SANGOMA: Stochastic Assimilation for the Next Generation Ocean Model Applications EU FP7 SPACE-2011-1 project 283580

Deliverable 3.1:
Non-linear data assimilation methods V0 report
Due date: 01/05/2012
Delivery date: 31/11/2012
Delivery type: Report, public

Peter Jan Van Leeuwen Sanita Vetra-Carvalho
University of Reading, UK

Lars Nerger
Alfred-Wegener-Institute, GERMANY

Arnold Heemink Nils van Velzen
Martin Verlaan M. Umer Altaf
Delft University of Technology, NETHERLANDS

Jean-Marie Beckers Alexander Barth
University of Liège, BELGIUM

Pierre Brasseur Jean-Michel Brankart
CNRS-LEGI, FRANCE

Pierre de Mey
CNRS-LEGOS, FRANCE

Laurent Bertino
NERSC, NORWAY

Contents

1 Introduction **4**
1.1 The problem 4

2 Ensemble Kalman filters **6**
2.1 The original ensemble square root filter (EnSRF) 7
2.2 The ensemble transform Kalman filter (ETKF) 9
2.3 The ensemble adjustment Kalman filter (EAKF) 10
2.4 The singular evolutive interpolated Kalman filter (SEIK) 11
2.5 The error-subspace transform Kalman filter (ESTKF) 12
2.6 The original ensemble Kalman filter (EnKF) 14

3 Particle filters **16**
3.1 The equivalent-weights scheme 17
3.2 The auxiliary particle filter 19

4 Adaptive Gaussian mixture filter **21**

5 Four dimensional ensemble methods **25**
5.1 Ensemble Smoother 26
5.2 Ensemble Kalman Smoother 27
5.3 4DEnVar 28

A Resampling methods **30**
A.1 Probabilistic resampling 30
A.2 Stochastic universal resampling (SUR) 31
A.3 Residual resampling (RR) 32

Bibliography **33**

Chapter 1

Introduction

The main objective of the SANGOMA project is to provide a coherent and transparent database of the current data assimilation methods suitable for non-linear and high-dimensional systems such as MyOcean and even more importantly to advance these probabilistic methods and their applicability to MyOcean systems. In this living document we have described a number of current ensemble and particle filter methods which are applicable to MyOcean systems; these include EnKF, ETKF, EAKF, SEIK, EnSRF, ESTKF, equivalent weight particle filter, auxiliary particle filter and adaptive Gaussian mixture filter. We have also included a selection of resampling techniques to be used with the particle and Gaussian mixture filters.

This is the basic version of the document and as this is a living document we aim to update it regularly including any new development in the non-linear probabilistic data assimilation methods, e.g. new ensemble or particle filter methods or variants of the existing ones which are better suited for MyOcean systems. Thus members of the project are strongly encouraged to share their knowledge of the methods they feel should be included in this document.

1.1 The problem

Consider the following nonlinear stochastic discrete-time dynamical system:

$$\mathbf{x}^{(m)} = \mathcal{M}_m\left(\mathbf{x}^{(m-1)}\right) + \boldsymbol{\beta}^{(m)} \tag{1.1}$$

$$\mathbf{y}^{(m)} = \mathcal{H}_m\left(\mathbf{x}^{(m)}\right) + \boldsymbol{\beta}_R o^{(m)}, \tag{1.2}$$

where $\mathbf{x}^{(m)} \in \mathcal{R}^n$ is the n dimensional state vector (to be estimated), $\mathbf{y}^{(m)} \in \mathcal{R}^{p_m}$ is the $p_m \ll n$ observation vector, $\mathcal{M}_m : \mathcal{R}^n \to \mathcal{R}^n$ is the forward model operator, $\mathcal{H}_m : \mathcal{R}^n \to \mathcal{R}^{p_m}$ is the observation operator, $\boldsymbol{\beta}^{(m)} \in \mathcal{R}^n$ is the model noise (or error) distributed according to a covariance matrix $\mathbf{Q}^{(m)}$and $\boldsymbol{\beta}_R o^{(m)} \in \mathcal{R}^{p_m}$ is the observation noise (or error) distributed according to a covariance matrix $\mathbf{R}^{(m)}$.

We will always denote the time index in the brackets in the upper right corner of the variables, except for operators such as $\mathcal{M}$ and $\mathcal{H}$ where it is in the lower right corner; however, we will omit the time index where it is not necessary to

ease the notation. We will refer to each ensemble member (or each particle) by $\mathbf{x}_j$ where the index $j = 1, ..., N$ and N is the total number of the ensemble members (or particles).

Chapter 2

Ensemble Kalman filters

After first proposed by Evensen [1994] the Ensemble Kalman Filter (EnKF) has become a popular tool for data assimilation because of its computational efficiency, applicability to non-linear dynamics and flexibility [Anderson, 2001, Bengtsson et al., 2003, Evensen, 2007, Ott et al., 2004, Whitaker and Hamill, 2002]. Since then many ensemble filter variants have been proposed, however, all of them have the following steps in common:

- **The initial ensemble generation step:** given an initial state estimate $\mathbf{x}^{a,(0)}$ and initial error covariance matrix $\mathbf{P}^{a,(0)}$ the initial ensemble $\mathbf{X}^{a,(0)} \in \mathcal{R}^{n\times N}$ is sampled around the mean $\mathbf{x}^{a,(0)}$ with covariance $\mathbf{P}^{a,(0)}$.

- **The forecast step:** the ensemble members at each time step between the observations $0 < m \leq k$ are updated using the full non-linear dynamical model:

$$\mathbf{x}_j^{f,(m)} = \mathcal{M}_m\left(\mathbf{x}_j^{a,(m)}\right) + \boldsymbol{\beta}^{(m)}, \tag{2.1}$$

  where $j = 1, ..., N$ is an ensemble member index.

- **The analysis step:** at the observation time k the ensemble forecast mean and covariance are then updated using the available observations to obtain a new analysis ensemble.

The various ensemble methods differ in the analysis step. Here we will discuss the current methods applicable for large-dimensional systems, namely, the original ensemble square root filter (EnSRF) [Tippett et al., 2003], the ensemble transform Kalman filter (ETKF) [Bishop et al., 2001], the ensemble adjustment Kalman filter (EAKF) [Anderson, 2001], the singular evolutive interpolated Kalman filter (SEIK) [Pham et al., 1998], the error subspace transform Kalman filter (ESTKF) [Nerger et al., 2012], and the original ensemble Kalman filter (EnKF) [Evensen, 1994]. We will present these methods in the square root format and point out the different ways the analysis ensemble is obtained in each of the methods. Tippett et al. [2003] gives a uniform framework for EnSRFs, which we follow closely here. In the rest of this section for ease of notation we omit the time index $(\cdot)^{(k)}$ since all of the analysis operations are done at time k.

The ensemble methods are based on the Kalman filter [Kalman, 1960], thus the updated ensemble is given by

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{K}(\mathbf{y} - \mathcal{H}(\overline{\mathbf{x}}^f)) \quad (2.2)$$

$$\mathbf{P}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^f \quad (2.3)$$

where $\mathbf{K}$ is the optimal Kalman gain given by

$$\mathbf{K} = \mathbf{P}^f\mathbf{H}^\mathsf{T}(\mathbf{H}\mathbf{P}^f\mathbf{H}^\mathsf{T} + \mathbf{R})^{-1}. \quad (2.4)$$

The analysis ensemble is then given by

$$\mathbf{X}^a = \overline{\mathbf{X}}^a + \mathbf{X}'^a, \quad (2.5)$$

where $\overline{\mathbf{X}}^a = (\overline{\mathbf{x}}^a, ..., \overline{\mathbf{x}}^a) \in \mathcal{R}^{n\times N}$ is a matrix with the ensemble analysis mean in each column and ensemble analysis perturbations are a scaled matrix square root of

$$\mathbf{P}^a = \frac{\mathbf{X}'^a\,(\mathbf{X}'^a)^\mathsf{T}}{N-1}.$$

To obtain the general square root form we write

$$\begin{aligned}
\mathbf{X}'^a(\mathbf{X}'^a)^\mathsf{T} &= \left(\mathbf{I} - \mathbf{P}^f\mathbf{H}^\mathsf{T}(\mathbf{H}\mathbf{P}^f\mathbf{H}^\mathsf{T} + \mathbf{R})^{-1}\mathbf{H}\right)\mathbf{X}'^f(\mathbf{X}'^f)^\mathsf{T} \\
&= \left(\mathbf{I} - \mathbf{X}'^f(\mathbf{X}'^f)^\mathsf{T}\mathbf{H}^\mathsf{T}(\mathbf{H}\mathbf{X}'^f(\mathbf{X}'^f)^\mathsf{T}\mathbf{H}^\mathsf{T} + (N-1)\mathbf{R})^{-1}\mathbf{H}\right)\mathbf{X}'^f(\mathbf{X}'^f)^\mathsf{T} \\
&= \mathbf{X}'^f\left(\mathbf{I} - (\mathbf{H}\mathbf{X}'^f)^\mathsf{T}(\mathbf{H}\mathbf{X}'^f(\mathbf{H}\mathbf{X}'^f)^\mathsf{T} + (N-1)\mathbf{R})^{-1}\mathbf{H}\mathbf{X}'^f\right)(\mathbf{X}'^f)^\mathsf{T} \\
&= \mathbf{X}'^f\left(\mathbf{I} - \mathbf{S}^\mathsf{T}(\mathbf{S}\mathbf{S}^\mathsf{T} + (N-1)\mathbf{R})^{-1}\mathbf{S}\right)(\mathbf{X}'^f)^\mathsf{T} \\
&= \mathbf{X}'^f\left(\mathbf{I} - \mathbf{S}^\mathsf{T}\mathbf{F}^{-1}\mathbf{S}\right)(\mathbf{X}'^f)^\mathsf{T}, \quad (2.6)
\end{aligned}$$

where $\mathbf{S} = (\mathbf{H}\mathbf{X}'^f)$ is the ensemble observation matrix and $\mathbf{F} = \mathbf{S}\,\mathbf{S}^\mathsf{T} + (N-1)\,\mathbf{R}$ is the innovation covariance. Thus to find the updated ensemble analysis perturbations $\mathbf{X}'^a$ we need to compute the square root of the matrix $\mathbf{T}\mathbf{T}^\mathsf{T} = \mathbf{I} - \mathbf{S}^\mathsf{T}\,\mathbf{F}^{-1}\,\mathbf{S}$, where $\mathbf{T}$ is called a transform matrix. Different ways exist to compute the transform matrix $\mathbf{T}$ and here we will discuss the current methods applicable for large-dimensional systems.

