Tellus SERIES A DYNAMIC METEOROLOGY AND OCEANOGRAPHY
PUBLISHED BY THE INTERNATIONAL METEOROLOGICAL INSTITUTE IN STOCKHOLM

A weak-constraint 4DEnsembleVar. Part I: formulation and simple model experiments

JAVIER AMEZCUA[1,2]*, MICHAEL GOODLIFF[1] and PETER JAN VAN LEEUWEN[1,2],
[1]*Department of Meteorology, University of Reading, Reading, UK;* [2]*National Centre for Earth Observation, Reading, UK*

(Manuscript received 9 January 2016; in final form 29 November 2016)

ABSTRACT

4DEnsembleVar is a hybrid data assimilation method which purpose is not only to use ensemble flow-dependent covariance information in a variational setting, but to altogether avoid the computation of tangent linear and adjoint models. This formulation has been explored in the context of perfect models. In this setting, all information from observations has to be brought back to the start of the assimilation window using the space-time covariances of the ensemble. In large models, localisation of these covariances is essential, but the standard time-independent localisation leads to serious problems when advection is strong. This is because observation information is advected out of the localisation area, having no influence on the update.

This is part I of a two-part paper in which we develop a weak-constraint formulation in which updates are allowed at observational times. This partially alleviates the time-localisation problem. Furthermore, we provide—for the first time—a detailed description of strong- and weak-constraint 4DEnVar, including implementation details for the incremental form.

The merits of our new weak-constraint formulation are illustrated using the Korteweg-de-Vries equation (propagation of a soliton). The second part of this paper deals with experiments in larger and more complicated models, namely the Lorenz (1996) model and a shallow water equations model with simulated convection.

KEYWORDS: hybrid data assimilation, ensemble-variational methods, model error

1. Introduction

The 4-dimensional ensemble-variational data assimilation (DA) scheme, 4DEnVar, is a hybrid DA method currently used (and still being researched) in Numerical Weather Prediction (NWP), and it is at the forefront of the next-generation DA methods. As with other hybrid methods, the basic motivation behind 4DEnVar is to use the flow-dependent background error covariance matrix from sequential methods based on the Kalman Filter (KF)—like the Ensemble Transform Kalman Filter (ETKF, Bishop et al., 2001; Wang et al., 2004) and the Local Ensemble Transform Kalman Filter (LETKF, Hunt et al., 2007)—and apply it in the 4-dimensional variational framework (4DVar) first proposed by Le Dimet and Talagrand (1986), and studied in Talagrand and Courtier (1987). What makes 4DEnVar different from all other hybrids, however, is the fact that it alleviates the need to compute tangent linear models (TLM's) and adjoint models (AM's).

The use of ensemble information in a variational framework was proposed by Lorenc (2003) and Zupanski (2005). Since then, several formulations have been proposed, experimented on, and used by operational centres like the UK Met Office and the Canadian Meteorological Centre (Environ Canada). The idea of these *hybrid* methods is to overcome certain restrictions of the basic 4DVar, for instance, the use of a static background error covariance matrix, which is usually a climatological error covariance matrix $\mathbf{B}_c$. Although it is possible to generate slowly-varying $\mathbf{B}_c$'s—e.g. difference covariances for different seasons–, the fast variations—e.g. those in the time-frame of an assimilation window—are not captured. Hence, climatological or slow-varying background matrices do not describe the flow-dependent *errors of the day*. Sequential ensemble-based methods, e.g. LETKF, can describe these features. Nonetheless, the sample covariances obtained in these ensemble methods—which we denote as $\mathbf{P}^b$—contain sampling errors and are seldom full rank, which is a problem climatological covariances do not have. It seems logical, then, to find ways to combine the advantages of both families of methods. Clayton et al. (2013) developed a hybrid 4DVar which combines the climatological background error covariance matrix with the flow dependent background error covariance matrix generated by the LETKF. Fairbarn et al. (2014) and Goodliff et al. (2015) then showed that a fully flow dependent background error covariance matrix outperforms a

*Corresponding author. e-mail: j.AmezcuaEspinosa@reading.ac.uk

Citation: Tellus A: 2017, **69**, 1271564, http://dx.doi.org/10.1080/16000870.2016.1271564

climatological or hybrid covariance in the 4DVar framework (4DVar-Ben). The former used model II of Lorenz (2005) with 240 variables, while the latter used the 3-variable Lorenz (1963) model with increasing non-linearity.

A non-trivial requirement for the implementation of 4DVar is generating TLMs and AMs for both the evolution and observation operators. Liu et al. (2008) developed the 4DEnVar technique, which overcomes this need. After a series of matrix transformations, the role of these matrices can be substituted by using four-dimensional space-time covariance matrices. In fact, the idea that an ensemble can be used to generate space-time covariances to use within an assimilation window was first proposed in the context of the Ensemble Kalman Smoother of Van Leeuwen and Evensen (1996) and Evensen and Van Leeuwen (2000).

A recent study by Lorenc et al. (2015) showed difficulties for the performance of 4DEnVar in larger systems, in comparison to that of 4DVar with hybrid background covariances. The authors point to the fact evolution and localisation of the covariance matrix do not commute. This means that it is not the same to evolve a localised a covariance matrix, or to evolve first and then localise.

So far, the work on 4DEnVar has been done considering a perfect model scenario, i.e. in a strong-constraint (SC) setting. In this work, we introduce a weak-constraint (WC) 4DEnVar. As explored by Tremolet (2006), the choice of control variables in WC4DVar is not unique. For reasons that will be thoroughly discussed in the paper, we use what we label an 'effective model error formulation', which stems from Tremolet's (2006) constant bias formulation.

This work is presented in two parts. Part I describes the 4DEnVar formulation in detail. We discuss its advantages over traditional DA methods, as well as its short-comings. We pay special attention to the problem of localising time cross-covariances using static localisation matrices. We illustrate this problem by experimenting with the Korteweg de Vries (KdV) equation for the evolution of a soliton (see e.g. Zakharov and Faddeev, 1971). In part I we also introduce our WC4DEnVar and discuss a proper localisation implementation for this method. We show that performing updates at times other than the initial one (as one does in the SC case), partially alleviates the impact of incorrectly localised time cross-covariances. Simple DA experiments are performed in this model.

In part II we move into larger models. We start with a detailed exploration of parameters (e.g. observation frequency, observation density in space, localisation radii, etc) using the well-known Lorenz (1996) chaotic model. Then we use a modified shallow water equations (SWE) model with simulated convection (Wursch and Craig, 2014), which allows to test our method in a more realistic setting. This is a larger, more non-linear model and serves as a good test bed for convective data assimilation methods.

