the MCMC methods discussed here (which are widely known by the scientific community), and we think that, by proposing samples along steepest descent approximations of (1.5), MCMC methods can rapidly reach high-probability zones of the posterior error distribution (1.4). The next section proposes two efficient EnKF formulations based on this general idea.

Algorithm 1 General framework of the Metropolis algorithm.

function COMPUTE_CHAIN_RW($x^{(0)}$, v)

Require: Initial state $x^{(0)}$ in the chain, and number of iterations v.

Ensure: A chain whose elements converge to high-probability zones of $\pi(x)$.

for $k \leftarrow 0$ to v **do**

Draw $z \sim \phi(x^{(k)})$.

Set,

$$x_{k+1} = \begin{cases} z & \text{, with probability } \min\left(1,\, \frac{\pi(z)}{\pi(x^{(k)})}\right), \\ x^{(k)} & \text{, otherwise.} \end{cases}$$

end for

Return $\left\{x^{(k)}\right\}_{k=0}^{v}$.

end function

3 Proposed Methods

In this section, we propose two matrix-free EnKF formulations, whose analysis steps support on MCMC methods. The first of them makes use of the traditional Random-Walk (2.7) while the last one uses the Crank-Nicholson proposal function (2.6). The target distribution in our framework is given by the posterior distribution (1.4). The precision covariance of the background error distribution $\mathbf{B}^{-1}$ is estimated via the modified Cholesky decomposition (2.3). The methods are derived on common assumptions done in the context of ensemble DA: ensemble sizes are much lesser than model dimensions ($N \ll n$), and n oscillates in the order of millions. As in any sequential DA scheme, we start with an initial ensemble $\mathbf{X}^b \in \mathbb{R}^{n\times N}$ of model states. The ensemble mean is computed $\overline{\mathbf{x}}^b$ and an iterative process is performed. Starting with $\mathbf{x}^{(0)} = \overline{\mathbf{x}}^b$, the idea is to generate a sequence of v states $\mathbf{x}^{(k)}$, for $1 \leq k \leq v$, in such manner that $\mathbf{x}^a \approx \overline{\mathbf{x}}^{(a)} = \mathbf{x}^{(v)}$, this is, the last state in the chain serves as an approximation of (1.6) which in turn is the analysis mean. The posterior ensemble can be built about $\overline{\mathbf{x}}^a$ as is done in the P-EnKF context (2.5). At this point, of course, the observation operator is linearized about $\overline{\mathbf{x}}^a$.

3.1 A Random-Walk approach: the ensemble Kalman filter based on MCMC (EnKF-RW)

Intuitively, in MCMC methods, by using the precision covariance (2.3), at iteration k, for $1 \leq k \leq v$, the proposal distribution reads:

$$\mathbf{z} \sim \mathcal{N}\left(\mathbf{x}^{(k)}, \left[\widehat{\mathbf{B}}^{-1}\right]^{-1}\right),$$

or simply,

$$\mathbf{z} = \mathbf{x}^{(k)} + \boldsymbol{\nu} \in \mathbb{R}^{n\times 1}, \tag{3.1}$$

where the vector $\boldsymbol{\nu} \in \mathbb{R}^{n\times 1}$ can be computed via the solution of the following linear system of equations:

$$\left[\mathbf{L}^T \cdot \mathbf{D}^{1/2}\right] \cdot \boldsymbol{\nu} = \boldsymbol{\xi}\,,$$

where, recall, $\boldsymbol{\xi} \in \mathbb{R}^{n\times 1}$ is standard Normally distributed. However, by the Central Limit Theorem, we know that proposed states in the form (3.1) will lie in a small shell about $\mathbf{x}^{(k)}$. Consequently, samples can be proposed far from high-probability zones of the target distribution (1.4) which can impact the speed of convergence in MCMC methods. To overcome this situation, we consider a first order Taylor polynomial about the current state $\mathbf{x}^{(k)}$,

$$\mathcal{H}\left(\mathbf{x}\right) \approx \mathcal{G}_k(\mathbf{x}) = \mathcal{H}\left(\mathbf{x}^{(k)}\right) + \mathbf{H}_{\mathbf{x}^{(k)}} \cdot \left[\mathbf{x} - \mathbf{x}^{(k)}\right]\,,$$

where $\mathbf{H}_{\mathbf{x}^{(k)}} \in \mathbb{R}^{m\times n}$ is the Jacobian matrix of $\mathcal{H}(\mathbf{x})$ at $\mathbf{x}^{(k)}$. Hence, a convex approximation of the 3D-Var cost function (1.5), about $\mathbf{x}^{(k)}$, is as follows:

$$\widehat{\mathcal{J}}(\mathbf{x}) = \frac{1}{2}\cdot\left\|\mathbf{x}-\mathbf{x}^b\right\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\cdot\left\|\mathbf{y}-\mathcal{G}_k\left(\mathbf{x}\right)\right\|^2_{\mathbf{R}^{-1}}\,, \tag{3.2a}$$

whose gradient reads,

$$\nabla\widehat{\mathcal{J}}(\mathbf{x}) = \widehat{\mathbf{B}}^{-1}\cdot\left[\mathbf{x}-\mathbf{x}^b\right] - \mathbf{H}^T_{\mathbf{x}^{(k)}}\cdot\mathbf{R}^{-1}\cdot\left[\mathbf{y}-\mathcal{G}_k(\mathbf{x})\right] \in \mathbb{R}^{n\times 1}. \tag{3.2b}$$

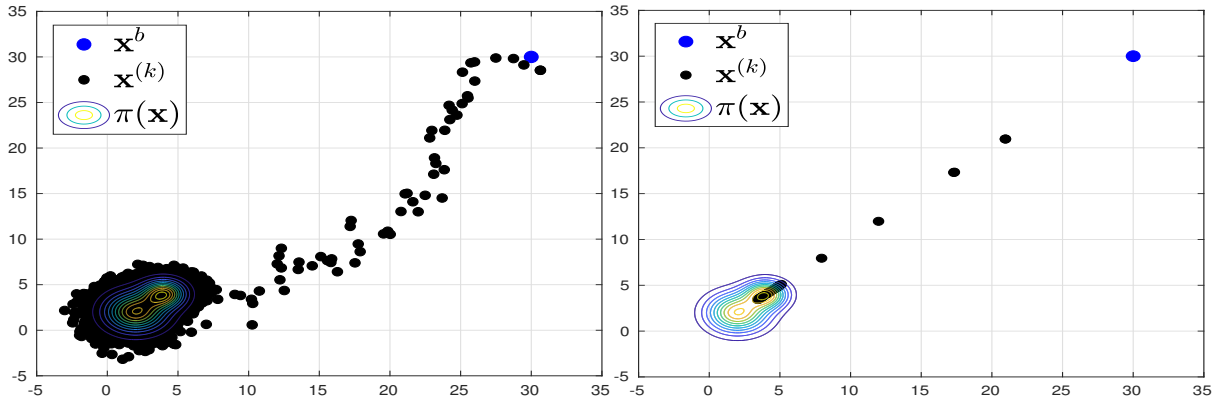
Then, the proposal distribution (3.1) is replaced by an Uniform distribution wherein samples along the direction $-\nabla\widehat{\mathcal{J}}(\mathbf{x}^{(k)})$ have the same probability of occurrence, such samples can be generated as follows:

$$\mathbf{z} = \mathbf{x}^{(k)} + \lambda\cdot\left[-\frac{\nabla\widehat{\mathcal{J}}\left(\mathbf{x}^{(k)}\right)}{\left\|\nabla\widehat{\mathcal{J}}\left(\mathbf{x}^{(k)}\right)\right\|}\right]\,,\ \text{with}\ \lambda\sim\mathcal{U}\left(0,\,\beta\right)\,, \tag{3.3}$$

where $\mathcal{U}\left(0,\,\beta\right)$ stands for an Uniform distribution on the interval $(0,\,\beta)$. An intuitive range for β can be,

$$\beta \in \left[1,\,\left\|\nabla\widehat{\mathcal{J}}\left(\mathbf{x}^{(k)}\right)\right\|\right]\,,$$

where, $\left\|\nabla\widehat{\mathcal{J}}\left(\mathbf{x}^{(k)}\right)\right\|$ can be chosen for (nearly) linear operators while its counterpart for highly non-linear ones. The main motivation for using this range is that, since $\mathcal{H}(\mathbf{x})$ can be highly non-linear (and even more, non-smooth), (3.2a) can be a good approximation of (1.5) only in a small region about $\mathbf{x}^{(k)}$. In the abscense of prior information about β, one can choose 1.
For the two dimensional target distribution $\pi(\mathbf{x}) = \exp(-1/2\cdot\left\|\mathbf{x}-[2,\,2]^T\right\|^2/2) + \exp(-1/2\cdot\left\|\mathbf{x}-[4,\,4]^T\right\|^2)$, the MCMC chains are shown in figure 1b for the proposal Random Walk (3.1) and the gradient driven one (3.3). The computational benefits are just evident.



(a) MCMC Random Walk (b) MCMC Gradient Driven.