2.1 The original ensemble square root filter (EnSRF)

Here we first perform an eigenvalue decomposition to invert $\mathbf{F}$; that is,

$$\mathbf{F}^{-1} = \mathbf{\Gamma}_S\mathbf{\Lambda}_S^{-1}\mathbf{\Gamma}_S^\mathsf{T}. \quad (2.7)$$

Thus, the ensemble analysis covariance is given by

$$\begin{aligned}
\mathbf{X}'^a(\mathbf{X}'^a)^\mathsf{T} &= \mathbf{X}'^f\left(\mathbf{I} - \mathbf{S}^\mathsf{T}\mathbf{\Gamma}_S\mathbf{\Lambda}_S^{-1}\mathbf{\Gamma}_S^\mathsf{T}\mathbf{S}\right)(\mathbf{X}'^f)^\mathsf{T} \\
&= \mathbf{X}'^f\left(\mathbf{I} - \mathbf{S}^\mathsf{T}\mathbf{\Gamma}_S\mathbf{\Lambda}_S^{-1/2}\mathbf{\Lambda}_S^{-1/2}\mathbf{\Gamma}_S^\mathsf{T}\mathbf{S}\right)(\mathbf{X}'^f)^\mathsf{T} \\
&= \mathbf{X}'^f\left(\mathbf{I} - \left[\mathbf{S}^\mathsf{T}\mathbf{\Gamma}_S\mathbf{\Lambda}_S^{-1/2}\right]\left[\mathbf{S}^\mathsf{T}\mathbf{\Gamma}_S\mathbf{\Lambda}_S^{-1/2}\right]^\mathsf{T}\right)(\mathbf{X}'^f)^\mathsf{T} \\
&= \mathbf{X}'^f\left(\mathbf{I} - \mathbf{X}_S\mathbf{X}_S^\mathsf{T}\right)(\mathbf{X}'^f)^\mathsf{T} \quad (2.8)
\end{aligned}$$

where $\mathbf{X}_S = \mathbf{S}^\mathsf{T}\mathbf{\Gamma}_S\mathbf{\Lambda}_S^{-1/2}$ and by decomposing $\mathbf{X}_S = \mathbf{U}_S\mathbf{\Sigma}_S\mathbf{Z}_S$ using SVD gives

$$\begin{aligned}
\mathbf{X'}^a(\mathbf{X'}^a)^\mathsf{T} &= \mathbf{X'}^f\left(\mathbf{I} - [\mathbf{U}_S\mathbf{\Sigma}_S\mathbf{Z}_S]\,[\mathbf{U}_S\mathbf{\Sigma}_S\mathbf{Z}_S]^\mathsf{T}\right)(\mathbf{X'}^f)^\mathsf{T} \\
&= \mathbf{X'}^f\left(\mathbf{I} - \mathbf{U}_S\mathbf{\Sigma}_S\mathbf{\Sigma}_S^\mathsf{T}\mathbf{U}_S^\mathsf{T}\right)(\mathbf{X'}^f)^\mathsf{T} \\
&= \mathbf{X'}^f\mathbf{U}_S\left(\mathbf{I} - \mathbf{\Sigma}_S\mathbf{\Sigma}_S^\mathsf{T}\right)\mathbf{U}_S^\mathsf{T}(\mathbf{X'}^f)^\mathsf{T} \\
&= \mathbf{X'}^f\mathbf{U}_S\left(\mathbf{I} - \mathbf{\Sigma}_S\mathbf{\Sigma}_S^\mathsf{T}\right)^{1/2}\left[\mathbf{X'}^f\mathbf{U}_S\left(\mathbf{I} - \mathbf{\Sigma}_S\mathbf{\Sigma}_S^\mathsf{T}\right)^{1/2}\right]^\mathsf{T} \qquad (2.9)
\end{aligned}$$

Thus ensemble analysis perturbations are given by

$$\mathbf{X'}^a = \mathbf{X'}^f\mathbf{U}_S\left(\mathbf{I} - \mathbf{\Sigma}_S\mathbf{\Sigma}_S^\mathsf{T}\right)^{1/2}\mathbf{U}_S^\mathsf{T}, \qquad (2.10)$$

where we have post-multiplied the ensemble analysis perturbations by the orthogonal matrix of the left singular vectors $\mathbf{U}_S^\mathsf{T}$ for the analysis ensemble to remain unbiased as suggested in [Livings et al., 2008, Sakov and Oke, 2008].

The ensemble analysis mean is obtained from

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{X'}^f\mathbf{S}^\mathsf{T}\mathbf{\Gamma}_S\mathbf{\Lambda}_S^{-1}\mathbf{\Gamma}_S^\mathsf{T}\left(\mathbf{y} - \mathcal{H}(\overline{\mathbf{x}}^f)\right). \qquad (2.11)$$

We note that we can combine both of the ensemble analysis update steps for the ensemble mean and perturbations into one step as follows,

$$\mathbf{X}^a = \overline{\mathbf{X}}^f + \mathbf{X'}^f\left(\overline{\mathbf{W}}_S + \mathbf{W'}_S\right), \qquad (2.12)$$

where $\overline{\mathbf{W}}_S = (\overline{\mathbf{w}}_S, ..., \overline{\mathbf{w}}_S)$ with

$$\overline{\mathbf{w}}_S = \mathbf{S}^\mathsf{T}\mathbf{\Gamma}_S\mathbf{\Lambda}_S^{-1}\mathbf{\Gamma}_S^\mathsf{T}\left(\mathbf{y} - \mathcal{H}(\overline{\mathbf{x}}^f)\right) \qquad (2.13)$$

and

$$\mathbf{W'}_S = \mathbf{U}_S\left(\mathbf{I} - \mathbf{\Sigma}_S\mathbf{\Sigma}_S^\mathsf{T}\right)^{1/2}\mathbf{U}_S. \qquad (2.14)$$

Typically, the EnSRF is applied with serial treatment of observations. In this case, each single observation is assimilated separately. Thus, $\mathbf{F}$ reduces to the scalar F and $\mathbf{S}\mathbf{S}^\mathsf{T}$ to the scalar S^2 . This method is possible if $\mathbf{R}$ is diagonal. For a single observation ($m = 1$), the matrix $\mathbf{X}_S$ becomes a vector given by:

$$\mathbf{X}_S = \frac{1}{\sqrt{F}}\mathbf{S}^\mathsf{T}. \qquad (2.15)$$

All singular values of $\mathbf{X}_S$ are zero except the first which is its norm,

$$\mathbf{\Sigma_S} = \frac{S}{\sqrt{F}}\mathbf{e} \qquad (2.16)$$

where $\mathbf{e}$ is a vector with N zero elements except the first which is 1. The first column of $\mathbf{U}_S$ corresponds to normalized vector $\mathbf{S}^\mathsf{T}$.

$$\mathbf{U}_S\mathbf{e} = \frac{1}{S}\mathbf{S}^\mathsf{T} \qquad (2.17)$$

The square root of the diagonal matrix in 2.10 can be written as a sum of the identify matrix and a matrix proportional to $\mathbf{e}\mathbf{e}^{\mathsf{T}}$:

$$\left(\mathbf{I}-\mathbf{\Sigma}_S\mathbf{\Sigma}_S^{\mathsf{T}}\right)^{1/2}=\mathbf{I}-\left(1-\sqrt{(N-1)R/F}\right)\mathbf{e}\mathbf{e}^{\mathsf{T}} \tag{2.18}$$

Using equation 2.17 and the fact that all columns of $\mathbf{U}$ are orthonormal, one obtains the following expression for $\mathbf{W'}_S$:

$$\mathbf{W'}_S=\mathbf{I}-\frac{1-\sqrt{(N-1)R/F}}{S^2}\mathbf{S}^{\mathsf{T}}\mathbf{S} \tag{2.19}$$

The Kalman gain $\mathbf{K}=\frac{1}{F}\mathbf{X'}^f\mathbf{S}^{\mathsf{T}}$ in combination with equation 2.19 allows to compute the analyzed state $\overline{\mathbf{x}}^a$ and the ensemble analysis perturbations by:

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f+\mathbf{K}\left(\mathbf{y}-\mathcal{H}(\overline{\mathbf{x}}^f)\right) \tag{2.20}$$

$$\mathbf{X'}^a = \mathbf{X'}^f-\alpha\mathbf{K}\mathbf{S} \tag{2.21}$$

where the coefficient α is given by

$$\begin{aligned}\alpha &= \frac{F}{S^2}(1-\sqrt{(N-1)R/F})\\ &= \frac{F}{S^2}\frac{1-(N-1)R/F}{1+\sqrt{(N-1)R/F}}\\ &= \frac{1}{1+\sqrt{(N-1)R/F}}\end{aligned}$$

2.2 The ensemble transform Kalman filter (ETKF)

In this method we require the inverse of the observation error covariance matrix $\mathbf{R}^{-1}$ to be readily available, then using Sherman-Morrison-Woodbury identity [Golub and Van Loan, 1996] we can rewrite $\mathbf{T}\mathbf{T}^{\mathsf{T}}$ as follows,

$$\mathbf{T}\mathbf{T}^{\mathsf{T}}=\mathbf{I}-\mathbf{S}^{\mathsf{T}}\mathbf{F}^{-1}\mathbf{S}=\left(\mathbf{I}+\frac{1}{N-1}\mathbf{S}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{S}\right)^{-1}. \tag{2.22}$$

We normalise the forecast observation ensemble perturbation matrix so that the observations are dimensionless with standard deviation one [Livings, 2005],

$$\tilde{\mathbf{S}}=\frac{1}{\sqrt{N-1}}\mathbf{R}^{-1/2}\mathbf{S}, \tag{2.23}$$

thus $\tilde{\mathbf{S}}\tilde{\mathbf{S}}^{\mathsf{T}}=\mathbf{S}\mathbf{R}^{-1}\mathbf{S}^{\mathsf{T}}$. Substituting (2.23) into (2.22) we obtain

$$\mathbf{T}\mathbf{T}^{\mathsf{T}}=\left(\mathbf{I}+\tilde{\mathbf{S}}^{\mathsf{T}}\tilde{\mathbf{S}}\right)^{-1}. \tag{2.24}$$

Next we perform a SVD on the scaled forecast ensemble observation perturbation matrix,

$$\tilde{\mathbf{S}}^{\mathsf{T}}=\mathbf{U}_T\mathbf{\Sigma}_T\mathbf{V}_T^{\mathsf{T}}. \tag{2.25}$$

By using a singular value decomposition we avoid performing the multiplication in $\tilde{\mathbf{S}}\tilde{\mathbf{S}}^{\mathsf{T}}$ thus preventing possible loss of accuracy due to rounding errors. Thus, we have

$$\begin{aligned}\mathbf{T}\mathbf{T}^{\mathsf{T}} &= \left(\mathbf{I}+(\mathbf{U}_T\mathbf{\Sigma}_T\mathbf{V}_T^{\mathsf{T}})(\mathbf{U}_T\mathbf{\Sigma}_T\mathbf{V}_T^{\mathsf{T}})^{\mathsf{T}}\right)^{-1}\\ &= \left(\mathbf{I}+\mathbf{U}_T\mathbf{\Sigma}_T\mathbf{\Sigma}_T^{\mathsf{T}}\mathbf{U}_T^{\mathsf{T}}\right)^{-1}\\ &= \mathbf{U}_T\left(\mathbf{I}+\mathbf{\Sigma}_T\mathbf{\Sigma}_T^{\mathsf{T}}\right)^{-1}\mathbf{U}_T^{\mathsf{T}}. \end{aligned} \tag{2.26}$$

The ensemble analysis perturbations are then given by

$$\mathbf{X'}^a = \mathbf{X'}^f\mathbf{U}_T\left(\mathbf{I}+\mathbf{\Sigma}_T\mathbf{\Sigma}_T^{\mathsf{T}}\right)^{-1/2}\mathbf{U}_T^{\mathsf{T}}. \tag{2.27}$$

and the ensemble mean is updated via

$$\begin{aligned}\overline{\mathbf{x}}^a &= \overline{\mathbf{x}}^f+\mathbf{K}(\mathbf{y}-\mathcal{H}(\overline{\mathbf{x}}^f))\\ &= \overline{\mathbf{x}}^f+\frac{1}{\sqrt{N-1}}\mathbf{X'}^f\mathbf{U}_T(\mathbf{\Sigma}_T^{\mathsf{T}}\mathbf{\Sigma}_T+\mathbf{I})^{-1}\mathbf{\Sigma}_T\mathbf{V}_T^{\mathsf{T}}\mathbf{R}^{-1/2}(\mathbf{y}-\mathcal{H}(\overline{\mathbf{x}}^f)), \end{aligned} \tag{2.28}$$

where

$$\begin{aligned}\mathbf{K} &= \mathbf{X'}^f\mathbf{S}^{\mathsf{T}}(\mathbf{S}\mathbf{S}^{\mathsf{T}}+(N-1)\mathbf{R})^{-1}\\ &= \frac{1}{\sqrt{N-1}}\mathbf{X'}^f\tilde{\mathbf{S}}^{\mathsf{T}}\mathbf{R}^{1/2}\mathbf{R}^{-1/2}(\tilde{\mathbf{S}}\tilde{\mathbf{S}}^{\mathsf{T}}+\mathbf{I})^{-1}\mathbf{R}^{-1/2}\\ &= \frac{1}{\sqrt{N-1}}\mathbf{X'}^f\tilde{\mathbf{S}}^{\mathsf{T}}(\tilde{\mathbf{S}}\tilde{\mathbf{S}}^{\mathsf{T}}+\mathbf{I})^{-1}\mathbf{R}^{-1/2}\\ &= \frac{1}{\sqrt{N-1}}\mathbf{X'}^f\mathbf{U}_T\mathbf{\Sigma}_T\mathbf{V}_T^{\mathsf{T}}(\mathbf{V}_T\mathbf{\Sigma}_T^{\mathsf{T}}\mathbf{\Sigma}_T\mathbf{V}_T^{\mathsf{T}}+\mathbf{I})^{-1}\mathbf{R}^{-1/2}\\ &= \frac{1}{\sqrt{N-1}}\mathbf{X'}^f\mathbf{U}_T(\mathbf{\Sigma}_T^{\mathsf{T}}\mathbf{\Sigma}_T+\mathbf{I})^{-1}\mathbf{\Sigma}_T\mathbf{V}_T^{\mathsf{T}}\mathbf{R}^{-1/2}. \end{aligned} \tag{2.29}$$