The layout of this paper is as follows. Sections 2 and 3 describe in details the methodology of the 4DEnVar framework. Section 2 covers the SC case, and in Section 3 we introduce our WC4DEnVar formulation. In Section 4 we use the KdV model and use it to illustrate the time evolution of the background error covariance matrix, as well as the problems that come from static localisation of cross-time covariance matrices. In Section 5 we perform some brief DA experiments. Section 6 summarises the work of this part.

2. The variational methods

2.1. Strong-constraint 4DVar

In this section we formulate the SC4DVar method, which forms the basis of any subsequent approximation. Consider the discrete-time evolution of a dynamical system determined by the following equation:

$$\mathbf{x}^t = m^{0\to t}(\mathbf{x}^0) \tag{1}$$

where $\mathbf{x} \in \mathcal{R}^{N_x}$ is the vector of state variables, and $m^{0\to t}$ is a map $\mathcal{R}^{N_x} \to \mathcal{R}^{N_x}$ which evolves the state variables from time 0 to time t. For the moment, we consider this model to be perfect. The initial condition $\mathbf{x}^0$ is not known with certainty, and can be considered a random variable (rv). In particular, we consider it to be a Gaussian rv, i.e. $\mathbf{x}^0 \sim N(\mathbf{x}^{0,b}, \mathbf{B})$, where $\mathbf{x}^{0,b}$ is a background value or first guess (usually coming from a previously generated forecast), and $\mathbf{B} \in \mathcal{R}^{N_x \times N_x}$ represents the background error covariance matrix, which is full rank and positive definite. One often replaces this matrix with a climatological background error covariance $\mathbf{B}_c$. There are several ways to obtain $\mathbf{B}_c$ it (e.g. Parrish and Derber, 1981; Yang et al., 2006); see e.g. Bannister (2008) for a review.

The system is observed from time to time. For a given observational time, the observation equation is:

$$\mathbf{y}^{o,t} = h(\mathbf{x}^t) + \eta^t \tag{2}$$

where $\mathbf{y}^o \in \mathcal{R}^{N_y}$ is the vector of observations and $h : \mathcal{R}^{N_x} \to \mathcal{R}^{N_y}$ is the observation operator. For simplicity we consider this operator constant through time. We use the term 'observation period' to denote the number of model time steps between consecutive observation vectors. The observations contain errors, which are represented by the rv $\eta^t \in \mathcal{R}^{N_y \times N_y}$. We consider this error to be Gaussian and unbiased: $\eta^t \sim N(\mathbf{0}, \mathbf{R})$, where $\mathbf{R} \in \mathcal{R}^{N_y \times N_y}$ is the observation error covariance matrix.

Let us consider a given time span from $t = 0$ to $t = \tau$, which we label 'assimilation window'. By incorporating information from the background at $t = 0$ and the observations within this window, we can find a better estimate (analysis) for the value of the state variables at every time step $\mathbf{x}^0, \mathbf{x}^1, \cdots, \mathbf{x}^\tau$. Since the model is perfect (SC4DVar), this reduces to finding the minimiser of a cost function that only depends on the initial condition $\mathbf{x}^0$:

$$J(\mathbf{x}^0) = \frac{1}{2}\,(\mathbf{x}^0 - \mathbf{x}^{b,0})^{\mathbf{T}}\mathbf{B}^{-1}(\mathbf{x}^0 - \mathbf{x}^{b,0}) + \frac{1}{2}\sum_{t=1}^{\tau}\rho^t(\mathbf{y}^{o,t} - h(m^{0\to t}(\mathbf{x}^0)))^{\mathbf{T}}\mathbf{R}^{-1}(\mathbf{y}^{o,t} - h(m^{0\to t}(\mathbf{x}^0))) \tag{3}$$

where the first term corresponds to the contribution from the background and the second term corresponds to the contributions from observations. To avoid an extra index to distinguish between time instants with and without observations, we introduce the indicator function $\rho^t = 1$ if t is a time instant with observations, and $\rho^t = 0$ if t is a time instant with no observations.

The complexity in minimising (3) depends on the particular characteristics of the operators h and m. For general non-linear operators, $J(\mathbf{x}^0)$is not convex and may have multiple minima, leading to a difficult minimisation process. Another issue comes from the condition number of $\mathbf{B}$. If this is large (i.e. the largest and smallest eigenvalues of $\mathbf{B}$ are separated by orders of magnitude), numerical minimisation methods can require many iterations for convergence. To solve these problems one can use a preconditioned formulation to reduce the ellipticity of the cost function. Moreover, an incremental formulation can be used to perform a quadratic approximation the original cost function, leading to a problem with a unique minimum. This linearisation process can be iterated several times; these iterations are the so-called outer loops.

To write down the preconditioned incremental problem, let us express the initial condition as an increment $\delta\mathbf{x}^0 \in \mathcal{R}^{N_x}$ from the original background value, and then precondition the increment:

$$\mathbf{x}^0 = \mathbf{x}^{b,0} + \delta\mathbf{x}^0 = \mathbf{x}^{b,0} + \mathbf{B}^{1/2}\mathbf{v}^0 \tag{4}$$

where $\mathbf{v}^0 \in \mathcal{R}^{N_x}$ is the new control variable. Since $\mathbf{B}$ is full rank and symmetric, one can find the (unique) symmetric square root $\mathbf{B}^{1/2} \in \mathcal{R}^{N_x\times N_x}$ without further complications. In this case, $\delta\mathbf{x}^0 \in \mathcal{R}^{N_x}$and $\mathbf{v}^0 \in \mathcal{R}^{N_x}$. Then, one can approximate (3) by performing the following first order Taylor expansion:

$$\mathbf{y}^{o,t} - h(m^{0\to t}(\mathbf{x}^0)) \approx \mathbf{d}^t - \mathbf{H}\mathbf{M}^{0\to t}\mathbf{B}^{1/2}\mathbf{v}^0$$

where

$$\mathbf{d}^t = \mathbf{y}^{o,t} - h(m^{0\to t}(\mathbf{x}^{b,0})) \tag{5}$$

is the departure of the observations with respect to the evolution of the background guess throughout the assimilation window. To compute these departures one uses the full (generally non-linear) evolution and observation operators. We notice that two matrices appear in the equation. These are the TL observation operator

$$\mathbf{H} = \frac{\partial h}{\partial \mathbf{x}} \in \mathcal{R}^{N_y\times N_x} \tag{6}$$

and TLM of the discrete-time evolution operator

$$\mathbf{M}^{0\to t} = \frac{\partial m^{0\to t}}{\partial \mathbf{x}} \in \mathcal{R}^{N_x\times N_x} \tag{7}$$

which is also known as the transition matrix from time 0 to time t. Both Jacobian matrices are evaluated with respect to the reference trajectory, i.e. the evolution of $\mathbf{x}^{b,0}$. Then, the preconditioned incremental form of SC-4DVar can be written as:

$$J(\mathbf{v}^0) = \frac{1}{2}(\mathbf{v}^0)^{\mathbf{T}}\mathbf{v}^0 + \frac{1}{2}\sum_{t=1}^{\tau}\rho^t(\mathbf{d}^t - \mathbf{H}\mathbf{M}^{0\to t}\mathbf{B}^{1/2}\mathbf{v}^0)^{\mathbf{T}}\mathbf{R}^{-1}(\mathbf{d}^t - \mathbf{H}\mathbf{M}^{0\to t}\mathbf{B}^{1/2}\mathbf{v}^0) \tag{8}$$