Figure 1: Path followed by traditional MCMC methods vs that based on gradient approximations. Gradient approximations can be exploited in order to speed-up the convergence of sampling methods towards high-probability zones of target distributions.

Algorithm 2 Analysis computation via MCMC with Descent Walk.

1: **function** COMPUTE_ANALYSIS_DW($\overline{\mathbf{x}}^b$, v, $\nabla\widehat{\mathcal{J}}(\mathbf{x})$, $\mathcal{J}(\mathbf{x})$)

Require: The background ensemble mean $\overline{\mathbf{x}}^b$, the maximum number of iterations v, the cost function $\mathcal{J}(\mathbf{x})$, and the gradient approximation $\nabla\widehat{\mathcal{J}}(\mathbf{x})$ of $\nabla\mathcal{J}(\mathbf{x})$.

Ensure: An approximation $\overline{\mathbf{x}}^a$ of the analysis state $\mathbf{x}^a$.

2: Set $\mathbf{x}^{(0)} \leftarrow \overline{\mathbf{x}}^b$

3: **for** $k = 0$ to v **do**

4: Draw $\lambda \sim \mathcal{U}(0,\, \beta)$

5: Set $\mathbf{z} \leftarrow \mathbf{x}^{(k)} + \lambda \cdot \left[-\frac{\nabla\widehat{\mathcal{J}}(\mathbf{x}^{(k)})}{\|\nabla\widehat{\mathcal{J}}(\mathbf{x}^{(k)})\|}\right]$

6: Compute $\delta \leftarrow \min\left(1,\, \mathcal{J}\left(\mathbf{x}^{(k)}\right)/\mathcal{J}\left(\mathbf{z}\right)\right)$

7: Draw $u \sim \mathcal{U}(0,\, 1)$

8: **if** $u < \delta$ **then**

9: Set $\mathbf{x}^{(k+1)} \leftarrow \mathbf{z}$

10: **end if**

11: **end for**

12: **return** $\mathbf{x}^{(v)}$ as $\overline{\mathbf{x}}^a$

13: **end function**

Notice, samples from distribution (3.3) can be generated by (3.1) as well (even if that is done with low probability) but, the subset of samples along the search direction (3.3) provide states which potentially can maximize the posterior probability. On the other hand, no matrix inversion is required in this context in order to propose states. The acceptance and rejection criteria of those rely on the Metropolis-Hastings criterion. The overall sampling procedure is detailed in the Algorithm 2. Note that, in line 6, a Padé approximant is utilized in order to estimate the ratio

$$\frac{\mathcal{P}\left(\mathbf{x}^{(k)}|\mathbf{y}\right)}{\mathcal{P}\left(\mathbf{z}|\mathbf{y}\right)} \propto \frac{\exp\left(-\mathcal{J}(\mathbf{x}^{(k)})\right)}{\exp\left(-\mathcal{J}(\mathbf{z})\right)} \approx \frac{1-\mathcal{J}(\mathbf{x}^{(k)})}{1-\mathcal{J}(\mathbf{z})} \approx \frac{\mathcal{J}(\mathbf{x}^{(k)})}{\mathcal{J}(\mathbf{z})}\,,$$

since in the DA context $\mathcal{J}(\mathbf{z}), \mathcal{J}(\mathbf{x}^{(k)}) \gg 1$. This approximation is done owing to computational aspects: exponential functions can grow/decay very rapidly in such manner that, having $\mathcal{J}(\mathbf{x}^{(k)})$ and $\mathcal{J}(\mathbf{z})$ as exponents can cause memory under/overflows and consequently, the standard ratio cannot be computed. Unlike MCMC methods based on Random-Walk, this procedure does not return a chain but only the last state in that. This mainly obeys to the fact that, gradient approximations of (1.5) are utilized among iterations and therefore, we expect final iterates to be stationary points of (1.6). On the other hand, the most likely state from the chain in the Algorithm 2 can be returned as well but, this choice is sensitive to overfitting, for instance, consider the linear case $\mathbf{H} = \mathbf{I}$ and $\mathbf{R} = \sigma_o^2 \cdot \mathbf{I}$,

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^b + \left[\widehat{\mathbf{B}}^{-1} + \sigma_o^2 \cdot \mathbf{I}\right]^{-1} \cdot \left[\mathbf{y} - \overline{\mathbf{x}}^b\right] , \tag{3.4}$$

and the Singular Value Decomposition of $\widehat{\mathbf{B}}^{-1} = \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{U}^T$ where $\mathbf{U} \in \mathbb{R}^{n \times n}$ is a matrix holding the left (right) singular vectors of $\widehat{\mathbf{B}}^{-1}$ whose corresponding singular values are stored in the diagonal matrix $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$, typically, in decreasing order. By using this decomposition, (3.4) can be written as follows:

$$\begin{aligned}
\overline{\mathbf{x}}^a &= \overline{\mathbf{x}}^b + \left[\mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{U}^T + \sigma_o^2 \cdot \mathbf{I}\right]^{-1} \cdot \left[\mathbf{y} - \overline{\mathbf{x}}^b\right] \\
&= \overline{\mathbf{x}}^b + \mathbf{U} \cdot \left[\mathbf{\Sigma} + \sigma_o^2 \cdot \mathbf{I}\right]^{-1} \cdot \mathbf{U}^T \cdot \left[\mathbf{y} - \overline{\mathbf{x}}^b\right] \\
&= \overline{\mathbf{x}}^b + \sum_{i=1}^{n} \left(\frac{1}{\sigma^{[i]} + \sigma_o^2}\right) \cdot \mathbf{u}^{[i]} \cdot \mathbf{u}^{[i]^T} \cdot \left[\mathbf{y} - \overline{\mathbf{x}}^b\right] ,
\end{aligned}$$

where $\sigma^{[i]}$ denotes the i-th singular value of $\widehat{\mathbf{B}}^{-1}$, for $1 \leq i \leq n$. As is well known, observations are composed by two sources of information: $\mathbf{y} = \mathbf{x}^* + \widetilde{\mathbf{y}}$, where $\widetilde{\mathbf{y}} \sim \mathcal{N}\left(\mathbf{0}, \widetilde{\mathbf{R}}\right)$, and $\widetilde{\mathbf{R}} \in \mathbb{R}^{m \times m}$ is the actual data error covariance matrix. Recall that, $\mathbf{x}^*$ is the actual state of the system. Thus, the likelihood function can be maximized even though $\sigma^{[i]} + \sigma_o^2 < \mathbf{u}^{[i]^T} \cdot \widetilde{\mathbf{y}}$, in such cases, noisy information is injected into the analysis mean $\overline{\mathbf{x}}^a$ and even worst, it can be amplified for small (near zero) singular values. Thus, checking for most likely states does not guaranty to obtain a better approximation of $\mathbf{x}^*$.

3.2 The Crank-Nicolson proposal: the ensemble Kalman filter based on Crank-Nicholson (EnKF-CN)

Since matrix dimensions in the context of DA range according to model resolutions, matrix computations must be performed as efficiently as possible. Efficient manners to generate samples are a must in this context, we propose a novel and easy manner to draw samples from a Crank-Nicholson (CN) proposal distribution. Consider the CN like proposal (2.6),

$$\left[\mathbf{I} + \frac{1}{2} \cdot \mathbf{T} \cdot \widehat{\mathbf{B}}^{-1}\right] \cdot \mathbf{z} = \left[\mathbf{I} - \frac{1}{2} \cdot \mathbf{T} \cdot \widehat{\mathbf{B}}^{-1}\right] \cdot \mathbf{x}^{(k)} + [2 \cdot \mathbf{T}]^{1/2} \cdot \boldsymbol{\xi} \tag{3.5}$$

where $\mathbf{B}^{-1}$ has been estimated via the modified Cholesky decomposition $\widehat{\mathbf{B}}^{-1}$. Consider $\mathbf{T} = \gamma \cdot \mathbf{I}$, where $\gamma > 0$ is a parameter to be chosen later, and by multiplying both sides by 2, we

obtain:

$$\mathbf{z} \;=\; \mathbf{q}^{(k)} + 2\cdot\sqrt{2\cdot\gamma}\cdot\left[2\cdot\mathbf{I}+\gamma\cdot\widehat{\mathbf{B}}^{-1}\right]^{-1}\cdot\boldsymbol{\xi}\,, \tag{3.6a}$$

where $\mathbf{q}^{(k)}\in\mathbb{R}^{n\times 1}$ is obtained via the solution of the next linear system:

$$\left[2\cdot\mathbf{I}+\gamma\cdot\widehat{\mathbf{B}}^{-1}\right]\cdot\mathbf{q}^{(k)} = \left[2\cdot\mathbf{I}-\gamma\cdot\widehat{\mathbf{B}}^{-1}\right]\cdot\mathbf{x}^{(k)}\,. \tag{3.6b}$$

Notice, samples proposed by (3.6a) can be characterized via the distribution:

$$\mathbf{z}\sim\mathcal{N}\left(\mathbf{q}^{(k)},\,8\cdot\gamma\cdot\left[2\cdot\mathbf{I}+\gamma\cdot\widehat{\mathbf{B}}^{-1}\right]^{-2}\right)\,. \tag{3.6c}$$