Again we can combine both of the ensemble analysis update steps for the ensemble mean and perturbations into one step as follows,

$$\mathbf{X}^a = \overline{\mathbf{X}}^f+\mathbf{X'}^f\left(\overline{\mathbf{W}}_T+\mathbf{W'}_T\right), \tag{2.30}$$

where $\overline{\mathbf{W}}_T=(\overline{\mathbf{w}}_T,...,\overline{\mathbf{w}}_T)$ with

$$\overline{\mathbf{w}}_T = \frac{1}{\sqrt{N-1}}\mathbf{U}_T(\mathbf{\Sigma}_T^{\mathsf{T}}\mathbf{\Sigma}_T+\mathbf{I})^{-1}\mathbf{\Sigma}_T\mathbf{V}_T^{\mathsf{T}}\mathbf{R}^{-1/2}(\mathbf{y}-\mathcal{H}(\overline{\mathbf{x}}^f)) \tag{2.31}$$

and

$$\mathbf{W'}_T = \mathbf{U}_T\left(\mathbf{I}+\mathbf{\Sigma}_T\mathbf{\Sigma}_T^{\mathsf{T}}\right)^{-1/2}\mathbf{U}_T^{\mathsf{T}}. \tag{2.32}$$

2.3 The ensemble adjustment Kalman filter (EAKF)

In this method as in the ETKF we require the inverse of the observation error covariance matrix $\mathbf{R}^{-1}$ to be readily available, then as in the ETKF using Sherman-Morrison-Woodbury identity [Golub and Van Loan, 1996] and scaling the ensemble forecast observation perturbations we can rewrite $\mathbf{T}\mathbf{T}^{\mathsf{T}}$ as follows,

$$\mathbf{T}\mathbf{T}^{\mathsf{T}} = \left(\mathbf{I}+\tilde{\mathbf{S}}\tilde{\mathbf{S}}^{\mathsf{T}}\right)^{-1}. \tag{2.33}$$

The same way as in the ETKF we perform a singular value decomposition on the scaled forecast ensemble observation perturbation matrix

$$\tilde{\mathbf{S}}^{\mathsf{T}} = \mathbf{U}_A \mathbf{\Sigma}_A \mathbf{V}_A^{\mathsf{T}}, \tag{2.34}$$

and additionally in the EAKF we also use eigenvalue decomposition to obtain

$$\mathbf{P}^f = \mathbf{Z}_A \mathbf{\Gamma}_A \mathbf{Z}_A^{\mathsf{T}}. \tag{2.35}$$

Then the ensemble analysis perturbations are given by

$$\mathbf{X'}^a = \frac{1}{\sqrt{N-1}} \mathbf{X'}^f \mathbf{U}_A \left(\mathbf{I} + \mathbf{\Sigma}_A \mathbf{\Sigma}_A^{\mathsf{T}} \right)^{-1/2} \mathbf{\Gamma}_A^{-1/2} \mathbf{Z}_A^{\mathsf{T}} \mathbf{X'}^f. \tag{2.36}$$

We note that the EAKF ensemble perturbation analysis given by (2.36) is the same as applying the orthogonal matrix $\mathbf{\Gamma}_A^{-1/2} \mathbf{Z}_A^{\mathsf{T}} \mathbf{X'}^f$ instead of the orthogonal matrix $\mathbf{U}_T$ in the ETKF ensemble analysis perturbations given by (2.27) [Tippett et al., 2003].

The ensemble mean analysis is given the same as in the ETKF, and the ensemble mean is updated via

$$\begin{aligned} \overline{\mathbf{x}}^a &= \overline{\mathbf{x}}^f + \mathbf{K}(\mathbf{y} - \mathcal{H}(\overline{\mathbf{x}}^f)) \\ &= \overline{\mathbf{x}}^f + \frac{1}{\sqrt{N-1}} \mathbf{X'}^f \mathbf{U}_A (\mathbf{\Sigma}_A^{\mathsf{T}} \mathbf{\Sigma}_A + \mathbf{I})^{-1} \mathbf{V}_A^{\mathsf{T}} \mathbf{R}^{-1/2} (\mathbf{y} - \mathcal{H}(\overline{\mathbf{x}}^f)). \end{aligned} \tag{2.37}$$

Again we can combine both of the ensemble analysis update steps for the ensemble mean and perturbations into one step as follows,

$$\mathbf{X}^a = \overline{\mathbf{X}}^f + \mathbf{X'}^f \left(\overline{\mathbf{W}}_A + \mathbf{W'}_A \right), \tag{2.38}$$

where $\overline{\mathbf{W}}_A = (\overline{\mathbf{w}}_A, ..., \overline{\mathbf{w}}_A)$ with

$$\overline{\mathbf{w}}_A = \frac{1}{\sqrt{N-1}} \mathbf{U}_A (\mathbf{\Sigma}_A^{\mathsf{T}} \mathbf{\Sigma}_A + \mathbf{I})^{-1} \mathbf{V}_A^{\mathsf{T}} \mathbf{R}^{-1/2} (\mathbf{y} - \mathcal{H}(\overline{\mathbf{x}}^f)) \tag{2.39}$$

and

$$\mathbf{W'}_A = \mathbf{U}_A \left(\mathbf{I} + \mathbf{\Sigma}_A \mathbf{\Sigma}_A^{\mathsf{T}} \right)^{-1/2} \mathbf{\Gamma}_A^{-1/2} \mathbf{Z}_A^{\mathsf{T}} \mathbf{X'}^f. \tag{2.40}$$

Due to the large computational cost to compute the decomposition 2.35, the vector update of the EAKF is not applicable for large-scale systems. For this reason, it is typically applied with serial observation processing as the EnSRF.

2.4 The singular evolutive interpolated Kalman filter (SEIK)

The SEIK filter, while similar to the ETKF, has some important differences. In particular, the SEIK filter performs the analysis step in the ensemble error subspace. This is achieved by defining a matrix

$$\mathbf{L} = \mathbf{X}^f \mathbf{A}, \tag{2.41}$$

where $\mathbf{A} \in \mathcal{R}^{N \times N-1}$ is a matrix with full rank and zero column sums. Previously, matrix $\mathbf{A}$ was always identified as

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{N-1 \times N-1} \\ \mathbf{0}_{1 \times N-1} \end{bmatrix} - \frac{1}{N} \left[\mathbf{1}_{N \times N-1} \right], \tag{2.42}$$

where $\mathbf{0}$ is a matrix whose elements are equal to zero and $\mathbf{1}$ is a matrix whose elements are equal to one [Nerger et al., 2012]. Matrix $\mathbf{A}$ implicitly subtracts the ensemble mean when the matrix $\mathbf{L}$ is computed. In addition, $\mathbf{A}$ removes the last column of $\mathbf{X'}^f$. Thus $\mathbf{L}$ is an $N \times N-1$ matrix that holds the first $N-1$ ensemble perturbations. The product of the square root matrices in the ensemble error space becomes now

$$\mathbf{T}\mathbf{T}^\mathsf{T} = \left((N-1)\mathbf{A}^\mathsf{T}\mathbf{A} + (\mathbf{H}\mathbf{L})^\mathsf{T}\mathbf{R}^{-1}(\mathbf{H}\mathbf{L}) \right)^{-1}. \tag{2.43}$$

In contrast to the ETKF, $\mathbf{T}\mathbf{T}^\mathsf{T}$ is of size $N-1 \times N-1$. The square root $\mathbf{T}$ is obtained from Cholesky decomposition of $(\mathbf{T}\mathbf{T}^\mathsf{T})^{-1}$. Then, the ensemble analysis perturbations are given by

$$\mathbf{X'}^a = \sqrt{N-1}\, \mathbf{L}\mathbf{T}\mathbf{\Omega}, \tag{2.44}$$

where columns of $\mathbf{\Omega} \in \mathcal{R}^{N \times N-1}$ are orthonormal and orthogonal to the vector $(1, \ldots, 1)^\mathsf{T}$. $\mathbf{\Omega}$ can be either random or deterministic rotation matrix. However, if a deterministic $\mathbf{\Omega}$ is used then Nerger et al. [2012] shows that a symmetric square root of $\mathbf{T}\mathbf{T}^\mathsf{T}$, like in the ETKF, should be used for a more stable ensemble.

The ensemble analysis mean is given then by

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{L}\mathbf{T}\mathbf{T}^\mathsf{T}(\mathbf{H}\mathbf{L})^\mathsf{T}\mathbf{R}^{-1} \left(\mathbf{y} - \mathcal{H}(\overline{\mathbf{x}}^f) \right). \tag{2.45}$$

Again we can combine both of the ensemble analysis update steps for the ensemble mean and perturbations into one step as follows,

$$\mathbf{X}^a = \overline{\mathbf{X}}^f + \mathbf{L} \left(\overline{\mathbf{W}}_K + \mathbf{W'}_K \right), \tag{2.46}$$

where $\overline{\mathbf{W}}_K = (\overline{\mathbf{w}}_K, ..., \overline{\mathbf{w}}_K)$ with

$$\overline{\mathbf{w}}_K = \mathbf{L}\mathbf{T}\mathbf{T}^\mathsf{T}(\mathbf{H}\mathbf{L})^\mathsf{T}\mathbf{R}^{-1} \left(\mathbf{y} - \mathcal{H}(\overline{\mathbf{x}}^f) \right), \tag{2.47}$$

and

$$\mathbf{W'}_K = \mathbf{T}\mathbf{\Omega}. \tag{2.48}$$

2.5 The error-subspace transform Kalman filter (ESTKF)

Recently, the error-subspace transform Kalman filter (ESTKF) has been derived from the SEIK filter [Nerger et al., 2012]. The ESTKF exhibits better properties than the SEIK filter, like a minimum ensemble transformation like the ETKF. However, unlike the ETKF the ESTKF computes the ensemble transformation in the

error subspace spanned by the ensemble rather than using the ensemble representation of it. Similar to the SEIK filter, a projection matrix $\hat{\mathbf{A}} \in \mathcal{R}^{N \times N-1}$ is used whose elements are defined by

$$\hat{\mathbf{A}}_{i,j} := \begin{cases} 1 - \frac{1}{N} \frac{1}{\frac{1}{\sqrt{N}}+1} & \text{for } i = j, i < N \\ -\frac{1}{N} \frac{1}{\frac{1}{\sqrt{N}}+1} & \text{for } i \neq j, i < N \\ -\frac{1}{\sqrt{N}} & \text{for } i = N \end{cases} \tag{2.49}$$

With this projection, the basis vectors for the error subspace are given by

$$\mathbf{L} = \mathbf{X}^f \hat{\mathbf{A}}, \tag{2.50}$$

As for matrix $\mathbf{\Omega}$ in the SEIK filter, the columns of Matrix $\hat{\mathbf{A}}$ are orthonormal and orthogonal to the vector (1, . . . , 1)T. When the matrix $\mathbf{L}$ is computed, the multiplication with $\hat{\mathbf{A}}$ implicitly subtracts the ensemble mean. Further, $\hat{\mathbf{A}}$ subtracts a fraction of the last column of $\mathbf{X'}^f$ from all other columns. In this way, the last column of $\mathbf{X'}^f$ is not just dropped as in the SEIK filter, but its information is distributed over the other columns. The product of the square root matrices in the error subspace becomes now

$$\mathbf{T}\mathbf{T}^{\mathsf{T}} = ((N-1)\mathbf{I} + (\mathbf{H}\mathbf{L})^{\mathsf{T}}\mathbf{R}^{-1}(\mathbf{H}\mathbf{L}))^{-1}. \tag{2.51}$$