This is a quadratic equation which only depends on $\mathbf{v}^0$. In order to find its global minimum it suffices to find the gradient and set it equal to zero. The gradient of (8) is:

$$\nabla_{\mathbf{v}^0}J(\mathbf{v}^0) = \mathbf{v}^0 - \sum_{t=1}^{\tau}\rho^t(\mathbf{B}^{1/2})^{\mathbf{T}}(\mathbf{M}^{0\to t})^{\mathbf{T}}\mathbf{H}^{\mathbf{T}}\mathbf{R}^{-1}(\mathbf{d}^t - \mathbf{H}\mathbf{M}^{0\to t}\mathbf{B}^{1/2}\mathbf{v}^0) \tag{9}$$

2.2. *Strong-constraint 4DEnsembleVar*

Writing TLM's and AM's for large and complex models is a complicated process. Moreover, the calculations involving both TLM's and AM's are computationally expensive. To alleviate these problems, 4DEnVar was introduced. Once more, let us consider an assimilation window from $t = 0$ to $t = \tau$, with several observational times equally distributed throughout the assimilation window. Consider that at time t we have an ensemble of N_e forecasts. This ensemble can be initialized at time $t = 0$ with the analysis coming from an LETKF running in parallel to the 4DEnVar system. At any time we can express the background (or forecast) ensemble as the matrix:

$$\mathbf{X}^{b,t} = \left[\mathbf{x}_1^{b,t}, \cdots, \mathbf{x}_{N_e}^{b,t}\right] \in \mathcal{R}^{N_x\times N_e} \tag{10}$$

from which an ensemble mean can be calculated at any time as:

$$\bar{\mathbf{x}}^{b,t} = \frac{1}{N_e}\mathbf{X}^{b,t}\mathbf{1} \in \mathcal{R}^{N_x} \tag{11}$$

where $\mathbf{1} \in \mathcal{R}^{N_e}$ is a vector of ones. Finally, a normalized ensemble of perturbations can be computed as

$$\hat{\mathbf{X}}^{b,t} = \left[\frac{\mathbf{x}_1^{b,t} - \bar{\mathbf{x}}^{b,t}}{\sqrt{N_e - 1}}, \cdots, \frac{\mathbf{x}_M^{b,t} - \bar{\mathbf{x}}^{b,t}}{\sqrt{N_e - 1}}\right] \in \mathcal{R}^{N_x\times N_e} \tag{12}$$

In the EnVar framework we replace the background error covariance $\mathbf{B}$ at the beginning of an assimilation window by the sample estimator:

$$\mathbf{P}^{b,0} = \hat{\mathbf{X}}^{b,0}(\hat{\mathbf{X}}^{b,0})^{\mathbf{T}} \tag{13}$$

In applications (e.g. NWP) it is often the case that $N_e \ll N_x$, so $\mathbf{P}^{b,0}$ is a low rank non-negative definite matrix, which has consequences we discuss later. A straightforward implementation of 4DEnVar is to equate $\mathbf{B}^{1/2} = \hat{\mathbf{X}}^{b,0}$. Then, we can do the following substitution:

$$\mathbf{H}\mathbf{M}^{0\to t}\mathbf{B}^{1/2} \approx \mathbf{H}\mathbf{M}^{0\to t}\hat{\mathbf{X}}^{b,0} \approx \mathbf{H}\hat{\mathbf{X}}^{b,t} \approx \hat{\mathbf{Y}}^{b,t} \tag{14}$$

An advantage of using ensemble estimators is that we can directly construct the ensemble of perturbations $\hat{\mathbf{Y}}^{b,t} \in \mathcal{R}^{N_y \times N_e}$ using the full non-linear operators and the ensemble $\mathbf{X}^{b,t}$ in the way described, e.g., in Hunt et al. (2007). Briefly, one can write:

$$\mathbf{Y}^t = \left[\mathbf{y}_1^t = h(f^{0\to t}(\mathbf{x}_1^0)), \cdots, \mathbf{y}_{N_e}^t = h(f^{0\to t}(\mathbf{x}_{N_e}^0))\right] \tag{15}$$

and it follows that:

$$\hat{\mathbf{Y}}^{b,t} = \left[\frac{\mathbf{y}_1^{b,t} - \bar{\mathbf{y}}^{b,t}}{\sqrt{N_e - 1}}, \cdots, \frac{\mathbf{y}_{N_e}^{b,t} - \bar{\mathbf{y}}^{b,t}}{\sqrt{N_e - 1}}\right] \in \mathcal{R}^{N_y \times N_e} \tag{16}$$

and the gradient (9) can be written simply as:

$$\nabla_{\mathbf{v}^0} J(\mathbf{v}^0) = \mathbf{v}^0 - \sum_{t=1}^{\tau} \rho^t (\hat{\mathbf{Y}}^{b,t})^{\mathbf{T}} \mathbf{R}^{-1} \left(\mathbf{d}^t - \hat{\mathbf{Y}}^{b,t}\mathbf{v}^0\right) \tag{17}$$

and it should be clear that up to this moment we have not applied localisation. Hence the size of the control variable we solve for is $\mathbf{v}^0 \in \mathcal{R}^{N_e}$.

A graphic summary of the SC4DEnVar process is depicted in Fig. 1. This simple schematic depicts the three-stage-process of SC4DEnVar. These are:

i. First an ensemble of 'free' (DA-less) trajectories of the model is run for the length of the assimilation window. In NWP applications these trajectories are available since meteorological centres often have an ensemble forecast system.

ii. Second, the 4DVar minimisation process is performed using 4-dimensional cross-time covariances instead of TLM's and AM's. These cross-time covariances are computed from the family of free model runs of step i.

iii. Finally an LETKF (or any other ensemble KF) is run throughout the assimilation window to create new initial conditions for the next window. The mean of this LETKF is replaced by the solution of 4DEnVar, considered to be more accurate, i.e. the ensemble keeps its perturbations but it is re-centred.In this implementation, two assimilation systems need to be operated and maintained. There are other hybrid systems such as ECMWF's ensemble DA system (Bonavita et al., 2012) which are self-sufficient, but they require the use of TLM's and AM's.