Following a similar reasoning to that of the EnKF-RW, a descent direction of (1.5) at $\mathbf{q}^{(k)}$ can be approximated as follows:

$$\nabla\widetilde{\mathcal{J}}(\mathbf{x}) = \widehat{\mathbf{B}}^{-1}\cdot\left[\mathbf{x}-\mathbf{x}^{b}\right] - \mathbf{H}^{T}_{\mathbf{q}^{(k)}}\cdot\mathbf{R}^{-1}\cdot\left[\mathbf{y}-\mathcal{H}\left(\mathbf{q}^{(k)}\right)\right]\;\in\mathbb{R}^{n\times 1}. \tag{3.7}$$

The CN proposal (3.6) is replaced in such manner that, samples along the direction (3.7) are proposed with equal probability, such samples can be generated as follows:

$$\mathbf{z} = \mathbf{q}^{(k)} + \lambda\cdot\left[-\frac{\nabla\widetilde{\mathcal{J}}\left(\mathbf{q}^{(k)}\right)}{\left\|\nabla\widetilde{\mathcal{J}}\left(\mathbf{q}^{(k)}\right)\right\|}\right]\;,\;\text{with}\;\lambda\sim\mathcal{U}\left(0,\,\beta\right)\;. \tag{3.8}$$

Since large matrix dimensions are common in this context, the direct solution of linear system (3.6b) is prohibitive, let us write such linear system in the form,

$$\left[2\cdot\mathbf{I}+\gamma\cdot\widehat{\mathbf{B}}^{-1}\right]\cdot\mathbf{q}^{(k)} = \mathbf{w}^{(k)}\,, \tag{3.9}$$

where $\mathbf{w}^{(k)} = \left[2\cdot\mathbf{I}-\gamma\cdot\widehat{\mathbf{B}}^{-1}\right]\cdot\mathbf{x}^{(k)}\in\mathbb{R}^{n\times 1}$. Equation (3.9) can be written as follows,

$$2\cdot\mathbf{q}^{(k)} + \gamma\cdot\widehat{\mathbf{B}}^{-1}\cdot\mathbf{q}^{(k)} = \mathbf{w}^{(k)}\,, \tag{3.10}$$

which yields to the iterative method,

$$2\cdot\mathbf{q}^{(k,\,p+1)} = \mathbf{w}^{(k)} - \gamma\cdot\widehat{\mathbf{B}}^{-1}\cdot\mathbf{q}^{(k,\,p)}\,,$$

or equivalently,

$$\mathbf{q}^{(k,\,p+1)} = \frac{1}{2}\cdot\left[\mathbf{w}^{(k)} - \gamma\cdot\widehat{\mathbf{B}}^{-1}\cdot\mathbf{q}^{(k,\,p)}\right]\,. \tag{3.11}$$

To ensure convergence of the iteration (3.11), we need the γ value to satisfy (Nino-Ruiz, 2020, Theorem 1):

$$\left\|\gamma\cdot\widehat{\mathbf{B}}^{-1}\right\| < 2\,, \tag{3.12}$$

which can be guaranteed, for instance, by either,

$$\gamma = \left[\sum_{i=1}^{n} \left(\left| \left\{ \widehat{\mathbf{B}}^{-1} \right\}_{i,i} \right| + \sum_{j \in P(i,r)} \left| \left\{ \widehat{\mathbf{B}}^{-1} \right\}_{i,j} \right| \right) \right]^{-1}, \tag{3.13a}$$

or

$$\gamma = \left[\max_{1 \leq i,j \leq n} \left| n^2 \cdot \left\{ \widehat{\mathbf{B}}^{-1} \right\}_{ij} \right| \right]^{-1}. \tag{3.13b}$$

The value (3.13b) can be easily obtained by noting that $\widehat{\mathbf{B}}^{-1}$ is a covariance matrix and therefore, per row, the largest value is on the diagonal. This allows us to write (3.13b) as follows:

$$\gamma = \left[\max_{1 \leq i \leq n} \left| n^2 \cdot \left\{ \widehat{\mathbf{B}}^{-1} \right\}_{ii} \right| \right]^{-1},$$

or, by the definition of $\widehat{\mathbf{B}}^{-1}$ (2.3),

$$\gamma = \left[\max_{1 \leq i \leq n} \left(n^2 \cdot \sum_{j \in P(i,r)} \{\mathbf{L}\}_{i,j}^2 \cdot \{\mathbf{D}\}_{ii} \right) \right]^{-1},$$

which is a computational friendly expression of (3.13b). Besides, the necessary number of iterations q to approximate the solution (3.9) via the iteration (3.11) with a precision η is given by (Nino-Ruiz, 2020, Theorem 2),

$$q \geq \frac{\log\left(\frac{2 \cdot \epsilon}{\left\| \mathbf{w}^{(k)} \right\|} \right)}{\log\left(\frac{\left\| \gamma \cdot \widehat{\mathbf{B}}^{-1} \right\|}{2} \right)}.$$

Putting it all together, a posterior mode (1.6) can be estimated based on the ensemble (2.1a) via the Crank-Nicholson proposal as is shown in the Algorithm 3.

Algorithm 3 Analysis computation via MCMC with CN proposal.

function COMPUTE_ANALYSIS_CN($\overline{\mathbf{x}}^b$, v, $\widehat{\mathbf{B}}^{-1}$, $\nabla\widehat{\mathcal{J}}\,(\mathbf{x})$, $\mathcal{J}\,(\mathbf{x})$, η)

Require: The background ensemble mean $\overline{\mathbf{x}}^b$, the number of iterations v, an estimate $\widehat{\mathbf{B}}^{-1}$ of the precision matrix $\mathbf{B}^{-1}$ via a modified Cholesky decomposition, the cost function $\mathcal{J}\,(\mathbf{x})$, the gradient approximation $\nabla\widehat{\mathcal{J}}\,(\mathbf{x})$ of $\nabla\mathcal{J}(\mathbf{x})$, and the precision η.

Ensure: An approximation $\overline{\mathbf{x}}^a$ of the analysis state $\mathbf{x}^a$.

Choose γ according to (3.13).
Set $\mathbf{x}^{(0)} \leftarrow \overline{\mathbf{x}}^b$.
for $k = 1$ to v **do**
  Set $\mathbf{w}^{(k)} \leftarrow \left[2 \cdot \mathbf{I} - \gamma \cdot \widehat{\mathbf{B}}^{-1}\right] \cdot \mathbf{x}^{(k)}$.
  Set $q \leftarrow \log\left(\frac{2\cdot\eta}{\|\mathbf{w}^{(k)}\|}\right) / \log\left(\frac{\|\gamma\cdot\widehat{\mathbf{B}}^{-1}\|}{2}\right)$
  for $p = 0$ to q **do**
    Set $\mathbf{q}^{(k,\,p+1)} \leftarrow \frac{1}{2} \cdot \left[\mathbf{w}^{(k)} - \gamma \cdot \widehat{\mathbf{B}}^{-1} \cdot \mathbf{q}^{(k,\,p)}\right]$
  end for
  Set $\mathbf{q}^{(k)} \leftarrow \mathbf{q}^{(k,\,v)}$
  Draw $\lambda \sim \mathcal{N}\,(0,\,\beta)$
  Set $\mathbf{z} \leftarrow \mathbf{q}^{(k)} + \lambda \cdot \left[-\frac{\nabla\widetilde{\mathcal{J}}\left(\mathbf{x}^{(k)}\right)}{\|\nabla\widetilde{\mathcal{J}}\left(\mathbf{x}^{(k)}\right)\|}\right]$
  Compute $\delta \leftarrow \min\left(1,\,\mathcal{J}\left(\mathbf{x}^{(k)}\right)/\mathcal{J}\,(\mathbf{z})\right)$
  Draw $u \sim \mathcal{U}(0,\,1)$
  if $u < \delta$ **then**
    Set $\mathbf{x}^{(k+1)} \leftarrow \mathbf{z}$
  end if
end for
return $\mathbf{x}^{(v)}$ as $\overline{\mathbf{x}}^a$

end function

3.3 Proposed matrix-free sampling filter

Once a posterior mode of the error distribution is obtained by either, the Crank-Nicholson proposal or the traditional Random-Walk, the posterior ensemble is built about it. The Algorithm 4 details a complete assimilation cycle of the proposed methods. We denote by EnKF-RW the EnKF implementation based on the Random-Walk (3.3) while EnKF-CN stands for that based on the Crank-Nicholson proposal (3.8).

Algorithm 4 Analysis computation via MCMC with CN proposal.

1: **function** PERFORM_ASSIMILATION($\mathbf{X}^b$, $\mathbf{y}$, v, $\nabla\widehat{\mathcal{J}}\,(\mathbf{x})$, $\mathcal{J}\,(\mathbf{x})$, ϵ)

Require: The background ensemble $\mathbf{X}^b$, the observation $\mathbf{y}$, the number of iterations v for the sampling methods, the cost function $\mathcal{J}\,(\mathbf{x})$, the gradient approximation $\nabla\widehat{\mathcal{J}}\,(\mathbf{x})$ of $\nabla\mathcal{J}(\mathbf{x})$, and the precision ϵ.