By performing the SVD of the symmetric matrix $(\mathbf{T}\mathbf{T}^{\mathsf{T}})^{-1} = \mathbf{U}_E \mathbf{\Sigma}_E \mathbf{U}_E^{\mathsf{T}}$ we obtain the symmetric square root

$$\mathbf{T} = \mathbf{U}_E \mathbf{\Sigma}_E^{-1/2} \mathbf{U}_E^{\mathsf{T}}. \tag{2.52}$$

The matrix of ensemble analysis perturbations is given by

$$\mathbf{X'}^a = \sqrt{N-1}\, \mathbf{L}\mathbf{T}\hat{\mathbf{A}}^{\mathsf{T}}. \tag{2.53}$$

The ensemble analysis mean is given then by

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{L}\mathbf{U}_E \mathbf{\Sigma}_E^{-1} \mathbf{U}_E^{\mathsf{T}} (\mathbf{H}\mathbf{L})^{\mathsf{T}} \mathbf{R}^{-1} \left(\mathbf{y} - \mathcal{H}(\overline{\mathbf{x}}^f) \right). \tag{2.54}$$

Again we can combine both of the ensemble analysis update steps for the ensemble mean and perturbations into one step as follows,

$$\mathbf{X}^a = \overline{\mathbf{X}}^f + \mathbf{L} \left(\overline{\mathbf{W}}_E + \mathbf{W'}_E \right), \tag{2.55}$$

where $\overline{\mathbf{W}}_E = (\overline{\mathbf{w}}_E, ..., \overline{\mathbf{w}}_E)$ with

$$\overline{\mathbf{w}}_E = \mathbf{U}_E \mathbf{\Sigma}_E^{-1} \mathbf{U}_E^{\mathsf{T}} (\mathbf{H}\mathbf{L})^{\mathsf{T}} \mathbf{R}^{-1} \left(\mathbf{y} - \mathcal{H}(\overline{\mathbf{x}}^f) \right), \tag{2.56}$$

and

$$\mathbf{W'}_E = \mathbf{U}_E \mathbf{\Sigma}_E^{-1/2} \mathbf{U}_E^{\mathsf{T}} \hat{\mathbf{A}}. \tag{2.57}$$

Compared to the SEIK filter, both the matrices $\mathbf{A}$ and $\mathbf{\Omega}$ are replaced by $\hat{\mathbf{A}}$ in the ESTKF. In addition, the ESTKF uses the symmetric square root of $\mathbf{T}\mathbf{T}^{\mathsf{T}}$. The use of $\hat{\mathbf{A}}$ leads to consistent projection onto the error subspace and back onto the state space, while the symmetric square root ensures that the minimum transformation is obtained. It is also possible to apply the ESTKF with a random ensemble transformation. For this case, the matrix $\hat{\mathbf{A}}$ in equations (2.53) and (2.57) is replaced by a random matrix with the same properties as the deterministic $\mathbf{hatA}$.

2.6 The original ensemble Kalman filter (EnKF)

The original EnKF as introduced by Evensen [1994] is the only method discussed here with perturbed observations, that is

$$\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N) \in \mathcal{R}^{m \times N}, \tag{2.58}$$

where perturbed observations are given by $\mathbf{y}_j = \mathbf{y} + \boldsymbol{\epsilon}_j$, $j = 1, ..., N$ with the added noise drawn from some distribution with mean $\mathbf{y}$ and covariance $\mathbf{R}$. Thus the observational noise (perturbation) matrix

$$\mathbf{Y'} = (\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, ..., \boldsymbol{\epsilon}_N) \in \mathcal{R}^{m \times N} \tag{2.59}$$

forms the ensemble representation of the observational errors as follows

$$\mathbf{R}_e = \frac{\mathbf{Y'Y'}^{\mathsf{T}}}{N-1}. \tag{2.60}$$

Using an ensemble representation of the observation errors results in a modified innovation covariance

$$\hat{\mathbf{F}} = \mathbf{S}\mathbf{S}^{\mathsf{T}} + \mathbf{Y'Y'}^{\mathsf{T}} \tag{2.61}$$

and a modified transform matrix

$$\hat{\mathbf{T}}\hat{\mathbf{T}}^{\mathsf{T}} = \mathbf{I} - \mathbf{S}^{\mathsf{T}}\hat{\mathbf{F}}^{-1}\mathbf{S}. \tag{2.62}$$

We can decompose $\hat{\mathbf{F}}$ using an eigenvalue decomposition but this is costly if $m \gg N$ [Evensen, 2003]. Instead, if forecast and observation errors are uncorrelated (as they are assumed to be), i.e.

$$\mathbf{S}\mathbf{Y'}^{\mathsf{T}} \equiv 0, \tag{2.63}$$

then

$$\hat{\mathbf{F}} = \mathbf{S}\mathbf{S}^{\mathsf{T}} + \mathbf{Y'}\mathbf{Y'}^{\mathsf{T}} = (\mathbf{S} + \mathbf{Y'})(\mathbf{S} + \mathbf{Y'})^{\mathsf{T}} \tag{2.64}$$

and we can use SVD to decompose $\mathbf{S} + \mathbf{Y'} = \mathbf{U}_F \boldsymbol{\Sigma}_F \mathbf{V}_F$, hence

$$\hat{\mathbf{F}} = \mathbf{U}_F \boldsymbol{\Sigma}_F \boldsymbol{\Sigma}_F^{\mathsf{T}} \mathbf{U}_F^{\mathsf{T}}. \tag{2.65}$$

This has a much smaller computational cost [Evensen, 2003]. Then

$$\begin{aligned} \hat{\mathbf{T}}\hat{\mathbf{T}}^{\mathsf{T}} &= \mathbf{I} - \mathbf{S}^{\mathsf{T}}\mathbf{U}_F \boldsymbol{\Sigma}_F^{-1} (\boldsymbol{\Sigma}_F^{-1})^{\mathsf{T}} \mathbf{U}_F^{\mathsf{T}} \mathbf{S} \\ &= \mathbf{I} - \mathbf{G}_F \boldsymbol{\Gamma}_F \mathbf{Z}_F (\mathbf{G}_F \boldsymbol{\Gamma}_F \mathbf{Z}_F)^{\mathsf{T}} \\ &= \mathbf{I} - \mathbf{G}_F \boldsymbol{\Gamma}_F \boldsymbol{\Gamma}_F^{\mathsf{T}} \mathbf{G}_F^{\mathsf{T}} \\ &= \mathbf{G}_F \left(\mathbf{I} - \boldsymbol{\Gamma}_F \boldsymbol{\Gamma}_F^{\mathsf{T}} \right) \mathbf{G}_F^{\mathsf{T}}, \end{aligned} \tag{2.66}$$

where we have used SVD decomposition of $\mathbf{S}^{\mathsf{T}}\mathbf{U}_F \boldsymbol{\Sigma}_F^{-1} = \mathbf{G}_F \boldsymbol{\Gamma}_F \mathbf{Z}_F$. Thus the ensemble analysis perturbations are given by

$$\mathbf{X'}^a = \mathbf{X'}^f \mathbf{G}_F \left(\mathbf{I} - \boldsymbol{\Gamma}_F \boldsymbol{\Gamma}_F^{\mathsf{T}} \right)^{1/2} \mathbf{G}_F^{\mathsf{T}} \tag{2.67}$$

and the ensemble analysis mean is given by

$$\overline{\mathbf{x}}^a = \overline{\mathbf{x}}^f + \mathbf{X'}^f \mathbf{S}^\mathsf{T} \mathbf{U}_F \mathbf{\Sigma}_F^{-1} (\mathbf{\Sigma}_F^{-1})^\mathsf{T} \mathbf{U}_F^\mathsf{T} (\mathbf{Y} - \mathcal{H}(\mathbf{X}^f)). \tag{2.68}$$

Again we can combine both of the ensemble analysis update steps for the ensemble mean and perturbations into one step as follows,

$$\mathbf{X}^a = \overline{\mathbf{X}}^f + \mathbf{X'}^f \left(\overline{\mathbf{W}}_F + \mathbf{W'}_F \right), \tag{2.69}$$

where $\overline{\mathbf{W}}_F = (\overline{\mathbf{w}}_F, ..., \overline{\mathbf{w}}_F)$ with

$$\overline{\mathbf{w}}_F = \mathbf{S}^\mathsf{T} \mathbf{U}_F \mathbf{\Sigma}_F^{-1} (\mathbf{\Sigma}_F^{-1})^\mathsf{T} \mathbf{U}_F^\mathsf{T} (\mathbf{Y} - \mathcal{H}(\mathbf{X}^f)), \tag{2.70}$$

and

$$\mathbf{W'}_F = \mathbf{G}_F \left(\mathbf{I} - \mathbf{\Gamma}_F \mathbf{\Gamma}_F^\mathsf{T} \right)^{1/2} \mathbf{G}_F^\mathsf{T}. \tag{2.71}$$

Chapter 3

Particle filters

Particle filters, like ensemble methods, are variants of the Monte Carlo methods in which the probability distribution of the model state given some observations is approximated by a number of particles; however, unlike Kalman filter based ensemble methods, particle filters are fully non-linear data assimilation techniques. While particle filters are not a new concept, until very recently they have been deemed to be computationally unfeasible for large-dimensional systems due to the filter degeneracy problem. However, recently there has been a new development in the field and particle filter variants have emerged which have been shown to work for large dimensional systems with a limited number of particles. These methods exploit the future observational information by relaxing particles towards the future observations. In this document we will consider two such variants of the particle filters: the equivalent weights particle filter [Van Leeuwen, 2010, 2011, Ades and van Leeuwen, 2012] and the auxiliary particle filter [Pitt and Shephard, 1999]. Another interesting particle filter for high-dimensional systems, the so called implicit particle filter, is not discussed here as it needs a 4D-Var in each particle.

At a given time k a probability distribution function (pdf) represented by N particles or ensemble members is given by a sum of delta functions centred on the particles,

$$p\left(\mathbf{x}^{(k)}\right) = \frac{1}{N}\sum_{i=1}^{N}\delta\left(\mathbf{x}^{(k)} - \mathbf{x}_i^{(k)}\right), \tag{3.1}$$

where $\mathbf{x}^{(k)} \in \mathcal{R}^n$ is a n-dimensional state of the system that has been integrated forward using the stochastic forward model over k time steps starting at time zero. Time k is when we have a current set of observations with the last observation set at time 0. The stochastic forward model between observations $0 \leq m < k$ for each particle $j = 1, ..., N$ is given by

$$\mathbf{x}_j^{(m)} = \mathcal{M}_m\left(\mathbf{x}_j^{(m-1)}\right) + \boldsymbol{\beta}_j^{(m)}, \tag{3.2}$$

where $\boldsymbol{\beta}_j^{(m)} \in \mathcal{R}^n$ are random terms representing the model error distributed according to a given covariance matrix $\mathbf{Q}$ and $\mathcal{M}_m : \mathcal{R}^n \to \mathcal{R}^n$ is the deterministic model from time $m-1$ to m. Thus model state transition from time $m-1$ to m is fully described by the transition density given by

$$p\left(\mathbf{x}_j^{(m)}|\mathbf{x}_j^{(m-1)}\right) = \mathcal{N}\left(\mathcal{M}_m\left(\mathbf{x}^{(m-1)}\right), \mathbf{Q}\right). \tag{3.3}$$

However, instead of (3.2) and (3.3) a modified forward model is also used to nudge particles towards the future observations and is given by

$$\mathbf{x}_j^{(m)} = \mathcal{M}_m\left(\mathbf{x}_j^{(m-1)}\right) + \tilde{\boldsymbol{\beta}}_j^{(m)} + \boldsymbol{\Upsilon}\left[\mathbf{y}^{(k)} - \mathcal{H}_k\left(\mathbf{x}_j^{(m-1)}\right)\right], \tag{3.4}$$

where $\tilde{\boldsymbol{\beta}}_j^{(m)} \in \mathcal{R}^n$ are random terms representing the model error distributed according to a given covariance matrix $\tilde{\mathbf{Q}}$, $\mathcal{M}_m$ is the same deterministic model as in equation (3.2), $\boldsymbol{\Upsilon}$ is a relaxation matrix which we will chose later, $\mathbf{y}^{(k)} \in \mathcal{R}^{p_k}$ is the vector of p_k observations at time k and $\mathcal{H}_k : \mathcal{R}^n \to \mathcal{R}^{p_k}$ is the observation operator mapping model space in to observation space. Note that the observations $\mathbf{y}^{(k)}$ are at later time $k \geq m$. The model error covariance matrices are usually assumed to be equal, i.e. $\tilde{\mathbf{Q}} = \mathbf{Q}$. The modified transition density is given by

$$q\left(\mathbf{x}_j^{(m)}|\mathbf{x}_j^{(m-1)}, \mathbf{y}^{(k)}\right) = \mathcal{N}\left(\mathcal{M}_m\left(\mathbf{x}^{(m-1)}\right) + \boldsymbol{\Upsilon}\left[\mathbf{y}^{(k)} - \mathcal{H}_k\left(\mathbf{x}^{(m-1)}\right)\right], \mathbf{Q}\right). \tag{3.5}$$