2.2.1. Applying localization. The quality of $\mathbf{P}^{b,t}$ depends on the size of the ensemble. In particular, the quality of this estimator is quite poor when $N_e \ll N_x$, or at least when it is smaller than the number of positive Lyapunov exponents in the system. A common practice is to eliminate spurious correlations by tampering the covariance matrix in the following manner (Whitaker and Hamill, 2002):

$$\mathbf{P}^{b,t} = \hat{\mathbf{X}}^{b,t}(\hat{\mathbf{X}}^{b,t})^{\mathbf{T}} \to \tilde{\mathbf{P}}^{b,t} = \left(\hat{\mathbf{X}}^{b,t}(\hat{\mathbf{X}}^{b,t})^{\mathbf{T}}\right) \circ \mathbf{L}_{\mathbf{xx}} \tag{18}$$

where $\circ$ denotes the Schur (element-wise) product, and $\mathbf{L}_{\mathbf{xx}} \in \mathcal{R}^{N_x \times N_x}$ is the so-called localisation matrix. The element $\{\mathbf{L}_{\mathbf{xx}}\}_{ij}$ is a decreasing function of the physical distance between grid points i and j. This is usually chosen as the

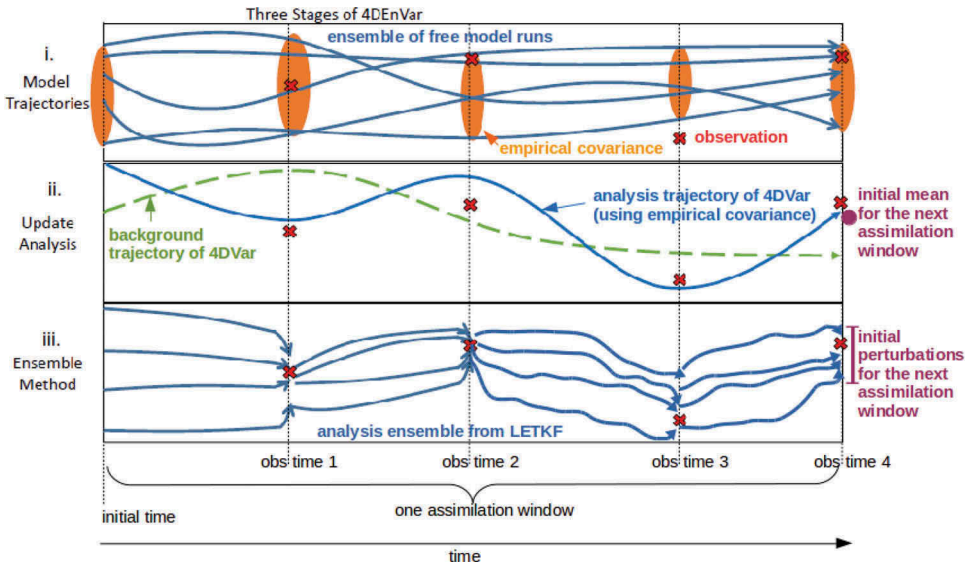

Fig. 1. Schematic depicting the three stage process of 4DEnVar. In part i an ensemble of 'free' (DA-less) trajectories of the model is run for the length of the assimilation window. In part ii the 4DVar minimisation process is performed using 4-dimensional cross-time covariances instead of tangent linear and adjoint models. In part iii an LETKS is run to the end of the assimilation window to create new initial conditions for the next window. The mean of this LETKS is replaced by the solution of 4DEnVar, considered to be more accurate.

Gaspari and Cohn (1999) compact support approximation to a Gaussian function.

In the gradient of the 4DVar cost function the product $\mathbf{BH^T} \in \mathcal{R}^{N_x \times N_y}$ appears. In the ensemble setting, this product corresponds to $\mathbf{P}^{b,t}_{\mathbf{x},\mathbf{y}}$, which can be computed and localised as:

$$\mathbf{P}^{b,t}_{\mathbf{xy}} = \hat{\mathbf{X}}^{b,t}(\hat{\mathbf{Y}}^{b,t})^{\mathbf{T}} \rightarrow \tilde{\mathbf{P}}^{b,t}_{\mathbf{xy}} = \left(\hat{\mathbf{X}}^{b,t}(\hat{\mathbf{Y}}^{b,t})^{\mathbf{T}}\right) \circ \mathbf{L_{xy}} \tag{19}$$

which clearly requires a new matrix $\mathbf{L_{xy}} \in \mathcal{R}^{N_y \times N_x}$. For linear observation operators, the two localisation matrices are related simply as:

$$\mathbf{L_{xy}} = \mathbf{L_{xx}}\mathbf{H^T} \tag{20}$$

In order to write a localised version of (17) we require factorisations of both $\tilde{\mathbf{P}}^{b,t}$ and $\tilde{\mathbf{P}}^{b,t}_{\mathbf{xy}}$. Buehner (2005) showed that one can get the factorisation

$$\left(\hat{\mathbf{X}}^{b,t}(\hat{\mathbf{X}}^{b,t})^{\mathbf{T}}\right) \circ \mathbf{L_{xx}} = \hat{\tilde{\mathbf{X}}}^{b,t}(\hat{\tilde{\mathbf{X}}}^{b,t})^{\mathbf{T}} \tag{21}$$

with

$$\hat{\tilde{\mathbf{X}}}^{b,t} = \left[diag(\mathbf{x}_1^{b,t})\mathbf{L_x}^{1/2}, \cdots, diag(\mathbf{x}_M^{b,t})\mathbf{L_x}^{1/2}\right] \in \mathcal{R}^{N_x \times rN_e} \tag{22}$$

where the operator *diag* converts a vector into a matrix with said vector in its main diagonal and zeros elsewhere, $\mathbf{L_x}^{1/2} \in \mathcal{R}^{N_x \times r}$is the square root of the $\mathbf{L_{xx}}$, and $r = rank\ (\mathbf{L_{xx}})$. If the localisation only depends on the distance amongst grid points (i.e. there is not a special cross-variable localisation or vertical localisation) then $r = N_g$, i.e. simply the number of grid points. Constructing $\mathbf{L_x}^{1/2}$ can be done relatively easily via the following eigenvalue decomposition:

$$\mathbf{L_{xx}} = \mathbf{C}\Gamma\mathbf{C^T} \tag{23}$$

where $\mathbf{C} \in \mathcal{R}^{N_x \times N_x}$ is the unitary matrix of eigenvectors, and $\Gamma \in \mathcal{R}^{N_x \times N_x}$ is a diagonal matrix containing the eigenvalues, and we have used the fact that $\mathbf{L_{xx}}$ is symmetric. Then:

$$\mathbf{L_x}^{1/2} = \mathbf{C}\Gamma^{1/2} \tag{24}$$

If there are many variables per grid point, the decomposition can be performed by blocks. Also, it is often more convenient to work with a truncated localisation matrix with a given rank $r < N_g$. This is discussed thoroughly in Appendix 1.