Ensure: Approximated samples $\mathbf{X}^a$ of the posterior distribution (1.4).

2: Compute the ensemble mean $\overline{\mathbf{x}}^b$ according to (2.1b).

3: Estimate $\widehat{\mathbf{B}}^{-1}$ via the modified Cholesky decomposition (2.3).

4: Set $\overline{\mathbf{x}}^a \leftarrow$ COMPUTE_ANALYSIS_CN($\overline{\mathbf{x}}^b, v, \widehat{\mathbf{B}}^{-1}, \nabla\widehat{\mathcal{J}}\,(\mathbf{x})\,, \mathcal{J}\,(\mathbf{x})\,, \epsilon$). ▷ (EnKF-CN)

5: Set $\overline{\mathbf{x}}^a \leftarrow$ COMPUTE_ANALYSIS_RW($\overline{\mathbf{x}}^b, v, \nabla\widehat{\mathcal{J}}\,(\mathbf{x})\,, \mathcal{J}\,(\mathbf{x})$). ▷ (EnKF-RW)

6: Linearize $\mathcal{H}(\mathbf{x})$ about $\overline{\mathbf{x}}^a$.

7: Compute the posterior ensemble $\mathbf{X}^a$ via (2.5).

8: **return** $\mathbf{X}^a$

9: **end function**

4 Experimental Results

In this section, experimental tests are performed in order to assess the accuracy of the proposed EnKF implementations: the EnKF-RW and the EnKF-CN. We make use of the Lorenz-96 model (Lorenz, 2005) as our surrogate model during the experiments. The Lorenz-96 model is described by the following set of ordinary differential equations (Fertig, Harlim and Hunt, 2007):

$$\frac{dx_j}{dt} = \begin{cases} (x_2 - x_{n-1}) \cdot x_n - x_1 + F & \text{for } j = 1, \\ (x_{j+1} - x_{j-2}) \cdot x_{j-1} - x_j + F & \text{for } 2 \leq j \leq n-1, \\ (x_1 - x_{n-2}) \cdot x_{n-1} - x_n + F & \text{for } j = n, \end{cases} \tag{4.1}$$

where F is external force and $n = 40$ is the number of model components. Periodic boundary conditions are assumed. When $F = 8$ units the model exhibits chaotic behavior, which makes it a relevant surrogate problem for atmospheric dynamics (Karimi and Paul, 2010; Gottwald and Melbourne, 2005). A time unit in the Lorenz-96 represents 7 days in the atmosphere. The experimental settings are described below:

- We obtain an initial condition $\widetilde{\mathbf{x}}^b_{-2}$ after a long-time numerical integration of a random vector state. As should be expected, the solution $\widetilde{\mathbf{x}}^b_{-2}$ is consistent with the model dynamics (4.1).

- We sample a perturbed state from the following distribution

$$\widetilde{\mathbf{x}}^b_{-2} \sim \mathcal{N}\left(\mathbf{x}^*_{-2},\, 0.05^2 \cdot \mathbf{I}\right)\,,$$

  we numerically integrate this state for a long-time period, after which, the initial background solution $\mathbf{x}^b_{-1}$ is obtained.

- Following the previous steps, we sample from the distribution,

$$\widetilde{\mathbf{x}}_{-1}^{b[\widehat{e}]} \sim \mathcal{N}\left(\mathbf{x}_{-1}^{b},\, 0.05^2 \cdot \mathbf{I}\right)\,,\ \text{for}\ 1 \leq \widehat{e} \leq \widehat{N}\,,$$

to obtain samples that become initial ensemble members after a propagation time of 10 time units. We perform this step several times to build an initial pool of ensemble members $\widehat{\mathbf{X}^b}_0$ whose dimension reads $\widehat{N} = 10^4$ members. A two dimensional projection of this pool is shown in figure 2a.

- We build an assimilation window of $M = 20$ time-spaced observations. Observations are taken every 3.5 days, and we describe their error statistics via the following Gaussian distribution,

$$\mathbf{y}_k \sim \mathcal{N}\left(\mathcal{H}_k\left(\mathbf{x}_k^*\right),[\epsilon^o]^2 \cdot \mathbf{I}\right)\,,\ \text{for}\ 1 \leq k \leq M\,,$$

where the standard deviations of observational errors $\epsilon^o = 10^{-2}$ (as is typicall, in practice). The components are randomly chosen at the different assimilation cycles. We use two observation operators from the literature, a non-linear and non-smooth operator (van Leeuwen, Cheng and Reich, 2015):

$$\left\{\mathcal{H}\left(\mathbf{x}\right)\right\}_j = \frac{\left\{\mathbf{x}\right\}_j}{2} \cdot \left[\left(\frac{\left|\left\{\mathbf{x}\right\}_j\right|}{2}\right)^{\alpha-1} + 1\right]\,, \tag{4.2}$$

and a very challenging one, an exponential operator (Attia and Sandu, 2015):

$$\left\{\mathcal{H}\left(\mathbf{x}\right)\right\}_j = \exp\left(\left\{\mathbf{x}\right\}_j\right)\,, \tag{4.3}$$

where j denotes the j-th observed component from the model state.

- The radius of influence is set to $r = 1$ while the inflation factor is set to 1.02 (a typical value).

- The ensemble size for the benchmarks is $N = 20$. These members are randomly chosen from the pool $\widehat{\mathbf{X}^b}_0$ for the different experiments in order to form the initial ensemble $\mathbf{X}_0^b$ for the assimilation window. Evidently, $\mathbf{X}_0^b \subset \widehat{\mathbf{X}^b}_0$.

- The ℓ_2 norm of errors are utilized as a measure of accuracy at the assimilation step k,

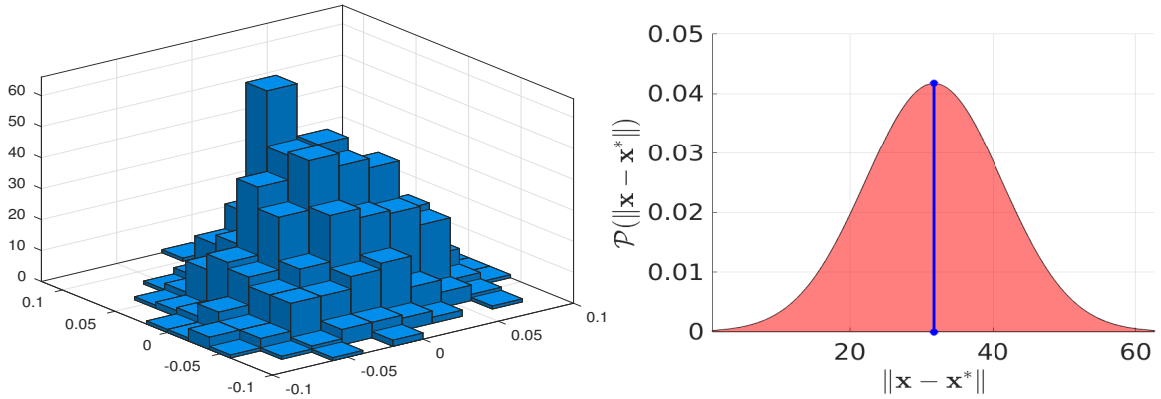
$$\epsilon_k = \sqrt{\left[\mathbf{x}_k^* - \mathbf{x}_k^a\right]^T \cdot \left[\mathbf{x}_k^* - \mathbf{x}_k^a\right]}\,, \tag{4.4}$$

where $\mathbf{x}_k^*$ and $\mathbf{x}_k^a$ are the reference and the analysis solutions, respectively. In figure 2b, the ℓ_2 norm error distribution is shown for the initial background. The estimated error, in average, reads $\epsilon^b \approx 31.73$ while its standard deviation equals $\sigma_{\epsilon^b} \approx 3.09$. By convenience, these values are expressed in the log scale: $\log(\epsilon^b) = 3.45$, and $\log(\sigma_{\epsilon^b}) = 1.13$.