Using Bayes theorem and the Markovian property of the model, the full posterior at observation time k is written as

$$p\left(\mathbf{x}_j^{(k)}|\mathbf{y}^{(k)}\right) = \sum_{j=1}^{N} w_j^{(k)} \delta\left(\mathbf{x}^{(k)} - \mathbf{x}_j^{(k)}\right) \tag{3.6}$$

where the weights w_j^k are given by

$$w_j^{(k)} \propto p\left(\mathbf{y}^{(k)}|\mathbf{x}_j^{(k)}\right) \frac{p\left(\mathbf{x}_j^{(k)}|\mathbf{x}_j^{(k-1)}\right)}{q\left(\mathbf{x}_j^{(k)}|\mathbf{x}_j^{(k-1)}, \mathbf{y}^{(k)}\right)} w_j^{(k-1)} \tag{3.7}$$

and w_j^{k-1} are all the weights from all time steps up to time $k-1$, i.e. fractions $\frac{p\left(\mathbf{x}_j^{(m)}|\mathbf{x}_j^{(m-1)}\right)}{q\left(\mathbf{x}_j^{(m)}|\mathbf{x}_j^{(m-1)}, \mathbf{y}^{(k)}\right)}$ for each time $0 \leq m < k$. The conditional pdf $p\left(\mathbf{y}^{(k)}|\mathbf{x}^{(k)}\right)$ is the pdf of the observations given the model state $\mathbf{x}^{(k)}$ which is often taken to be Gaussian

$$p\left(\mathbf{y}^{(k)}|\mathbf{x}^{(k)}\right) = B \exp\left[-\frac{1}{2}\left(\mathbf{y}^{(k)} - \mathcal{H}_k\left(\mathbf{x}^{(k)}\right)\right)^{\mathsf{T}} \mathbf{R}^{-1}\left(\mathbf{y}^{(k)} - \mathcal{H}_k\left(\mathbf{x}^{(k)}\right)\right)\right]. \tag{3.8}$$

3.1 The equivalent-weights scheme

In the equivalent weights scheme [Van Leeuwen, 2010, Ades and van Leeuwen, 2012] we aim to ensure that equally significant particles are picked from the posterior density. In this scheme we perform the following steps:

1. Before observation time k for each time step $0 \leq m < k$ and for each particle $j = 1, ..., N$:

   (a) Advect the model state in time

$$\mathbf{x}_j^{(m)} = \mathcal{M}_m\left(\mathbf{x}_j^{(m-1)}\right) + \tilde{\boldsymbol{\beta}}_j^{(m)} + \boldsymbol{\Upsilon}\left[\mathbf{y}^{(k)} - \mathcal{H}_k\left(\mathbf{x}_j^{(m-1)}\right)\right] \tag{3.9}$$

where $\boldsymbol{\Upsilon} = \mathbf{Q}(\mathbf{H}^{(k)})^{\mathsf{T}}\left[\mathbf{H}^{(k)}\mathbf{Q}(\mathbf{H}^{(k)})^{\mathsf{T}} + \mathbf{R}\right]^{-1}$ with $\mathbf{Q}$ being the covariance matrix of model errors, $\mathbf{R}$ the covariance matrix of observation errors, and $\mathbf{H}^{(k)}$ the linearised version of the observation operator $\mathcal{H}_k$.

(b) Compute weights

$$w_j^{(m)} = w_j^{(m-1)} \frac{p\left(\mathbf{x}_j^{(m)}|\mathbf{x}_j^{(m-1)}\right)}{q\left(\mathbf{x}_j^{(m)}|\mathbf{x}_j^{(m-1)}, \mathbf{y}^{(k)}\right)}, \tag{3.10}$$

where both transition densities are assumed to be Gaussian and are calculated according to

$$\begin{aligned} p\left(\mathbf{x}_j^{(m)}|\mathbf{x}_j^{(m-1)}\right) &= \exp\left[-\frac{1}{2}\left(\boldsymbol{\Upsilon}\left[\mathbf{y}^{(k)} - \mathbf{H}^{(k)}\mathbf{x}_j^{(m)}\right] + \tilde{\boldsymbol{\beta}}_j^{(m)}\right)^{\mathsf{T}}\mathbf{Q}^{-1}\right. \\ &\quad \left.\left(\boldsymbol{\Upsilon}\left[\mathbf{y}^{(k)} - \mathbf{H}^{(k)}\mathbf{x}_j^{(m)}\right] + \tilde{\boldsymbol{\beta}}_j^{(m)}\right)\right] \qquad (3.11) \\ q\left(\mathbf{x}_j^{(m)}|\mathbf{x}_j^{(m-1)}, \mathbf{y}^{k}\right) &= \exp\left[-\frac{1}{2}\left(\tilde{\boldsymbol{\beta}}_j^{(m)}\right)^{\mathsf{T}}\mathbf{Q}^{-1}\tilde{\boldsymbol{\beta}}_j^{(m)}\right]. \qquad (3.12) \end{aligned}$$

2. At observation time k:

(a) Calculate the maximum weight value for each particle

$$\begin{aligned} C_j &= -\log w_j^{(k-1)} + \frac{1}{2}\left[\mathbf{y}^{(k)} - \mathcal{H}_k\left(\mathcal{M}_k\left(\mathbf{x}_j^{(k-1)}\right)\right)\right]^{\mathsf{T}} \\ &\quad \left[\mathbf{H}^{(k)}\mathbf{Q}\left(\mathbf{H}^{(k)}\right)^{\mathsf{T}} + \mathbf{R}\right]^{-1}\left[\mathbf{y}^{(k)} - \mathcal{H}_k\left(\mathcal{M}_k\left(\mathbf{x}_j^{(k-1)}\right)\right)\right]. \end{aligned}$$

Then choose a target weight C such that 80% (or any other suitable percentage) of particles can reach this weight, i.e. that 80% of C_j are less than C.

(b) Find the deterministic particle analysis update (for the particles which can reach the target weight C) via

$$\check{\mathbf{x}}_j^{(k)} = \mathcal{M}_k\left(\mathbf{x}_j^{(k-1)}\right) + \alpha_j \boldsymbol{\Upsilon}\mathbf{d}_j^{(k)}, \tag{3.13}$$

where

$$\begin{aligned} \boldsymbol{\Upsilon} &= \mathbf{Q}\left(\mathbf{H}^{(k)}\right)^{\mathsf{T}}\left[\mathbf{H}^{(k)}\mathbf{Q}\left(\mathbf{H}^{(k)}\right)^{\mathsf{T}} + \mathbf{R}\right]^{-1} && (3.14) \\ \alpha_j &= 1 - \sqrt{1 - \frac{b_j}{a_j}} && (3.15) \\ a_j &= \frac{1}{2}\left(\mathbf{d}_j^{(k)}\right)^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}^{(k)}\boldsymbol{\Upsilon}\mathbf{d}_j^{(k)} && (3.16) \\ b_j &= \frac{1}{2}\left(\mathbf{d}_j^{(k)}\right)^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{d}_j^{(k)} - C - \log w_j^{(k-1)} && (3.17) \\ \mathbf{d}_j^{(k)} &= \mathbf{y}^{(k)} - \mathcal{H}^{(k)}\left(\mathbf{x}_j^{(k)}\right). && (3.18) \end{aligned}$$

(c) Perturb each particle with random perturbations

$$\mathbf{x}_j^{(k)} = \check{\mathbf{x}}_j^{(k)} + \mathbf{d}\boldsymbol{\beta}_j^{(k)} \tag{3.19}$$

where the perturbation $\mathbf{d}\boldsymbol{\beta}_j^{(k)}$ is drawn from a mixture of uniform and Gaussian distributions, given by

$$\mathbf{d}\boldsymbol{\beta} \sim (1-\epsilon)\mathbf{Q}^{1/2}\mathcal{U}(-\gamma_U\mathbf{I}, +\gamma_U\mathbf{I}) + \epsilon\mathcal{N}(0, \gamma_N^2\mathbf{Q}). \tag{3.20}$$

Choosing $\epsilon = 0.001/N$ ensures that we mainly sample from the uniform distribution, but the possibility to sample from the Gaussian distribution ensures the support of the proposal density is at least equal to the support of the model prior. Other parameters are chosen to be as follows:

$$\gamma_U = 10^{-5} \tag{3.21}$$

$$\gamma_N = \frac{2^{n/2}\epsilon\gamma_U^n}{\pi^{n/2}(1-\epsilon)}. \tag{3.22}$$

(d) Calculate the full weights at time k

$$w_j^{(k)} = w_j^{(k-1)} \frac{p\left(\mathbf{x}_j^{(k)}|\mathbf{x}_j^{(k-1)}\right) p\left(\mathbf{y}^{(k)}|\mathbf{x}_j^{(k)}\right)}{q\left(\mathbf{x}_j^{(k)}|\mathbf{x}_j^{(k-1)}, \mathbf{y}^{(k)}\right)}. \tag{3.23}$$

taking the final perturbation into the account using the transition density

$$q\left(\mathbf{x}_j^{(k)}|\mathbf{x}_j^{(k-1)}, \mathbf{y}^{(k)}\right) = (1-\epsilon)\mathbf{Q}^{1/2}\mathcal{U}(-\gamma_U\mathbf{I}, +\gamma_U\mathbf{I}) + \epsilon\mathcal{N}(0, \gamma_N^2\mathbf{Q}). \tag{3.24}$$

(e) Resample to obtain a full ensemble again, e.g. using universal resampling. After resampling the weights of the resampled particles are set to be equal, i.e. $w_j^{(k)} = 1/N$.

3.2 The auxiliary particle filter

The auxiliary particle filter avoids filter degeneracy at time k by performing the weighting and corresponding resampling at a previous observation time $t = 0$.

1. Integrate each particle forward from time 0 to k with simplified dynamics (e.g. no model noise, simpler dynamics)

$$\mathbf{x}_j^{(k)} = \tilde{\mathcal{M}}_m\left(\mathbf{x}_j^{(k-1)}\right), \tag{3.25}$$

where $\tilde{\mathcal{M}}_m$ is a simplified dynamical model at time $0 \leq m \leq k$.

2. Weight each particle with the new observations

$$w_j^{(k)} \propto \tilde{p}\left(\mathbf{y}^{(k)}|\mathbf{x}_j^{(k)}\right). \tag{3.26}$$

These weights are called first stage weights or simulation weights.

3. Resample particles at time 0 with these weights, and use this ensemble as a representation of the proposal density by integrating it forward to k with the full stochastic model

$$\mathbf{x}_j^{(m)} = \mathcal{M}_m\left(\mathbf{x}_j^{(m-1)}\right) + \boldsymbol{\beta}_j^{(m)}, \tag{3.27}$$

   where $\mathcal{M} : \mathcal{R}^n \to \mathcal{R}^n$ is a deterministic model and $\boldsymbol{\beta}$ is a model noise.

4. Reweigh the members with weights

$$w_j^{(k)} = \frac{1}{A}\frac{p\left(\mathbf{y}^{(k)}|\mathbf{x}_j^{(k)}\right)}{\tilde{p}\left(\mathbf{y}^{(k)}|\mathbf{x}_j^{(k)}\right)}, \tag{3.28}$$

   where $A = \overline{w^{(k)}}$ is the normalization factor over all ensemble members at time k.

5. A resampling step can be done but is not necessary since the actual resampling is done at step 3.

It should be noted that $2N$ integrations have to be performed with this method, one ensemble integration to find the proposal, and one for the actual pdf. If adding the stochastic noise is not expansive, step 1. can be done with the stochastic model, which comes down to doing the Sequential Importance Sampling twice. One can imagine to do it even more times, zooming in on the likelihood, but at a cost of performing more and more integrations of the model.