In the case of $\tilde{\mathbf{P}}^{b,t}_{\mathbf{xy}}$ we need to write

$$\left(\hat{\mathbf{X}}^{b,t}(\hat{\mathbf{Y}}^{b,t})^{\mathbf{T}}\right) \circ \mathbf{L_{xy}} = \hat{\tilde{\mathbf{X}}}^{b,t}(\hat{\tilde{\mathbf{Y}}}^{b,t})^{\mathbf{T}} \tag{25}$$

with $\hat{\tilde{\mathbf{X}}}^{b,t}$ defined as before, and

$$\hat{\tilde{\mathbf{Y}}}^{b,t} = \left[diag(\mathbf{y}_1^{b,t})\mathbf{L_y}^{1/2}, \cdots, diag(\mathbf{y}_M^{b,t})\mathbf{L_y}^{1/2}\right] \tag{26}$$

where $\mathbf{L_y}^{1/2} \in \mathcal{R}^{N_y \times r}$ can easily be found as:

$$\mathbf{L_y}^{1/2} = \mathbf{H}\mathbf{L_x}^{1/2} = \mathbf{H}\mathbf{C}\mathrm{G}^{1/2} \tag{27}$$

The actual implementation we use can be found in Appendix 1. It is closer to Lorenc (2003) and his so-called α-control-variable formulation. Wang et al. (2007) proved that both formulations are equivalent.

2.2.2. Localisation in 4-dimensional cross-time covariances. Recall that we have to compute the product $\mathbf{B}(\mathbf{M}^{0\rightarrow t})^{\mathbf{T}}\mathbf{H^T}$, which we replace by the sample estimator $\hat{\mathbf{X}}^{b,0}(\hat{\mathbf{Y}}^{b,t})^{\mathbf{T}}$. Note that these covariances involve both the ensemble of state variables at time *0*, and the ensemble of equivalent observations at time *t*. Hence, different products should be localised with different localisation matrices:

$$(\hat{\mathbf{X}}^{b,0}(\hat{\mathbf{Y}}^{b,0})^{\mathbf{T}}) \circ \mathbf{L}^{0,0}_{\mathbf{xy}}, (\hat{\mathbf{X}}^{b,0}(\hat{\mathbf{Y}}^{b,1})^{\mathbf{T}}) \circ \mathbf{L}^{0,1}_{\mathbf{xy}}, \cdots, (\hat{\mathbf{X}}^{b,0}(\hat{\mathbf{Y}}^{b,\tau})^{\mathbf{T}}) \circ \mathbf{L}^{0,\tau}_{\mathbf{xy}} \tag{28}$$

Computing the *static* matrix $\mathbf{L}^{0,0}_{\mathbf{xy}}$ is not difficult since it requires no information about the flow. However, there is no simple way of computing an optimal $\mathbf{L}^{0,t}_{\mathbf{xy}}$, since that would require taking into account the dynamics of the model. Generally, the implementation of SC4DEnVar uses $\mathbf{L}^{0,0}_{\mathbf{xy}}$ for all time steps, which can result in an unfavourable performance with respect to SC4DVar (Lorenc et al. 2015). This is a crucial impediment for an adequate performance of SC4DEnVar with long assimilation windows. We will illustrate this issue in detail in Section 4, where we provide a concrete example.

2.3. Weak-constraint 4DVar

Let us revisit the discrete-time dynamical system studied before, but this time not assuming perfect model. The evolution equation becomes:

$$\mathbf{x}^t = m^{(t-1)\rightarrow t}(\mathbf{x}^{t-1}) + v^t \tag{29}$$

In this case $m^{(t-1)\rightarrow t}$ is the map $\mathcal{R}^{N_x} \rightarrow \mathcal{R}^{N_x}$ which evolves the state variables from time $t-1$ to time t, and $v^t \in \mathcal{R}^{N_x}$ is a rv representing the model error. Let it be a Gaussian rv centred in zero (unbiased), i.e. $v^t \sim N(\mathbf{0}, \mathbf{Q})$where $\mathbf{Q} \in \mathcal{R}^{N_x \times N_x}$ is the model error covariance matrix.

Tremolet (2006) provides a detailed account on the way to include the model error term into the 4DVar cost function. As he points out, there is no unique choice for control variable in this case. For our implementation, we choose a variation of what he labels *model-bias control variable*. Consider the following representation of the true value of the state variables at any time:

$$\mathbf{x}^t = m^{0\rightarrow t}(\mathbf{x}^0) + \mathrm{b}^t \tag{30}$$

i.e. the state variables at any time can be written as the perfect evolution of the initial condition—given by the discrete-time map $m^{0\to t}$—plus an *effective* model error $\mathrm{b}^t \in \mathcal{R}^{N_x}$, which contains the accumulated effect of the real mode errors at each model step, along with their evolution. While Tremolet (2006) considered the effective model error to be constant for all observational times throughout the assimilation window ($\mathrm{b}^t = \mathrm{b} \forall 0 \leq t \leq \tau$), we let it vary for different observational times.

For a linear model $m^{t'\to t} = \mathbf{M}^{t'\to t}$, the relationship between the statistical characteristics of the model error n^t at every time step from 0 to t and those of the effective model error b^t at t can be found in the following way:

$$v^t \sim (\mathbf{0}, \mathbf{Q}) \to \mathrm{b}^t \sim \left(\mathbf{0}, \mathbf{Q}^t = \sum_{t'=0}^{t} \mathbf{M}^{t'\to t} \mathbf{Q} (\mathbf{M}^{t'\to t})^{\mathbf{T}} \right) \tag{31}$$

which is not an easy relationship, but later we will discuss simpler—and more modest—ways to approximate $\mathbf{Q}^t$. For the moment we consider this error to be unbiased.

Let us express the WC4DVar cost function as:

$$\begin{aligned} J(\boldsymbol{x}^0, \boldsymbol{\beta}^{1:\tau}) = & \frac{1}{2} (\boldsymbol{x}^0 - \boldsymbol{x}^{b,0})^{\mathbf{T}} \mathbf{B}^{-1} (\boldsymbol{x}^0 - \boldsymbol{x}^{b,0}) + \frac{1}{2} \sum_{t=1}^{\tau} (\boldsymbol{\beta}^t)^{\mathbf{T}} (\mathbf{Q}^t)^{-1} \boldsymbol{\beta}^t \\ & + \frac{1}{2} \sum_{t=1}^{\tau} \rho^t (\mathbf{y}^{o,t} - h(m^{0\to t}(\boldsymbol{x}^0) + \boldsymbol{\beta}^t))^{\mathbf{T}} \mathbf{R}^{-1} (\mathbf{y}^{o,t} - h(m^{0\to t}(\boldsymbol{x}^0) + \boldsymbol{\beta}^t)) \end{aligned} \tag{32}$$

where the function now depends upon $\mathbf{x}^0$ and $\boldsymbol{\beta}^{1:\tau} = \{\boldsymbol{\beta}^1, \cdots, \boldsymbol{\beta}^\tau\}$. Notice that all the (perfect) model evolutions $m^{0\to t}$ start at time 0. This is why we have chosen (30) as a definition of our control variables. If we chose the model error increment for every time step $\boldsymbol{v}^t$, we would find a cost function with terms of the form $m^{t'\to t}(\mathbf{x}^{t'})$, i.e. perfect model evolutions initialised at different model time steps, where $\mathbf{x}^{t'}$ contains the effects of all $\boldsymbol{v}$ for previous model time steps. This would complicate the problem considerably, especially when using ensembles.