- We employ the Root-Mean-Square-Error (RMSE) as a measure of performance for a

given assimilation window,

$$\epsilon = \sqrt{\frac{1}{M} \cdot \sum_{k=1}^{M} \epsilon_k^2}\,. \tag{4.5}$$



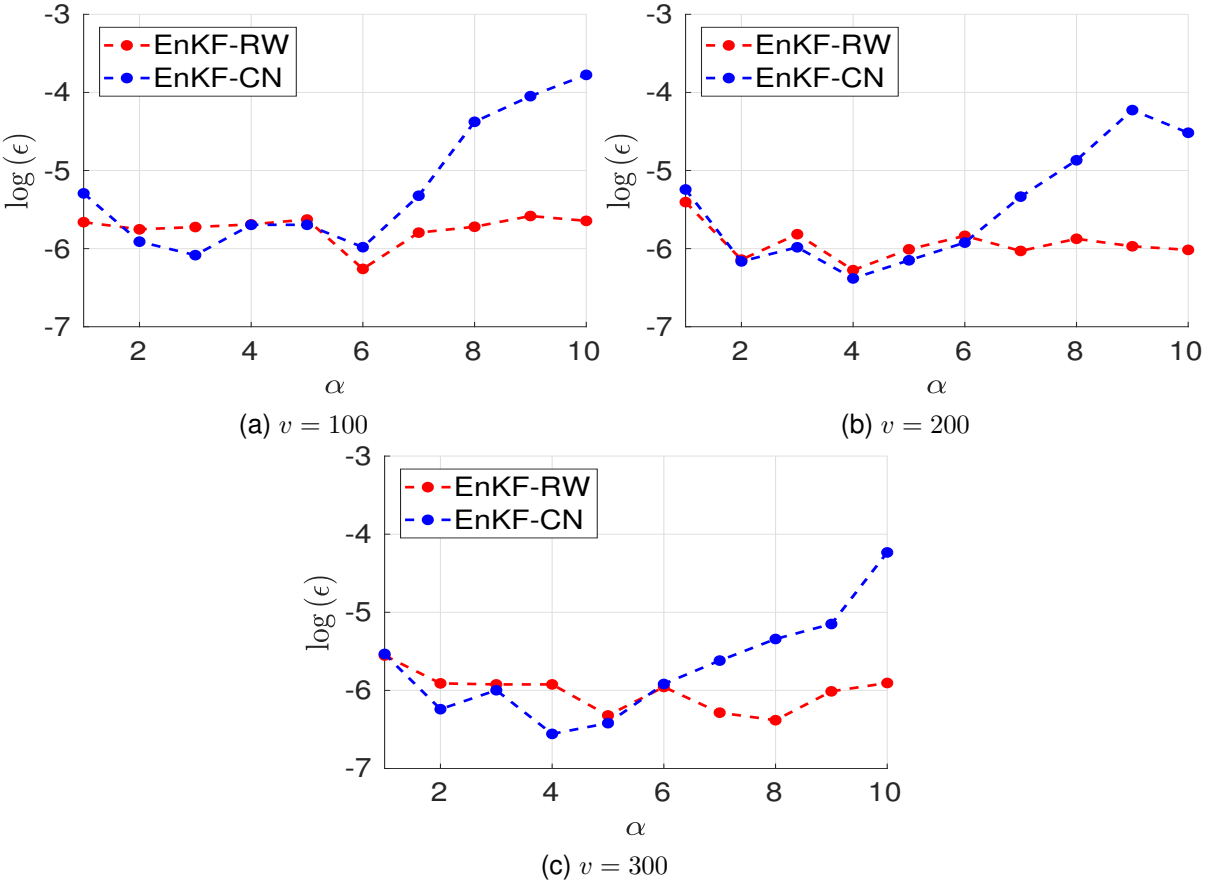
(a) 2D projection of $\mathbf{X}^b$

(b) ℓ_2 norm error distribution

Figure 2: 2D projection of the background ensemble (pool of $N = 10^4$ members) as well as its estimated ℓ_2 norm error distribution. In average, the ℓ_2 norm of errors are 31.73 with standard deviation of 3.09.

4.1 Single Assimilation Step - Full Observational Network

In this part, we make use of the non-linear observation operator (4.2) with a full observational network for a single assimilation step. The values of α are varied from 1 (linear operator) to 10 (highly non-linear operator). We set the number of iterations to $v = 100$, 200, and 300. The experiments, for each configuration $(\alpha,\, v)$, are run 10 times in order to assess, in average, the errors. The logarithms of such errors are shown in figure 6. As can be seen, the EnKF-RW is lesser sensitive to increments in α than the EnKF-CN implementation. This can be explained as follows: as the values of α increases, the high-probability zones of the posterior distribution become sharper, the proposed states in the EnKF-CN context are based on candidates of a shifted state and therefore, since their acceptances are probabilistic, candidates with a lower posterior probability can be accepted along steepest descent approximations of $\mathcal{J}(\mathbf{x})$ when these are near to current solutions. Nevertheless, as the number of iterations is increased, the impact of increments in α is mitigated. On the other hand, the EnKF-RW formulation does not suffer from such condition, recall that, in this method candidates are compared against the current solution. For all cases, the estimated posterior errors are lesser than those of the prior $\log(\epsilon^b) \approx 3.45$ and the observations $\log(\epsilon^o) \approx -2$ as should be expected for full observational networks. In the Table 1, we report the standard deviations of errors across different runs as well. Note that, the uncertainty of errors in all cases is lesser than that of the prior distribution $\log(\sigma_{\epsilon^b}) \approx 1.13$, and in some cases, by several order of magnitudes.



(a) $v = 100$ (b) $v = 200$

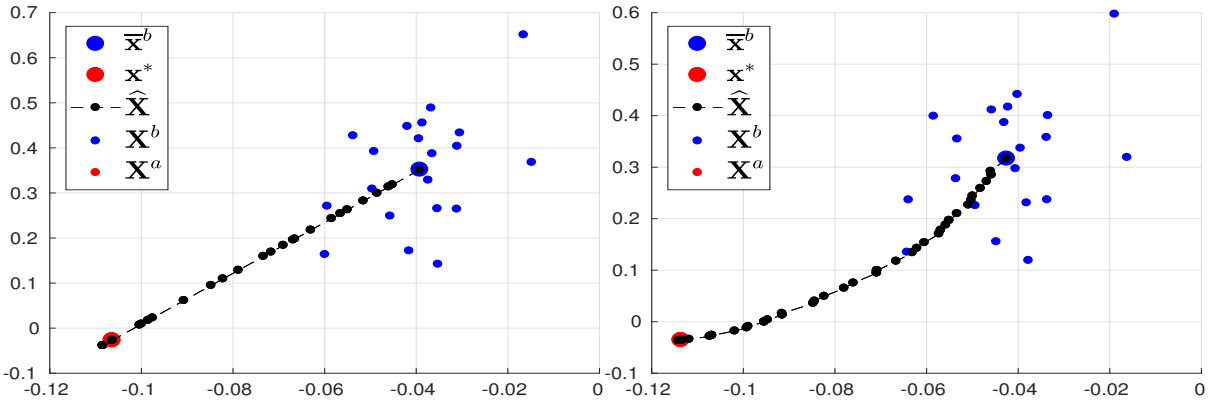
(c) $v = 300$

Figure 3: Experimental results with the Lorenz-96 model (4.1) and the non-smooth observation operator (4.2). The average of the ℓ_2 error norms across 10 iterations are shown for different values of v and α.

Figure 4 shows some snapshots of the initial ensemble, the ensemble mean (background), the actual state of the system, the posterior ensemble, and the path followed by the proposed methods starting from the ensemble mean for different values of α and $v = 100$. The percentage of variance explained by such plots is about 90%. As can be seen, the EnKF-RW path is affected as the value of α increases, such case is not notorious in the EnKF-CN context. For instance, regardless the value of α, the EnKF-CN path seems to be "straight". The EnKF-CN method can quickly reach high-probability zones of posterior distributions but, as we noted before, it is sensitive to sharp shapes (given by the α value) in those.

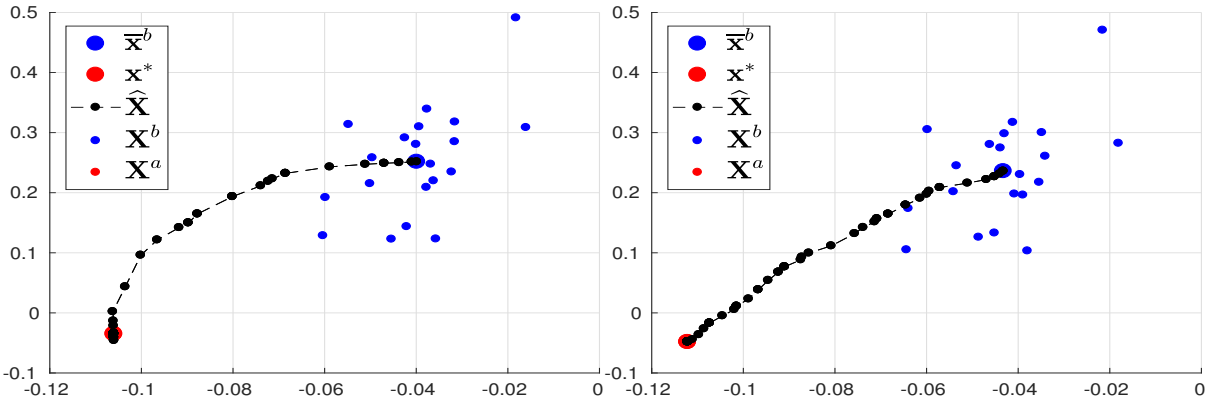
		$\log(\bar{\epsilon})$		$\log(\sigma_\epsilon)$	
v	α	EnKF-CN	EnKF-RW	EnKF-CN	EnKF-RW
100	1	-5.66	-5.29	-6.09	-4.78
	2	-5.75	-5.91	-5.76	-6.13
	3	-5.72	-6.08	-4.85	-6.47
	4	-5.69	-5.70	-6.46	-6.20
	5	-5.63	-5.70	-5.89	-6.44
	6	-6.26	-5.98	-6.38	-6.12
	7	-5.80	-5.32	-5.62	-4.60
	8	-5.72	-4.38	-5.53	-3.27
	9	-5.58	-4.05	-5.02	-3.32
	10	-5.64	-3.78	-6.04	-3.13
200	1	-5.40	-5.24	-5.32	-4.87
	2	-6.14	-6.16	-6.35	-6.07
	3	-5.82	-5.98	-5.41	-6.02
	4	-6.28	-6.38	-6.85	-7.60
	5	-6.01	-6.15	-6.12	-6.59
	6	-5.84	-5.92	-6.15	-6.56
	7	-6.03	-5.33	-6.74	-4.45
	8	-5.87	-4.87	-6.21	-3.91
	9	-5.97	-4.23	-7.00	-3.17
	10	-6.02	-4.52	-6.13	-3.61
300	1	-5.55	-5.53	-5.41	-5.49
	2	-5.91	-6.24	-6.16	-7.14
	3	-5.92	-6.00	-7.37	-9.02
	4	-5.92	-6.56	-4.91	-7.74
	5	-6.32	-6.42	-6.91	-7.28
	6	-5.96	-5.92	-6.83	-6.45
	7	-6.29	-5.62	-5.77	-4.56
	8	-6.38	-5.34	-8.37	-4.44
	9	-6.01	-5.15	-8.23	-4.36
	10	-5.91	-4.23	-6.05	-3.33