Chapter 4

Adaptive Gaussian mixture filter

In Gaussian mixture filters [Hoteit et al., 2008] the prior distribution is approximated by a mixture density [Silverman, 1986] where each ensemble member forms the centre of a Gaussian density function. The mixture density including weights is propagated through the dynamical system and updated according to Bayes' rule on the arrive of new observations [Stordal et al., 2011].

In a Gaussian mixture filter we approximate the density function by

$$p(\mathbf{x}) = \sum_{j=1}^{N} w_j \phi(\mathbf{x} - \mathbf{x}_j, \mathbf{P}), \tag{4.1}$$

where $\{\mathbf{x}_j\}_{j=1}^N$ is a sample of N particles (ensemble members), $\phi(\mathbf{x}, \mathbf{P})$ denotes a zero mean multivariate Gaussian kernel density with covariance matrix $\mathbf{P}$, and $\{w_j\}_{j=1}^N$ are scalar weights so that $\sum_{j=1}^N w_j = 1$. Note that each particle represents the mean of a Gaussian kernel and that the uncertainty associated with each particle is given by the covariance of that Gaussian kernel [Stordal et al., 2011].

Differently to the EnKF and similarly to the SEIK filter in the Gaussian mixture filter the analysis step is performed in the error subspace by defining a matrix

$$\tilde{\mathbf{T}} = \begin{bmatrix} \mathbf{I}_{N-1\times N-1} \\ \mathbf{0}_{1\times N-1} \end{bmatrix} - \frac{1}{N-1}\left[\mathbf{1}_{N\times N-1}\right], \tag{4.2}$$

where $\mathbf{0}$ is a matrix whose elements are equal to zero and $\mathbf{1}$ is a matrix whose elements are equal to one [Nerger et al., 2012]. Thus applying the matrix $\tilde{\mathbf{T}}$ to the ensemble matrix, denoted as $\mathbf{X} = \{\mathbf{x}_j\}_{j=1}^N$,

$$\mathbf{L} = \mathbf{X}\tilde{\mathbf{T}} \tag{4.3}$$

implicitly subtracts the ensemble mean and in addition removes the last column of $\mathbf{X}$, thus $\mathbf{L}$ is an $N \times N-1$ matrix that holds the first $N-1$ ensemble perturbations. Thus in the Gaussian mixture filter the ensemble covariance $\mathbf{P}$ is defined as

$$\mathbf{P} = h^2 \mathbf{L}\mathbf{U}\mathbf{L}^{\mathsf{T}}, \tag{4.4}$$

where $\mathbf{U} = \left(\tilde{\mathbf{T}}^{\mathsf{T}}\mathbf{W}^{-1}\tilde{\mathbf{T}}\right)^{-1}$ is the matrix of the transformed particle weights and h is a small tuning parameter called bandwidth. The optimal choice of the bandwidth h is $h_{opt} \sim N^{-1/5}$ if we are only interested in the marginal properties of the

individual components of $\mathbf{x}$; however, since the choice of the bandwidth parameter determines the magnitude of the Kalman update step, it might be beneficial to choose $h > h_{opt}$ to reduce the risk of filter divergence [Stordal et al., 2011]. Thus the parameter h is treated as the design parameter and is defined by the user.

To solve the problem defined by equations (1.1) and (1.2) we perform the following steps:

1. At initial time $m = 0$
   (a) Choose the bandwidth parameter h and the ensemble size threshold N_c.
   (b) Draw N independent identically distributed (iid) particles, $\left\{\mathbf{x}_j^{a,(0)}\right\}_{j=1}^{N}$, from the initial density $p\left(\mathbf{x}^{(0)}\right)$.
   (c) Set initial weights to be
   $$\mathbf{w}^{(0)} = \left\{w_j^{(0)}\right\}_{j=1}^{N} = 1/N$$
   or in matrix form $\mathbf{W}^0 = \text{diag}(\mathbf{w}^0)$ where weights are on the diagonal of the $\mathbf{W}^{(0)}$ matrix with other entries in the matrix being zero.
   (d) Obtain ensemble perturbations by transforming the ensemble members
   $$\mathbf{L}^{a,(0)} = \mathbf{X}^{a,(0)}\tilde{\mathbf{T}},$$
   where $\tilde{\mathbf{T}}$ is defined as in eq. (4.2) and obtain transformed weights from
   $$\mathbf{U}^{a,(0)} = \left(\tilde{\mathbf{T}}^{\mathsf{T}}\left(\mathbf{W}^{(0)}\right)^{-1}\tilde{\mathbf{T}}\right)^{-1}.$$
   (e) Construct the ensemble covariance matrix
   $$\mathbf{P}^{(0)} = h^2\mathbf{L}^{a,(0)}\mathbf{U}^{a,(0)}\left(\mathbf{L}^{a,(0)}\right)^{\mathsf{T}},$$
   where h is the bandwidth parameter.
2. Time evolution, $m > 0$,
   (a) Forward the ensemble in time to the next time the observations are available
   $$\mathbf{X}^{f,(m)} = \mathcal{M}\left(\mathbf{X}^{a,(m-1)}\right) \tag{4.5}$$
   where $\mathbf{X}^{a,(m-1)}$ represents the ensemble analysis state at previous observation time $m-1$ and $\mathbf{X}^{f,(m)}$ represents ensemble forecast at current observation time m.
   (b) Update ensemble members at time m using the ensemble Kalman filter update equations (for clarity the time index m is dropped unless it is different from m):
   $$\mathbf{L}^f = \mathbf{X}^f\tilde{\mathbf{T}} \tag{4.6}$$
   $$\mathbf{P}^f = \mathbf{L}^f\mathbf{U}^{a,(m-1)}(\mathbf{L}^f)^{\mathsf{T}} \tag{4.7}$$
   $$\mathbf{K} = \mathbf{P}^f\mathbf{H}^{\mathsf{T}}(\mathbf{H}\mathbf{P}^f\mathbf{H}^{\mathsf{T}} + \mathbf{R})^{-1} \tag{4.8}$$
   $$\mathbf{X}^a = \mathbf{X}^f + \mathbf{K}\left(\mathbf{y}\mathbf{1}_{p\times 1} - \mathcal{H}(\mathbf{X}^f)\right) \tag{4.9}$$

(c) Update ensemble covariance matrix at time m using the ensemble Kalman filter update equations (for clarity the time index m is dropped):

$$\mathbf{L}^a = \mathbf{X}^a\tilde{\mathbf{T}} \tag{4.10}$$

$$\overline{\mathbf{U}} = \left(\mathbf{U}^{a,(m-1)} + (\mathbf{HL})^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{HL}\right)^{-1} \tag{4.11}$$

$$\mathbf{B} = \mathbf{I}_{N-1} + \overline{\mathbf{U}}(\mathbf{HL}^f)^{\mathsf{T}}\mathbf{R}^{-1}\left[\mathbf{y1} - \mathbf{HX}^f\right]\tilde{\mathbf{T}} \tag{4.12}$$

$$\mathbf{U}^a = \left(\mathbf{B}^{\mathsf{T}}\overline{\mathbf{U}}^{-1}\mathbf{B}\right)^{-1} \tag{4.13}$$

$$\mathbf{P}^a = \mathbf{L}^a\mathbf{U}^a(\mathbf{L}^a)^{\mathsf{T}}. \tag{4.14}$$

(d) Update the weights for each particle $j = 1, ..., N$ at time m:

$$\tilde{w}_j^{(m)} = w_j^{(m-1)} \exp\left(-\frac{1}{2}\left[\mathbf{y}^{(m)} - \mathbf{H}^{(m)}\mathbf{x}_j^{(m)}\right]^{\mathsf{T}} (\mathbf{H}^{(m)}\mathbf{P}^{f,(m)}\mathbf{H}^{(m)} + \mathbf{R}^{(m)})^{-1}\left[\mathbf{y}^{(m)} - \mathbf{H}^{(m)}\mathbf{x}_j^{(m)}\right]\right) \tag{4.15}$$

(e) Normalise the weights

$$w_j^{(m)} = \frac{\tilde{w}_j^{(m)}}{\sum_{i=1}^N \tilde{w}_i^{(m)}}. \tag{4.16}$$

(f) Until above step the filter described here is the standard Gaussian mixture filter. The adaptive part of the filter is included from Stordal et al. [2011], which has been shown to avoid the filter divergence due to collapse thus allowing to choose smaller values of the bandwidth parameter h. However, if the standard version of the method is required then this step should be ignored.
In the adaptive Gaussian mixture filter we interpolate the original analysis weights in equation (4.16) with a uniform weight as follows,

$$w_j^\alpha = \alpha w_j + (1-\alpha)N^{-1}, \tag{4.17}$$

where $\alpha = N_{eff}N^{-1}$ is the adaptive parameter and $N_{eff} = \frac{1}{\sum_{l=1}^N (w_l)^2}$ is the effective ensemble size.

(g) Resample. If $N_{eff} < N_c$[1]

i. Sample new set of particle indices in array I from the weights using one of the resampling methods in Appendix **A**.

ii. Draw a random number $\xi_j \sim \mathcal{N}(0,1)$ for $j = 1, ..., N$.

iii. Sample new particles $\mathbf{x}_j^{a,(m)} = \mathbf{x}_{I(j)}^{a,(m)} + \left(\mathbf{P}^{a,(m)}\right)^{1/2}\xi_j$.

iv. Reset the weights $\mathbf{W}^{(m)} = \mathbf{W}^{(0)}$ to the weights at the initial time.

[1] If using standard Gaussian mixture filter then define the effective ensemble size as $N_{eff} = \frac{1}{\sum_{l=1}^N (w_l)^2}$.

If $N_{eff} \geq N_c$ then don't perform resampling. The particles remain unchanged $\mathbf{X}^{a,(m)}$ and weights are reset to the initial weights $\mathbf{W}^{(m)} = \mathbf{W}^{(0)}$. Note that, since the results of the method will be affected by how often resampling is performed, resampling could be done at each assimilation time or any other time frequency; also the ensemble number threshold N_c could be made time dependent [Stordal et al., 2011].

Chapter 5

Four dimensional ensemble methods

Recently there has been a lot of attention given to fusing variational and ensemble methods to overcome shortcomings of each of the methods individually and to combine their benefits; comparisons of variational or ensemble methods can be found for example in Lorenc [2003], Kalnay et al. [2007], Fairbairn et al. [2014]. The main benefits of using both ensemble and variational frameworks in one method are that background error covariances are flow dependent (and in hybrid methods full rank), localization is done correctly in the state space, combined methods is more robust than pure EnKF for limited ensemble sizes and large model errors, see Buehner et al. [2010a,b], Fairbairn et al. [2014] for tests and comparisons of the different methods.

Currently there is the alpha control variable method, ensemble smoother methods and ensemble of data assimilation methods (EDA). Alpha control variable method [Lorenc, 2003, Buehner, 2005, Wang et al., 2008a,b] is a hybrid method (we follow recommendations of Lorenc [2013] on referring to EnVar DA methods) concerned with incorporating ensemble perturbations directly into variational cost function through extended control variable thus introducing a flow dependence into the static variational background error covariances; this is done using "errors of the day" from an ensemble. Still in its core alpha control variable method remains a variational method and requires the adjoint model. Ensemble smoothers [Van Leeuwen and Evensen, 1996, Evensen and Van-Leeuwen, 2000] use EnKF to obtain priors of the ensemble solution and covariances and use them to compute the smoothed solution with either an EnKF method or variational minimization. EDA methods couple ensemble and variational methods (mainly 4DVar) without the need of adjoint models. EDA methods have started to develop in atmospheric and oceanographic sciences only in the last 10 years, with the maximum likelihood ensemble filter (MLEF) [Zupanski, 2008], 4DEnKF [Hunt et al., 2004, Fertig et al., 2007] and the four-dimensional ensemble-based variational data assimilation method (4DEnVar) [Liu et al., 2008, Liu and Xiao, 2013]. Since Sangoma project is concerned with data assimilation methods which do not involve adjoints and tangent linear models, in this chapter we will give an overview of three such methods, the ensemble smoother (ES), ensemble Kalman smoother (EnKS) and a variant of EDA, the 4DEnVar method.