As in the 4DVar case, we can write down a preconditioned incremental formulation. Recalling that:

$$(\mathbf{B}^t)^{1/2} = \mathbf{M}^{0\to t} (\mathbf{B}^0)^{1/2} \tag{33}$$

we express the control variables as:

$$\begin{aligned} \mathbf{x}^0 &= \mathbf{x}^{b,0} + \delta \mathbf{x}^0 = \mathbf{x}^{b,0} + \mathbf{B}^{1/2} \mathbf{v}^0 \\ \mathrm{b}^t &= \mathbf{0} + \delta \mathrm{b}^t = (\mathbf{B}^t)^{1/2} \mathbf{v}^t \end{aligned} \tag{34}$$

which allows to perform the following first order Taylor expansion:

$$\begin{aligned} \mathbf{y}^{o,t} - h(m^{0\to t}(x^0) + \beta^t) &\approx d^t - \mathrm{H} \mathrm{M}^{0\to \mathrm{t}} (\mathrm{B}^0)^{1/2} \mathrm{v}^0 - \mathrm{H} (\mathrm{B}^{\mathrm{t}})^{1/2} \mathrm{v}^{\mathrm{t}} \\ &\approx d^t - \mathrm{H} (\mathrm{B}^{\mathrm{t}})^{1/2} (\mathrm{v}^0 + \mathrm{v}^{\mathrm{t}}) \end{aligned} \tag{35}$$

where $\mathbf{d}^t = \mathbf{y}^{o,t} - h(m^{0\to t}(\mathbf{x}^{b,0}))$ as in the SC case. The incremental preconditioned form of the cost function for WC4DVar is:

$$\begin{aligned} J(\mathbf{v}^{0:\tau}) = & \frac{1}{2} (\mathbf{v}^0)^{\mathbf{T}} \mathbf{v}^0 + \frac{1}{2} \sum_{t=1}^{\tau} ((\mathbf{B}^t)^{1/2} \mathbf{v}^t)^{\mathbf{T}} (\mathbf{Q}^t)^{-1} (\mathbf{B}^t)^{1/2} \mathbf{v}^t) \\ & + \frac{1}{2} \sum_{t=1}^{\tau} \rho^t (\mathbf{d}^t - \mathbf{H} (\mathbf{B}^t)^{1/2} (\mathbf{v}^0 + \mathbf{v}^t))^{\mathbf{T}} \mathbf{R}^{-1} (\mathbf{d}^t - \mathbf{H} (\mathbf{B}^t)^{1/2} (\mathbf{v}^0 + \mathbf{v}^t)) \end{aligned} \tag{36}$$

We can write the gradient as the following vector:

$$\nabla J(\mathbf{v}^{0:\tau}) = \begin{bmatrix} \mathbf{v}^0 - \sum_{t=1}^{\tau} \rho^t (\mathbf{B}^t)^{1/2^{\mathbf{T}}} \mathbf{H}^{\mathbf{T}} \mathbf{R}^{-1} (\mathbf{d}^t - \mathbf{H} (\mathbf{B}^t)^{1/2} (\mathbf{v}^0 + \mathbf{v}^t)) \\ (\mathbf{B}^1)^{1/2^{\mathbf{T}}} (\mathbf{Q}^1)^{-1} (\mathbf{B}^1)^{1/2} \mathbf{v}^1 - \rho^1 (\mathbf{B}^1)^{1/2^{\mathbf{T}}} \mathbf{H}^{\mathbf{T}} \mathbf{R}^{-1} (\mathbf{d}^1 - \mathbf{H} (\mathbf{B}^1)^{1/2} (\mathbf{v}^0 + \mathbf{v}^1)) \\ \vdots \\ (\mathbf{B}^\tau)^{1/2^{\mathbf{T}}} (\mathbf{Q}^\tau)^{-1} (\mathbf{B}^\tau)^{1/2} \mathbf{v}^\tau - \rho^\tau (\mathbf{B}^\tau)^{1/2^{\mathbf{T}}} \mathbf{H}^{\mathbf{T}} \mathbf{R}^{-1} (\mathbf{d}^\tau - \mathbf{H} (\mathbf{B}^\tau)^{1/2} (\mathbf{v}^0 + \mathbf{v}^\tau)) \end{bmatrix} \tag{37}$$

where the control variable is $\mathbf{v}^{0:\tau} \in \mathcal{R}^{(\tau+1)N_x}$.

It would seem that the size of the problem has grown considerably with respect to the SC case. Nonetheless, let us look closer at the structure of (37). Remember that $\rho^t = 1$ only if t belongs to those time steps with observations (a set we denote as $\{t_{obs}\}$), and $\rho^t = 0$ otherwise. Hence the sum in the first block-element of $\nabla_{\mathbf{v}^{0:\tau}} J(\mathbf{v}^{0:\tau})$ only has τ_{obs} (total number of observational times) terms. For all the other block elements we have two terms: the first corresponding to the model error contribution and the second contribution to the observational error contribution. For all time steps with no observations, this later term is null. In these time steps there is nothing to make the model error to be different from its background value, which was zero by construction. Therefore $\mathbf{v}^t = 0$ for all time steps without observations, and consequently we can redefine our control variable from $\mathbf{v}^{0:\tau}$ to $\mathbf{v}^{0:\tau_{obs}} = \{\mathbf{v}^0, \mathbf{v}^{\{t_{obs}\}}\} \in \mathcal{R}^{(1+\tau_{obs})N_x}$. Hence, the number of control variables—as well as the computational cost of the problem—scales with the number of observational times included in a given assimilation window.