Table 1: Experimental results with the Lorenz-96 model (4.1). The average of the ℓ_2 error norms as well as their standard deviations across 10 runs are shown for different values of v and α.



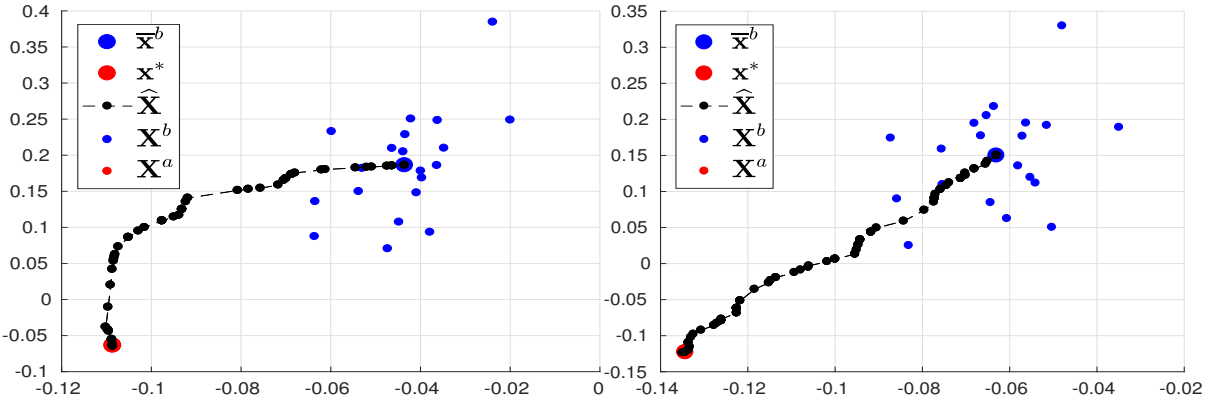
(a) EnKF-RW and $\alpha = 1$

(b) EnKF-CN and $\alpha = 1$



(c) EnKF-RW and $\alpha = 5$

(d) EnKF-CN and $\alpha = 5$



(e) EnKF-RW and $\alpha = 10$

(f) EnKF-CN and $\alpha = 10$

Figure 4: Path followed by the EnKF-RW and the EnKF-CN formulations for a single assimilation step and the non-smooth observation operator (4.2) for different values of α.

4.1.1 Exponential Operator

Following the experimental settings of the previous section, we report in figure 5 the results for the exponential observation operator (4.3). As can be seen, despite this operator is highly non-linear, it is smooth, and therefore, the EnKF-CN estimate posterior moments better than those of the EnKF-RW. Furthermore, as the number of iterations increases, errors in posterior estimates can be mitigated. Note that, in all cases, posterior errors are lesser than those of prior and observation ones. Besides, since the exponential operator is smooth, the actual gradient of (4.3) can be employed during the proposal steps, which can speed-up the convergence

of our proposed filter implementations. Figure 6 shows the path followed by the proposed implementations for some runs. Both methods exhibit similar behaviors among iterations. For instance, their convergence is not impacted by the number of iterations v. Moreover, the last members of the Markov chain are near the actual state of the system. In all cases, a good estimate of the posterior mode (1.6) is reached.

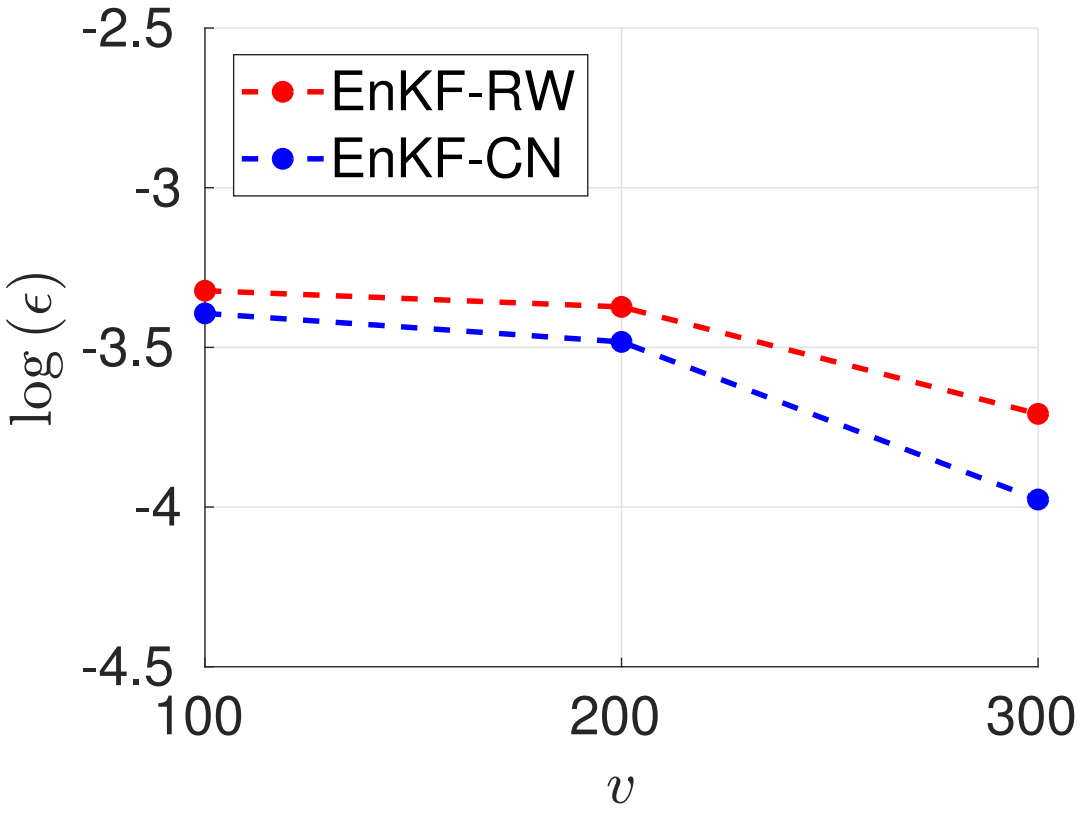
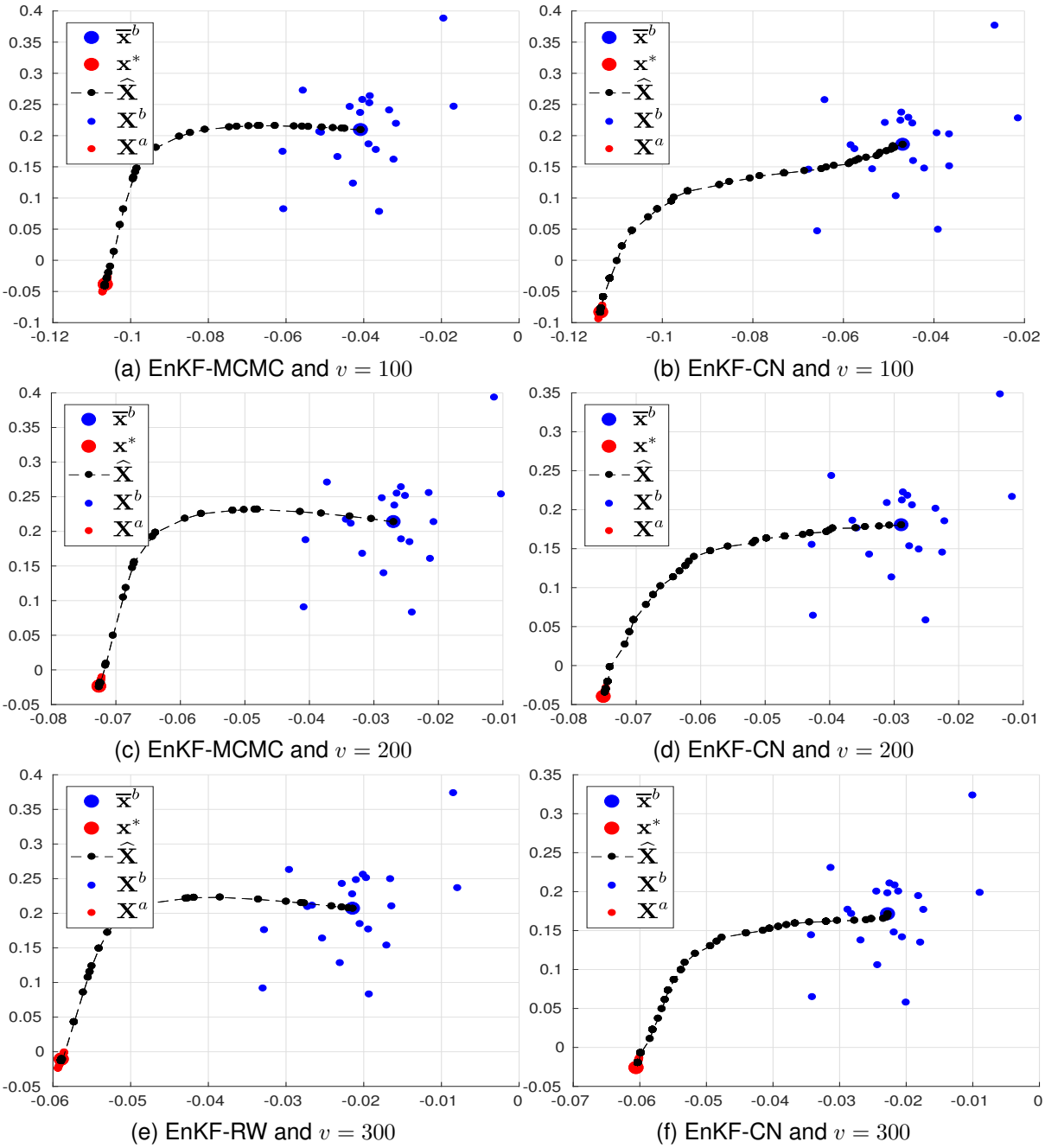