5.1 Ensemble Smoother

Ensemble Smoother (ES) was proposed by Van Leeuwen and Evensen [1996] and also described in Evensen [2007]. ES computes the analysis using all of the available observations over a time window $t_{0,\ldots,I}$; it finds an approximate update using the usual EnKF update equation

$$\tilde{\mathbf{X}}_k^a = \tilde{\mathbf{X}}_k^f + \tilde{\mathbf{K}}\left(\tilde{\mathbf{Y}} - \mathbf{H}\tilde{\mathbf{X}}_k^f\right)^{1}, \tag{5.1}$$

where the ensemble forecast matrix of the joint state from t_0 to t_I is given by

$$\tilde{\mathbf{X}}_I^f = \begin{pmatrix} \mathbf{X}_0 \\ \mathbf{X}_1^f \\ \ddots \\ \mathbf{X}_{m(I)}^f \end{pmatrix}, \tag{5.2}$$

and the corresponding space-time ensemble covariance matrix is given by

$$\tilde{\mathbf{P}}^f = \frac{\tilde{\mathbf{X}}'^f \left(\tilde{\mathbf{X}}'^f\right)^{\mathsf{T}}}{N-1}. \tag{5.3}$$

Kalman gain is given by

$$\tilde{\mathbf{K}} = \tilde{\mathbf{H}}^{\mathsf{T}}\tilde{\mathbf{P}}^f\left(\tilde{\mathbf{H}}^{\mathsf{T}}\tilde{\mathbf{P}}^f\tilde{\mathbf{H}} + \tilde{\mathbf{R}}\right)^{-1}, \tag{5.4}$$

observation matrix $\tilde{\mathbf{Y}}$, holding all observations over the time window, is given by

$$\tilde{\mathbf{Y}} = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_I \end{bmatrix}, \tag{5.5}$$

observation operator over the time window is given by

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_I \end{bmatrix}, \tag{5.6}$$

and observation error covariance matrix over the time window is given by

$$\tilde{\mathbf{R}} = \begin{bmatrix} \mathbf{R}_1 & & \\ & \ddots & \\ & & \mathbf{R}_I \end{bmatrix}. \tag{5.7}$$

Defining innovation matrix as

$$\mathbf{D} = \tilde{\mathbf{Y}} - \tilde{\mathbf{H}}\tilde{\mathbf{X}}_I^{f\,2}, \tag{5.8}$$

[1] Note, all the variables with $\tilde{}$ are concatenated space-time variables.

[2] $\tilde{\mathbf{X}}_I^f$ is a matrix holding ensemble forecasts at the end of time window t_I.

with associated innovation covariance matrix given by

$$\tilde{\mathbf{F}} = \tilde{\mathbf{S}}\tilde{\mathbf{S}}^{\mathsf{T}} + (N-1)\tilde{\mathbf{R}}, \tag{5.9}$$

the measurements of the ensemble perturbations as

$$\tilde{\mathbf{S}} = \tilde{\mathbf{H}}\tilde{\mathbf{X}'}^{f}_{I}, \tag{5.10}$$

we can write the analysis in 5.1 as

$$\tilde{\mathbf{X}}^{a}_{I} = \tilde{\mathbf{X}}^{f}_{I}\tilde{\mathbf{W}}_{ES} \tag{5.11}$$

where the weight matrix is given by $\tilde{\mathbf{W}}_{ES} = \left(\mathbf{I} + \tilde{\mathbf{S}}^{\mathsf{T}}\tilde{\mathbf{F}}^{-1}\tilde{\mathbf{D}}\right)$ and find the solution using one of the forms of the square root filter defined in Chapter **??**.

5.2 Ensemble Kalman Smoother

The Ensemble Smoother in the previous section used all available observations in the assimilation window to compute the analysis, an alternative approach presented in Evensen and Van-Leeuwen [2000] is to assimilate observations sequentially in time, this methods is called The Ensemble Kalman Smoother (EnKS). There are various ways to implement EnKS and here we follow the description presented in Evensen [2007].

Starting from the initial ensemble $\mathbf{X}_0$ at the beginning of the data assimilation window, t_0, we advect the ensemble forward in time using the dynamical model equations in 1.1 to first set of observations at time $t_{m(1)}$, where $m = 0, ..., L$ is the forward model time level index, L is the total number of forward model time levels and additionally $k = 1, ..., I$ is the observation index with I being the total number of time levels at which observations are available. We then concatenate all the ensemble forecasts from initial time to the time of the first observation

$$\tilde{\mathbf{X}}^{f}_{1} = \begin{pmatrix} \mathbf{X}_0 \\ \mathbf{X}^{f}_{1} \\ \ddots \\ \mathbf{X}^{f}_{m(1)} \end{pmatrix}. \tag{5.12}$$

Using the ES update 5.11 with 5.12 and using the first set of observations $\mathbf{y}_1$ under the assumption of a Gaussian pdf for the predicted ensemble we get

$$\tilde{\mathbf{X}}^{a}_{m,(1)} = \tilde{\mathbf{X}}^{f}_{m(1)}\tilde{\mathbf{W}}_1, \tag{5.13}$$

where as in the ES we have

$$\tilde{\mathbf{W}}_1 = \left(\mathbf{I} + \tilde{\mathbf{S}}^{\mathsf{T}}\tilde{\mathbf{F}}^{-1}\tilde{\mathbf{D}}\right) \tag{5.14}$$

$$\tilde{\mathbf{S}}_1 = \tilde{\mathbf{H}}_1\tilde{\mathbf{X}'}^{f}_{m(1)} \tag{5.15}$$

$$\tilde{\mathbf{F}}_1 = \tilde{\mathbf{S}}_1\tilde{\mathbf{S}}_1^{\mathsf{T}} + (N-1)\tilde{\mathbf{R}}_{m(1)} \tag{5.16}$$

$$\tilde{\mathbf{D}}_1 = \tilde{\mathbf{Y}}_1 - \tilde{\mathbf{H}}_1\tilde{\mathbf{X}}^{f}_{m(1)}. \tag{5.17}$$

The update in 5.11 is identical to the ES update in the case where the time interval covers $t \in [t_0, t_1]$, and the data are all contained in $\mathbf{y}_1$. In the EnKS $\tilde{\mathbf{X}}_1^a$ serves as the prior for the continued ensemble integration until the next observation time. Hence, at time t_m assimilating k^{th} observation set we define the ensemble forecast matrix as

$$\tilde{\mathbf{X}}^f_{m(k)} = \begin{pmatrix} \tilde{\mathbf{X}}^a_{m(k-1)} \\ \mathbf{X}^f_{m(k-1)+1} \\ \ddots \\ \mathbf{X}^f_{m(k)} \end{pmatrix}, \tag{5.18}$$

where $\tilde{\mathbf{X}}^f_{m(k)}$ contains all ensemble forecasts (and past analysis) from the initial time to the current observation time $t_{m(k)}$. Then the analysis update can again be compute again using 5.13

$$\tilde{\mathbf{X}}^a_{m(k)} = \tilde{\mathbf{X}}^f_{m(k)} \tilde{\mathbf{W}}_k, \tag{5.19}$$

where

$$\tilde{\mathbf{W}}_k = \left(\mathbf{I} + \tilde{\mathbf{S}}_k^{\mathsf{T}} \tilde{\mathbf{F}}_k^{-1} \tilde{\mathbf{D}}_k \right). \tag{5.20}$$

We note, that the ensemble in 5.19 has been update using all previous observations $\mathbf{y}_1.\mathbf{y}_2, \ldots \mathbf{y}_k$.

5.3 4DEnVar

The 4DEnVar as developed by Liu et al. [2008] is a method where the 4D background error covariances are estimated from the EnKF ensemble members to produce a 4D analysis with the variational data assimilation approach, but without the need for tangent linear or adjoint versions of the forecast model Buehner et al. [2010a]. The background perturbations are used to precondition the 4DVar control variables,

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{X'}_b \mathbf{w} \tag{5.21}$$

where $\mathbf{w}$ is the control variable and defining innovations as

$$\mathbf{d} = \mathcal{H}(\mathbf{x}_b) - \mathbf{y}, \tag{5.22}$$

the cost function in control variable space is written as

$$J(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} + \frac{1}{2}\sum_{i=0}^{I} \left(\mathbf{H}_i \mathbf{M}_i \mathbf{X'}_{b,i} \mathbf{w} + \mathbf{d}_i\right)^{\mathsf{T}} \mathbf{R}_i^{-1} \left(\mathbf{H}_i \mathbf{M}_i \mathbf{X'}_{b,i} \mathbf{w} + \mathbf{d}_i\right) \tag{5.23}$$

where I is the total number of time levels at which the observations are available. The gradient of 5.23 is given by

$$\nabla_{\mathbf{w}} J = \mathbf{w} + \sum_{i=0}^{I} \mathbf{X'}_{b,i}^{\mathsf{T}} \mathbf{M}_i^{\mathsf{T}} \mathbf{H}_i^{\mathsf{T}} \mathbf{R}_i^{-1} \left(\mathbf{H}_i \mathbf{M}_i \mathbf{X'}_{b,i} \mathbf{w} + \mathbf{d}_i\right). \tag{5.24}$$

and although 4DEnVar can be implemented by calculating the gradient using 5.24 the tangent linear model $\mathbf{M}$ and its adjoint $\mathbf{M}^{\mathsf{T}}$ have to be employed in the 4DVar minimization. To avoid this the idea of EnKF is adopted and the perturbation in observations space is introduced

$$\mathbf{HMX'}_b \approx \frac{1}{\sqrt{N-1}} \left[\mathcal{H}\left(\mathcal{M}\left(\mathbf{x}_{b1}\right)\right) - \mathcal{H}\left(\mathcal{M}\left(\overline{\mathbf{x}}_b\right)\right), ..., \mathcal{H}\left(\mathcal{M}\left(\mathbf{x}_{bN}\right)\right) - \mathcal{H}\left(\mathcal{M}\left(\overline{\mathbf{x}}_b\right)\right)\right]. \tag{5.25}$$

The gradient of the cost function 5.24 then becomes

$$\nabla_{\mathbf{w}} J = \mathbf{w} + \sum_{i=0}^{I} \left(\mathbf{H}_i \mathbf{M}_i \mathbf{X'}_{b,i}\right)^{\mathsf{T}} \mathbf{R}_i^{-1} \left(\mathbf{H}_i \mathbf{M}_i \mathbf{X'}_{b,i} \mathbf{w} + \mathbf{d}_i\right), \tag{5.26}$$

thus avoiding the adjoint model $\mathbf{M}^{\mathsf{T}}$ by the transformation of the the background error to observations space in 5.25. In eqn. 5.26 the background error in observation space is calculated just once using ensemble forecasts outside the minimzation iteration (this greatly reduces the computational and coding costs in comparison to 4DVar). Further, since $\mathbf{H}_i\mathbf{M}_i\mathbf{X'}_{b,i}$ is a $p_i \times N$ dimensional matrix (where p_i is the number of spatial observations at time i, the calculation of $\mathbf{H}_i\mathbf{M}_i\mathbf{X'}_{b,i}$ or $\left(\mathbf{H}_i\mathbf{M}_i\mathbf{X'}_{b,i}\right)^{\mathsf{T}}$ is not expensive and the reduction of the dimensions in 4DEnVar control variable makes the minimization cost even less Liu et al. [2008].

While the background error covariances in 4DEnVar are fully flow dependent (4DVar background covariances are static at the begining of the data assimilation window), they are not full rank[3] since $\mathbf{P}$ (also known as $\mathbf{B}$ in variational literarure)

$$\mathbf{P} = \mathbf{X'X'}^{\mathsf{T}} \tag{5.27}$$

is estimated using N ensemble memebers and can at most have $N-1$ degrees of freedom. Thus without localization the background error covariances in 4DEnVar are restricted to the low-dimensional subspace spanned by the ensemble members; localization in 4DEnVar can be performed using the Schur operator [Lorenc, 2003, Buehner, 2005] thus reducing the sampling error in the covariances and increasing the rank of the covariances. A special issue is that localisation has to be done in space and time. The latter is usually ignored, but within regions of strong advection the localisation matrix should be advected with the flow too. This is an area of active research.