2.4. Weak-constraint 4DEnsembleVar

As in the SC case, we want to find a way to avoid computing TLM's and AM's. In the absence of localisation, we substitute $(\mathbf{B}^t)^{1/2} = \hat{\mathbf{X}}^{b,t}$ and $\mathbf{H}(\mathbf{B}^t)^{1/2} = \hat{\mathbf{Y}}^{b,t}$ to yield the gradient of the cost function as:

$$\nabla J(\mathbf{v}^{0:\tau_{obs}}) =$$

$$\begin{bmatrix} \mathbf{v}^0 - \sum_{t=1}^{\tau_{obs}} (\hat{\mathbf{Y}}^{b,t})^{\mathbf{T}} \mathbf{R}^{-1} (\mathbf{d}^t - \hat{\mathbf{Y}}^{b,t} (\mathbf{v}^0 + \mathbf{v}^t)) \\ (\hat{\mathbf{X}}^{b,1})^{\mathbf{T}} (\mathbf{Q}^1)^{-1} \hat{\mathbf{X}}^{b,1} \mathbf{v}^1 - (\hat{\mathbf{Y}}^{b,1})^{\mathbf{T}} \mathbf{R}^{-1} (\mathbf{d}^1 - \hat{\mathbf{Y}}^{b,1} (\mathbf{v}^0 + \mathbf{v}^1)) \\ \vdots \\ (\hat{\mathbf{X}}^{b,\tau_{obs}})^{\mathbf{T}} (\mathbf{Q}^{\tau_{obs}})^{-1} \hat{\mathbf{X}}^{b,\tau_{obs}} \mathbf{v}^{\tau_{obs}} - (\hat{\mathbf{Y}}^{b,\tau_{obs}})^{\mathbf{T}} \mathbf{R}^{-1} (\mathbf{d}^{\tau_{obs}} - \hat{\mathbf{Y}}^{b,\tau_{obs}} (\mathbf{v}^0 + \mathbf{v}^{\tau_{obs}})) \end{bmatrix} \tag{38}$$

where the control variable is $\mathbf{v}^{0:\tau_{obs}} \in \mathcal{R}^{(1+\tau_{obs})N_e}$. As in the SC case, the actual implementation details can be found in Appendix 1.

In the SC case, the influence from observations at different times leads to a single increment that is computed at initial time $\mathbf{x}_0^b \rightarrow \mathbf{x}_0^a$, and the analysis trajectory comes from the evolution of the new initial condition. Roughly speaking, the information from observations at time t impact changes at time 0 via the covariance $(\hat{\mathbf{X}}^0\hat{\mathbf{Y}}^t) \circ \mathbf{L}_{\mathbf{xx}}^{0,t}$. Since we do not have $\mathbf{L}_{\mathbf{xx}}^{0,t}$ we simply use $\mathbf{L}_{\mathbf{xx}}^{0,0}$. This is not too inaccurate for observations occurring relatively close (in a time-distance sense) to the beginning of the assimilation window. However using $\mathbf{L}_{\mathbf{xx}}^{0,0}$ instead of $\mathbf{L}_{\mathbf{xx}}^{0,t}$ can lead to considerable errors as t grows. Hence, this issue is of particular importance for long assimilation windows with observations close to the end of the window.

In the WC formulation we propose, the impact of an observation at time t influences the state variable at $t = 0$ and at time $t = t_{obs}$. So although we still have an incorrectly localised expression of the form $(\hat{\mathbf{X}}^0\hat{\mathbf{Y}}^t) \circ \mathbf{L}_{\mathbf{xx}}^{0,t}$, we also have a correctly localised term $(\hat{\mathbf{X}}^t\hat{\mathbf{Y}}^t) \circ \mathbf{L}_{\mathbf{xx}}^{t,t}$, where obviously $\mathbf{L}_{\mathbf{xx}}^{t,t} = \mathbf{L}_{\mathbf{xx}}^{0,0}$. We have not solved the problem of localisation, but we try to ameliorate it by allowing for increments at time steps in which observations occur.

The way our formulation works, we have jumps (from background to analysis) at the initial time and at the time of the observations. We have not experimented with how to get a smooth trajectory for the whole assimilation window. A first approach would be to modify the initial conditions, and then distribute the updates at observational times amongst the time steps between observations, following a procedure similar to the Incremental Analysis Update of Bloom et al. (1996). This would avoid the sharp jumps from background to analysis at the times of the observations.

2.5. A note on $\mathbf{Q}^t$

We have not said anything about the computation of $\mathbf{Q}^t$, i.e. the covariance matrix of the effective model error β^t. In principle, it could be calculated if we had access to two ensembles: one evolved with no model error using (1): $\mathbf{X}_{per}^{0:\tau}$, and one evolved with model error using (29): $\mathbf{X}_{imp}^{0:\tau}$. For any time t we can use the expression (30) to write the imperfect state in terms of the perfect state at that time plus an effective error $\boldsymbol{\beta}^t$:

$$\boldsymbol{\beta}^t = \boldsymbol{x}_{imp}^t - \boldsymbol{x}_{per}^t \tag{39}$$

We can compute the first two moments of this error as:

$$\begin{aligned} E[\beta^t] &= E[x_{imp}^t] - E[x_{per}^t] \\ \mathbf{Q}^t &= Cov[\beta^t] = Cov[x_{imp}^t - x_{per}^t] \end{aligned} \tag{40}$$

and the previous expressions can be approximated by evaluating sample estimators:

$$\begin{aligned} \bar{\beta}^t &= (\mathbf{X}_{imp}^t - \mathbf{X}_{per}^t)\mathbf{1} \\ \hat{\mathbf{Q}}^t &= (\hat{\mathbf{X}}_{imp}^t - \hat{\mathbf{X}}_{per}^t)(\hat{\mathbf{X}}_{imp}^t - \hat{\mathbf{X}}_{per}^t)^{\mathbf{T}} \end{aligned} \tag{41}$$

There are two issues to do with these calculations. First of all, we need two ensembles running at the same time. This is not too demanding; in fact we could divide a moderate size ensemble into two groups and run each of these groups with and without model error respectively. The major complication, however, is the evaluation of

$$\left(\hat{\hat{\mathbf{Q}}}^t\right)^{-1} = \left(\left((\hat{\mathbf{X}}_{imp}^t - \hat{\mathbf{X}}_{per}^t)(\hat{\mathbf{X}}_{imp}^t - \hat{\mathbf{X}}_{per}^t)^{\mathbf{T}}\right) \circ \mathbf{L}_{xx}\right)^{-1} \tag{42}$$

in an efficient way. Some of the difficulties include the size of the resulting matrix, as well as the rank of the matrix. We have not done exploration in this direction. Instead we have chosen a rather simple parametrisation of this error as:

$$\mathbf{Q}^t \propto t\mathbf{Q} \tag{43}$$

which can be interpreted as considering the accumulated model error to behave like the Wiener process. This is exact for a linear model with $m = \mathbf{I}$, i.e. when the evolution model is the identity. A more precise approximation—or indeed an efficient way to compute the actual value—is beyond the scope of this paper.