Figure 5: Experimental results with the Lorenz-96 model (4.1) and the exponential observation operator (4.3). The average of the ℓ_2 error norms across ten iterations are shown for different values of v.



(a) EnKF-MCMC and $v = 100$

(b) EnKF-CN and $v = 100$

(c) EnKF-MCMC and $v = 200$

(d) EnKF-CN and $v = 200$

(e) EnKF-RW and $v = 300$

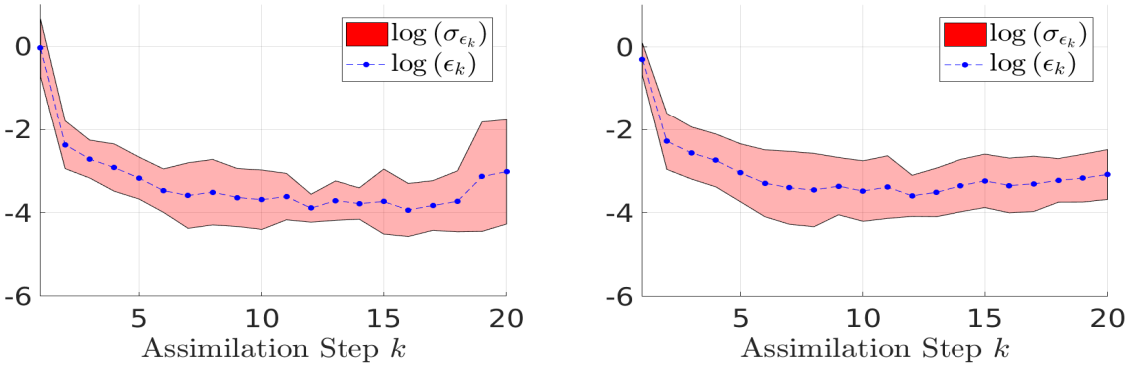
(f) EnKF-CN and $v = 300$

Figure 6: Path followed by the EnKF-RW and the EnKF-CN formulations for a single assimilation step and the exponential observation operator (4.3), and different values of v.

4.2 Assimilation Window - Exponential Observation Operator

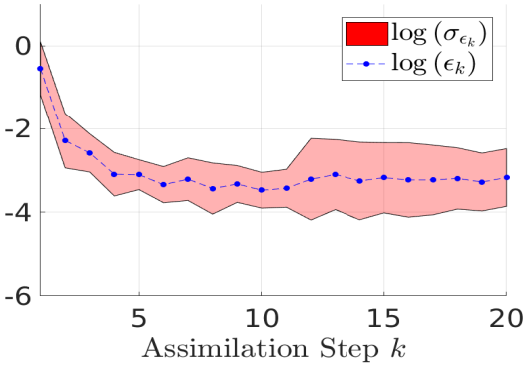
In this section, we test our proposed filters making use of an assimilation window with $M = 20$ observations (as we described above), the exponential observation operator, and three observational networks wherein $s = 70\%$, 80%, and 90% of model components are observed, respectively. Again, for each configuration, ten runs are performed. The observational network is randomly formed at each assimilation cycle. In figure 7 results for $s = 70\%$ and $s = 90\%$ are shown in terms of RMSE in a log scale. As expected, as the number of observed components is increased, a better approximation of the posterior state is obtained by either: the EnKF-

RW or the EnKF-CN. This is a direct consequence in ensemble-based methods since actual information about the physics and the dynamics of model components is injected into the imperfect numerical model. In both cases, posterior errors are lesser than those given by the prior, and even more, their standard deviations are small, considering that no all model components observed. This can be seen in the Table 2. Note that, as the number of iterations increases, the uncertainty of posterior errors can be mitigated.

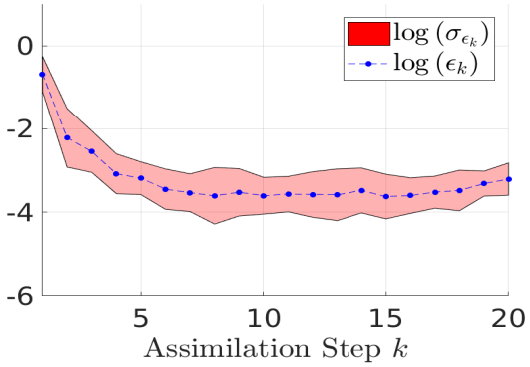


(a) EnKF-RW and $s = 70\%$

(b) EnKF-CN and $s = 70\%$



(c) EnKF-RW and $s = 90\%$



(d) EnKF-CN and $s = 90\%$

Figure 7: Mean and standard deviations, in log scale, of RMSE values across 10 runs for different values of s, and $v = 100$.

		$\log(\epsilon)$		$\log(\sigma_\epsilon)$	
s	v	EnKF-CN	EnKF-RW	EnKF-CN	EnKF-RW
70%	100	-1.65	-1.26	-1.65	-1.26
	200	-1.93	-1.34	-1.93	-1.34
	300	-1.74	-1.67	-1.74	-1.67
80%	100	-1.65	-1.65	-1.65	-1.65
	200	-2.07	-2.04	-2.07	-2.04
	300	-2.12	-1.89	-2.12	-1.89
90%	100	-2.01	-1.78	-2.01	-1.78
	200	-2.04	-1.79	-2.04	-1.79
	300	-1.78	-1.62	-1.78	-1.62

Table 2: Mean and standard deviations of errors for 10 realizations of experiments, and an assimilation window of $M = 20$ observations.

5 Conclusions

Two efficient EnKF formulations based on MCMC methods are proposed. The main idea behind our approaches is to extend the ensemble Kalman filter capabilities to account for non-linear observation operators. The first method employs the well-known Random-Walk to propose candidates while the last formulation does it via the Crank-Nicholson proposal function. The posterior ensemble is built similar to that of ensemble square-root formulations. Experimental tests are performed by using the Lorenz-96 model. Two observation operators are employed during the experiments: a non-smooth and non-linear operator and an exponential one. In both cases, prior errors are decreased by several orders of magnitudes for full observational networks. Moreover, for an assimilation window of 20 observations, the prior error is considerably decreased when 70%, 80%, and 90% of model components are observed in terms of Root-Mean-Square-Errors.

Acknowledgement

This work was supported by the Applied Math and Computer Science Laboratory (AML-CS) at Universidad del Norte, BAQ, COL.

References

Apte, A., Hairer, M., Stuart, A. and Voss, J. 2007. Sampling the posterior: An approach to non-gaussian data assimilation, *Physica D: Nonlinear Phenomena* **230**(1): 50–64.

Asmussen, S. and Glynn, P. W. 2011. A new proof of convergence of mcmc via the ergodic theorem, *Statistics & Probability Letters* **81**(10): 1482–1485.

Attia, A., Rao, V. and Sandu, A. 2015. A sampling approach for four dimensional data assimilation, *Dynamic Data-Driven Environmental Systems Science*, Springer, pp. 215–226.

Attia, A., Rao, V. and Sandu, A. 2017. A hybrid monte-carlo sampling smoother for four-dimensional data assimilation, *International Journal for Numerical Methods in Fluids* **83**(1): 90–112.