[3]Note that in the alpha control variable methods the covariance matrices are both flow dependent and full rank as in these methods ensemble perturbations are combined with the static background error covariances used in the variational assimilation.

Appendix A

Resampling methods

In this section we give descriptions of a number of resampling techniques which can be applied to the particle filter and Gaussian mixture filter methods to redistribute the filter thus improving the filter spread. The resampling techniques included here are probabilistic resampling, stochastic universal resampling and residual resampling; however, they are by no means exclusive and other techniques could be used.

A.1 Probabilistic resampling

The probabilistic resampling or the basic random resampling is the most straightforward to implement as we sample directly from the density given by the weights.

Given the weights $\{w_j\}_{j=1}^{N}$ associated with the ensemble of particles, where the sum of weights is equal to one, the total number of particles N and the number of particles to be generated M, we generate an index of the sampled particles using the Algorithm 1.

Algorithm 1 Algorithm of probabilistic resampling

function PR(w, N, M)
  $\hat{w}_1 \leftarrow w_1$
  for $j \leftarrow 2$ to N **do** ▷ compute cumulative weights
    $\hat{w}_j = \sum_{i=1}^{j} w_j$
  end for
  $c \leftarrow 1$
  for $j \leftarrow 1$ to M **do**
    $u \sim \mathcal{U}[0, M]$ ▷ generate a random number
    while $u > \hat{w}_c$ **do**
      $c \leftarrow c + 1$
    end while
    $I_j \leftarrow c$ ▷ assign an index of the sampled particle
    $c \leftarrow 1$
  end for
  return I
end function

The required input for the PR is: $w \in \mathcal{R}^N$ a vector of particle weights, N the total number of particles in the filter, and M the number of particles to be sampled and the method returns an index $I \in \mathcal{R}^M$ which can then be used to select the sampled particles $\mathbf{x}_j^* = \mathbf{x}_{I(j)}$ for $j = 1 : M$.

Note that this scheme introduces sampling noise by drawing M times from a uniform distribution.

A.2 Stochastic universal resampling (SUR)

Stochastic universal resampling also known as systematic resampling performs resampling in the same way as the basic random resampling algorithm except instead of drawing each u_j independently from $\mathcal{U}(0,1)$ for $j = 1, ..., N$, it uses a uniform random number u according to $u \sim \mathcal{U}[0, 1/N]$ and $u_j = u + (j-1)/N$ [Bolić et al., 2003].

Given the weights $\{w_j\}_{j=1}^N$ associated with the ensemble of particles, where the sum of weights is equal to one, the total number of particles N and the number of particles to be generated M, we generate an index of the sampled particles using the Algorithm 2.

Algorithm 2 Algorithm of stochastic universal resampling

function SUR(w, N, M)
  $\hat{w}_1 \leftarrow w_1$
  for $j \leftarrow 2$ to N **do** ▷ compute cumulative weights
    $\hat{w}_j = \sum_{i=1}^{j} w_j$
  end for
  $u \sim \mathcal{U}[0, 1/M]$ ▷ generate a random number
  $c \leftarrow 1$
  for $j \leftarrow 1$ to M **do**
    while $u > \hat{w}_c$ **do**
      $c \leftarrow c + 1$
    end while
    $I_j \leftarrow c$ ▷ assign an index of the sampled particle
    $u \leftarrow u + 1/M$
    $c \leftarrow 1$
  end for
  return I
end function

The required input for the SUR is: $w \in \mathcal{R}^N$ a vector of particle weights, N the total number of particles in the filter, and M the number of particles to be sampled and the method returns an index $I \in \mathcal{R}^M$ which can then be used to select the sampled particles $\mathbf{x}_j^* = \mathbf{x}_{I(j)}$ for $j = 1 : M$.

Note, that this method has a lower sampling noise than probabilistic resampling since only one random variable is drawn.

A.3 Residual resampling (RR)

The RR algorithm samples the particles in two parts; in the first part the number of replications of particles is calculated, but since the method does not guarantee that the number of resampled particles is M, the residual Nr is computed. The second step requires resampling which produces Nr of the final M particles. In Algorithm 3 this is done by SUR, but other resampling technique can be used. The required input for the RR is: $w \in \mathcal{R}^N$ a vector of particle weights, N the total

Algorithm 3 Algorithm of residual resampling

function RR(w, N, M)
  for $j \leftarrow 1$ to N **do**
    $\hat{w}_j \leftarrow \lfloor w_j \cdot M \rfloor$ ▷ the integer part or $w \cdot M$
  end for
  $Nr \leftarrow M$
  $c \leftarrow 1$ ▷ counter
  for $j \leftarrow 1$ to M **do**
    if $\hat{w}_j > 0$ **then**
      $I_{c \text{ to } c+\hat{w}_j} \leftarrow j$ ▷ select the copies of the index to sample ‘
      $c \leftarrow c + \hat{w}_j$
    end if
  end for

  if $Nr > 0$ **then**
    for $j \leftarrow 1$ to N **do**
      $\tilde{w}_j \leftarrow (w_j - \hat{w}_j)/Nr$ ▷ compute residual weights and normalize
    end for
    $IR \leftarrow PR(\hat{w}, N, Nr)$ ▷ sample the additional indices
    $I_{c \text{ to } M} \leftarrow IR$ ▷ store the extra indices at the end of the I array
  end if
  return I
end function

number of particles in the filter, and M the number of particles to be sampled and the method returns an index $I \in \mathcal{R}^M$ which can then be used to select the sampled particles $\mathbf{x}_j^* = \mathbf{x}_{I(j)}$ for $j = 1 : M$. Note, that we used the PR method to obtain an array $IR \in \mathcal{R}^{Nr}$ with the indices of the additional sampled particles, which we then stored in the remaining empty cells of the index array $I \in \mathcal{R}^M$.

Note, that this method reduces the sampling noise, but not as much as the SUR method.

Bibliography

M. Ades and P. J. van Leeuwen. An exploration of the equivalent weights particle filter. *Quarterly Journal of Royal Meteorological Society*, pages n/a–n/a, 2012. doi: $10.1002/\mathrm{qj}.1995$.

J. L. Anderson. An ensemble adjustment kalman filter for data assimilation. *Monthly Weather Review*, **129**:2884–2903, 2001.

T. Bengtsson, C. Snyder, and D. Nychka. Toward a nonlinear ensemble filter for high-dimensional systems. *J. Geophys. Res.*, **108**:8775–8785, 2003.

C. H. Bishop, B. J. Etherton, and S. J. Majumdar. Adaptive sampling with the ensemble transform kalman filter. part i: Theoretical aspects. *Mon. Weather Rev.*, **129**:420–436, 2001.

M. Bolić, P. M. Djurić, and S. Hong. New resampling algorithms for particle filters. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (2003)*, volume **2** of *IEEE*, pages II – 589 – 92, 2003.

M. Buehner. Ensemble-derived stationary and flow-dependent background-error covariances: Evaluation in a quasi-operational nwp setting. *Quarterly Journal of the Royal Meteorological Society*, **131**(607):1013–1043, 2005.

M. Buehner, P. L. Houtekamer, C. Charette, L. M. Herschell, and B. He. Intercomparison of variational data assimilation and the ensemble kalman filter for global deterministic nwp. part i: Description and single-observation experiments. *Mon. Wea. Rev.*, **138**:1550–1566, 2010a.

M. Buehner, P. L. Houtekamer, C. Charette, L. M. Herschell, and B. He. Intercomparison of variational data assimilation and the ensemble kalman filter for global deterministic nwp. part ii: One-month experiments with real observations. *Mon. Wea. Rev.*, **138**:1567–1586, 2010b.

G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte-carlo mothods to forecast error statistics. *J. Geophys. Res.*, **99**:10143–10162, 1994.

G. Evensen. The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, **53**:343–367, 2003.

G. Evensen. *Data assimilation: the ensemble Kalman filter*. Springer, 2007.

G. Evensen and P. J. Van-Leeuwen. An ensemble kalman smoother for nonlinear dynamics. *Mon. Wea. Rev.*, **128**:1852–1867, 2000.

D. Fairbairn, S. R. Pring, A. C. Loren, and I. Roulstone. A comparison of 4dvar with ensemble data assimilation methods. *Quarterly Journal of the Royal Meteorological Society*, **140**(678):281–294, 2014.

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

G. H. Golub and C. F. Van Loan. *Matrix computations*. The Johns Hopkins University Press, 3rd edition, 1996.

I. Hoteit, D.-T. Pham, G. Triantafyllou, and G. Korres. A new approximate solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography. *Monthly Weather Review*, **136**:317–334, 2008.

B. R. Hunt, E . Kalnay, E. J. Kostelich, E. Ott, D. J. Patil, T. Sauer, I. Szunyogh, J. A. Yorker, and A. V. Zimin. Four-dimensional ensemble kalman filtering. *Tellus*, **56A**:273–277, 2004.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the AMSE -Journal of Basic Engineering*, **82**(D):35–45, 1960.

E. Kalnay, H. Li, T. Miyoshi, S. Yang, and J. Ballabrera-Poy. 4d-var or ensemble kalman filter? *Tellus*, 59(A):758–773, 2007.

C. Liu and Q. Xiao. An ensemble-based four-dimensional variational data assimilation scheme. part iii: Antarctic applications with advanced research wrf using real data. *Mon. Wea. Rev.*, **141**(8):2721–2739, 2013.

C. Liu, Q. Xiao, and B. Wang. An ensemble-based four-dimensional variational data assimilation scheme. part i: Technical formulation and preliminary test. *Mon. Wea. Rev.*, **136**:3363–3373, 2008.

D. Livings. Aspects of the ensemble kalman filter. Master's thesis, University of Reading. Department of Mathematics, 2005.

D. Livings, S. L. Dance, and N. K. Nichols. Unbiased ensemble square rot filters. *Physica*, **237**:1021–1028, 2008.

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

A. C. Lorenc. Recommended nomenclature for envar data assimilation methods. http://www.wcrp-climate.org/WGNE/BlueBook/2013/individual-articles/01_Lorenc_Andrew_EnVar_nomenclature.pdf, 2013. Accessed: April 2014.

L. Nerger, T. Janjic, J. Schroter, and W. Hiller. A unification of ensemble square root filters. *Monthly Weather Review*, **140**(7):2335–2345, 2012.

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

D. T. Pham, J. Verron, and L. Gourdeau. Singular evolutive Kalman filters for data assimilation in oceanography. *C. R. Acad. Sci., Ser. II*, 326(4):255–260, 1998.

M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, **94**(446):590–599, 1999.

P. Sakov and P. R. Oke. Implications of the form of the ensemble transformation in the ensemble square root filters. *Monthly Weather Review*, **136**:1042–1053, 2008.

B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, New Work, 1986.

A. S. Stordal, H. A. Karlsen, G. Nævdal, H. J. Skaug, and B. Valles. Bridging the ensemble kalman filter and particle filters: the adaptive gaussian mixture filter. *Computational Geosciences*, **15**(2):293–305, 2011.

M. K. Tippett, J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker. Ensemble square root filters. *Mon. Weath. Rev.*, **131**(17):1485–1490, 2003.

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

P. J. Van Leeuwen. Efficient nonlinear data-assimilation in geophysical fluid dynamics. *Computers and Fluids*, **46**:52–58, 2011.

P. J. Van Leeuwen and G. Evensen. Data assimilation and inverse methods in terms of a probabilistic formulation. *Monthly Weather Review*, 124:2898–2913, 1996.

X. Wang, D. Barker, C. Snyder, and T. M. Hamill. A hybrid etkf-3dvar data assimilation scheme for the wrf model. part i: Observing system simulation experiment. *Mon. Wea. Rev.*, **136**:5116–5131, 2008a.

X. Wang, D. Barker, C. Snyder, and T. M. Hamill. A hybrid etkf-3dvar data assimilation scheme for the wrf model. part ii: Real observation experiments. *Mon. Wea. Rev.*, **136**:5133–5147, 2008b.

J. S. Whitaker and T. M. Hamill. Ensemble data assimilation without perturbed observations. *Mon. Weath. Rev.*, **130**:1913–1924, 2002.

M. Zupanski. Maximum likelihood ensemble filter: Theoretical aspects. *Mon. Wea. Rev.*, **133**:1710–1726, 2008.