Attia, A. and Sandu, A. 2015. A hybrid monte carlo sampling filter for non-gaussian data assimilation, *AIMS Geosciences* **1**: 41–78.

Attia, A., Ştefănescu, R. and Sandu, A. 2017. The reduced-order hybrid monte carlo sampling smoother, *International Journal for Numerical Methods in Fluids* **83**(1): 28–51.

Bannister, R. 2016. A review of operational methods of variational and ensemble-variational data assimilation, *Quarterly Journal of the Royal Meteorological Society* .

Beskos, A., Girolami, M., Lan, S., Farrell, P. E. and Stuart, A. M. 2017. Geometric mcmc for infinite-dimensional inverse problems, *Journal of Computational Physics* **335**: 327–351.

Bickel, D. R. and Padilla, M. 2014. A Prior-free Framework of Coherent Inference and Its Derivation of Simple Shrinkage Estimators, *Journal of Statistical Planning and Inference* **145**(0): 204–221.

Bishop, C. H. and Toth, Z. 1999. Ensemble Transformation and Adaptive Observations, *Journal of the Atmospheric Sciences* **56**(11): 1748–1765.

Buehner, M. 2005. 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.

Cotter, S. L., Roberts, G. O., Stuart, A. M., White, D. et al. 2013. Mcmc methods for functions: modifying old algorithms to make them faster, *Statistical Science* **28**(3): 424–446.

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

Evensen, G. 2009. The ensemble kalman filter for combined state and parameter estimation, *IEEE Control Systems* **29**(3).

Fan, Y., Huang, G., Baetz, B. W., Li, Y., Huang, K., Li, Z., Chen, X. and Xiong, L. 2016. Parameter uncertainty and temporal dynamics of sensitivity for hydrologic models: A hybrid sequential data assimilation and probabilistic collocation method, *Environmental Modelling & Software* **86**: 30–49.

Fertig, E. J., Harlim, J. and Hunt, B. R. 2007. A comparative study of 4d-var and a 4d ensemble kalman filter: Perfect model simulations with lorenz-96, *Tellus A* **59**(1): 96–100.

Gil, R. A., Johanyák, Z. C. and Kovács, T. 2018. Surrogate model based optimization of traffic lights cycles and green period ratios using microscopic simulation and fuzzy rule interpolation, *Int. J. Artif. Intell* **16**(1): 20–40.

Gillijns, S., Mendoza, O., Chandrasekar, J., De Moor, B. L. R., Bernstein, D. and Ridley, A. 2006. What is the Ensemble Kalman Filter and How Well Does It Work?, *American Control Conference, 2006*, pp. 6 pp.–.

Gottwald, G. A. and Melbourne, I. 2005. Testing for chaos in deterministic systems with noise, *Physica D: Nonlinear Phenomena* **212**(1): 100–110.

Hadfield, J. D. et al. 2010. Mcmc methods for multi-response generalized linear mixed models: the mcmcglmm r package, *Journal of Statistical Software* **33**(2): 1–22.

Hoang, V. H., Schwab, C. and Stuart, A. M. 2013. Complexity analysis of accelerated mcmc methods for bayesian inversion, *Inverse Problems* **29**(8): 085010.

Hu, Z., Yao, Z. and Li, J. 2017. On an adaptive preconditioned crank–nicolson mcmc algorithm for infinite dimensional bayesian inference, *Journal of Computational Physics* **332**: 492–503.

Hunt, B. R., Kostelich, E. J. and Szunyogh, I. 2007. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform kalman filter, *Physica D: Nonlinear Phenomena* **230**(1): 112–126.

Karimi, A. and Paul, M. R. 2010. Extensive chaos in the lorenz-96 model, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **20**(4): 043105.

Keppenne, C. L. 2000. Data Assimilation into a Primitive-Equation Model with a Parallel Ensemble Kalman Filter, *Monthly Weather Review* **128**(6): 1971–1981.

Lombardi, M. J. 2007. Bayesian inference for α-stable distributions: A random walk mcmc approach, *Computational statistics & data analysis* **51**(5): 2688–2700.

Lorenc, A. C. 2003. The potential of the ensemble kalman filter for nwp—a comparison with 4d-var, *Quarterly Journal of the Royal Meteorological Society* **129**(595): 3183–3203.

Lorenz, E. N. 2005. Designing chaotic models, *Journal of the Atmospheric Sciences* **62**(5): 1574–1587.

Nino, E. D., Sandu, A. and Deng, X. 2016. An ensemble kalman filter implementation based on modified cholesky decomposition for inverse covariance matrix estimation, *arXiv preprint arXiv:1605.08875* .

Nino-Ruiz, E. D. 2017. A matrix-free posterior ensemble kalman filter implementation based on a modified cholesky decomposition, *Atmosphere* **8**(7): 125.

Nino-Ruiz, E. D. 2018. Implicit surrogate models for trust region based methods, *Journal of Computational Science* .

Nino-Ruiz, E. D. 2020. A numerical method for solving linear systems in the preconditioned crank-nicolson algorithm, *Applied Mathematics Letters* pp. 1–18.

Nino-Ruiz, E. D., Mancilla, A. and Calabria, J. C. 2017. A posterior ensemble kalman filter based on a modified cholesky decomposition, *Procedia Computer Science* **108**: 2049–2058.

Nino-Ruiz, E. D. and Morales-Retat, L. E. 2018. A tabu search implementation for adaptive localization in ensemble-based methods, *Soft Computing* pp. 1–17.

Nino-Ruiz, E. D., Sandu, A. and Deng, X. 2015. A parallel ensemble kalman filter implementation based on modified cholesky decomposition, *Proceedings of the 6th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*, ACM, p. 4.

Nino-Ruiz, E. D., Sandu, A. and Deng, X. 2017. A parallel implementation of the ensemble kalman filter based on modified cholesky decomposition, *Journal of Computational Science* .

Nino-Ruiz, E. D., Sandu, A. and Deng, X. 2018. An ensemble kalman filter implementation based on modified cholesky decomposition for inverse covariance matrix estimation, *SIAM Journal on Scientific Computing* **40**(2): A867–A886.

Nino Ruiz, E., Sandu, A. and Anderson, J. 2014. An Efficient Implementation of the Ensemble Kalman Filter Based on an Iterative Sherman–Morrison Formula, *Statistics and Computing* pp. 1–17.

Nowaková, J., Prílepok, M. and Snášel, V. 2017. Medical image retrieval using vector quantization and fuzzy s-tree, *Journal of medical systems* **41**(2): 18.

Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., Kalnay, E., Patil, D. J. and Yorke, J. A. 2004. A local ensemble kalman filter for atmospheric data assimilation, *Tellus A* **56**(5): 415–428.

Ott, E., Hunt, B., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., Kalnay, E., Patil, D. J. and Yorke, J. A. 2008. A Local Ensemble Transform Kalman Filter Data Assimilation System for the NCEP Global Model, *Tellus A* **60**(1): 113–130.

Pham, D. T. 2001. Stochastic methods for sequential data assimilation in strongly nonlinear systems, *Monthly weather review* **129**(5): 1194–1207.

Plummer, M., Best, N., Cowles, K. and Vines, K. 2006. Coda: convergence diagnosis and output analysis for mcmc, *R news* **6**(1): 7–11.

Precup, R.-E. and Preitl, S. 2004. Optimisation criteria in development of fuzzy controllers with dynamics, *Engineering Applications of Artificial Intelligence* **17**(6): 661–674.

Preitl, S., Precup, R.-E., Preitl, Z., Vaivoda, S., Kilyeni, S. and Tar, J. K. 2007. Iterative feedback and learning control. servo systems applications, *IFAC Proceedings Volumes* **40**(8): 16–27.

Roman, R.-C., Precup, R.-E., Bojan-Dragos, C.-A. and Szedlak-Stinean, A.-I. 2019. Combined model-free adaptive control with fuzzy component by virtual reference feedback tuning for tower crane systems, *Procedia Computer Science* **162**: 267–274.

Ruiz, E. D. N. and Sandu, A. 2016. A derivative-free trust region framework for variational data assimilation, *Journal of Computational and Applied Mathematics* **293**: 164–179.

Sarma, K. K. 2009. Neural network based feature extraction for assamese character and numeral recognition, *International Journal of Artificial Intelligence* **2**(S09): 37–56.

Sengupta, B., Friston, K. J. and Penny, W. D. 2016. Gradient-based mcmc samplers for dynamic causal modelling, *NeuroImage* **125**: 1107–1118.

Snyder, C., Bengtsson, T., Bickel, P. and Anderson, J. 2008. Obstacles to high-dimensional particle filtering, *Monthly Weather Review* **136**(12): 4629–4640.

van Leeuwen, P. J. 2010. Nonlinear data assimilation in geosciences: an extremely efficient particle filter, *Quarterly Journal of the Royal Meteorological Society* **136**(653): 1991–1999.

van Leeuwen, P. J., Cheng, Y. and Reich, S. 2015. Nonlinear data assimilation, vol. 2 of frontiers in applied dynamical systems: Reviews and tutorials.

Zupanski, M. 2005. Maximum likelihood ensemble filter: Theoretical aspects, *Monthly Weather Review* **133**(6): 1710–1726.