3. Illustration of the time-propagation of information: TLM vs 4D cross-time covariances

3.1. The KdV equation

In this section we illustrate the effects of using a static localisation matrix to localise time cross-covariances. We perform experiments using the Korteweg de Vries (KdV) system as model. The KdV equation is a non-linear partial differential equation in one dimension:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial s} + \frac{\partial^3 u}{\partial s^3} = 0 \tag{44}$$

where t denotes time, $s$is the spatial dimension, and u is a one-dimensional velocity field. This equation admits solutions called solitons, which correspond to the propagation of a single coherent wave, see e.g. Shu (1987) or Zakharov and Faddeev (1971).

This system has been used before for DA experiments in the context of *feature* DA and *alignment* error (e.g. Van Leeuwen, 2003; Lawson and Hansen, 2005). This system can be challenging for DA, since the propagation of coherent structures renders the linear and Gaussian assumptions in both the background and observational error to become less accurate.

Nonetheless, the system is suited to show the emergence of asymmetric off-diagonal elements in the time cross-covariance matrices. Hence, we will use it for illustrative purposes and simple DA experiments. More detailed experiments are shown in part II of this paper with more appropriate models.

We reduce (44) into an ordinary differential equation (ODE) by discretising s into N_g grid points $\mathbf{s} = [s_1, \cdots, s_j, \cdots, s_{N_g}]$ separated by Δs, and perform a central and 2^{nd}-order-accurate finite-difference approximation to both the dispersion and advection terms (after writing the latter in conservative form). Denoting $u_j = u(s_j)$ and $\mathbf{u} = [u_1, \cdots, u_j, \cdots, u_{N_g}]$, we can write:

$$\frac{du_j}{dt} = f(\mathbf{u}) = \frac{u_{j-2} - u_{j+2}}{2(\Delta s)^3} + \frac{u_{j-1}}{4\Delta s}\left(u_{j-1} - \frac{4}{(\Delta s)^2}\right) - \frac{u_{j+1}}{4\Delta s}\left(u_{j+1} - \frac{4}{(\Delta s)^2}\right) \tag{45}$$

We choose $N_g = 15$ grid points and $\Delta s = 1$, and allow for periodic boundary conditions, i.e. $u_j \equiv u_{jmod(N_g)}$, where *mod* denotes the modulo operation. We use a 4^{th}-order Runge-Kutta method with time step $\Delta t = 0.25$ to numerically integrate (45) in time. This integration renders a discrete-time map $m^{0\to t}$.

The initial condition of a perfect soliton is given by:

$$u_j(t=0) = 3A\,sech^2\left(\frac{\sqrt{A}}{2}(s_j - s_o)\right) \tag{46}$$

where $s_o = 5$ is the center of the soliton at $t = 0$, and $3A$ corresponds to the maximum velocity of the soliton. We choose $A = 1$ (the propagation speed) which, combined with our steps $\Delta t = 0.25$ and $\Delta s = 1$, yield a Courant number $C = 0.75$. Figure 2 is a Hovmoller plot showing the time evolution of the soliton through the domain. The horizontal axis represents different grid points, the vertical axis represents time (evolving from bottom to top), and the shading is proportional to the velocity at each grid point.

For our experiments we require the discrete-time TLM (or transition matrix) $\mathbf{M}^{0\to t}$of the model. We first obtain the continuous-time TLM $\mathbf{F} = \frac{\partial f}{\partial \mathbf{s}}$. This is a hollow circulant matrix in which the j^{th} row has all elements equal to zero except for:

$$\begin{aligned} \mathbf{F}[j,(j\mp 2)] &= \pm\tfrac{1}{2(\Delta s)^3} \\ \mathbf{F}[j,(j\mp 1)] &= \pm\tfrac{1}{2\Delta s}u_{j-1} \mp \tfrac{1}{(\Delta s)^3} \end{aligned} \tag{47}$$

where we recall that the indices are modular. Obtaining the transition matrix $\mathbf{M}$ from the continuous-time TLM $\mathbf{F}$ involves a straightforward solution of a matrix differential equation, see e.g. Simon (2006).

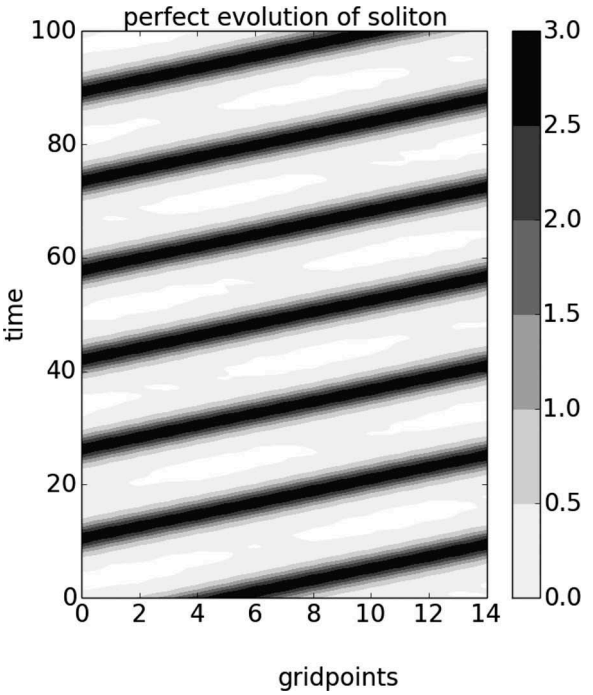

Fig. 2. Hovmoller plot showing the perfect evolution of a soliton produced by a numerical integration of the KdV equation over a periodical domain of 15 grid points.

3.2. Covariance propagation

Let us start by obtaining the nature run of our system. As initial condition we use the profile described in (46), and we evolve the model until $t = 100$ (i.e. 400model steps). The result of this run is illustrated in Fig. 2, this model run will later be used as the *nature*—or *true*—run of our DA experiments. The soliton propagates from left to right of the domain, taking about $\Delta t = 15$ (60time steps) to complete a loop around the domain.

We now examine the time propagation of a given $\mathbf{B}^c$. We construct this matrix following the method of Yang et al. (2006), which we describe briefly in Appendix 2. We obtain a circulant matrix, which normalized values for the j^{th} row are:

$$\mathbf{B}^c[j,:] = [\cdots, 0.0, -0.1274, 0.3060, 0.5220, 1.0, 0.5220, 0.3060, -0.1274, 0.0, \cdots] \tag{48}$$

Since we have the transition matrix of the model, we can explicitly compute both the covariance of the state at any time t as:

$$\mathbf{B}^t = \mathbf{M}^{0\to t}\mathbf{B}^0\mathbf{M}^{0\to t\mathbf{T}} \tag{49}$$

and the cross-covariance between time $t = 0$ and time t as:

$$Cov(\mathbf{x}^0, \mathbf{x}^t) = \mathbf{B}^0\mathbf{M}^{0\to t\mathbf{T}} \tag{50}$$

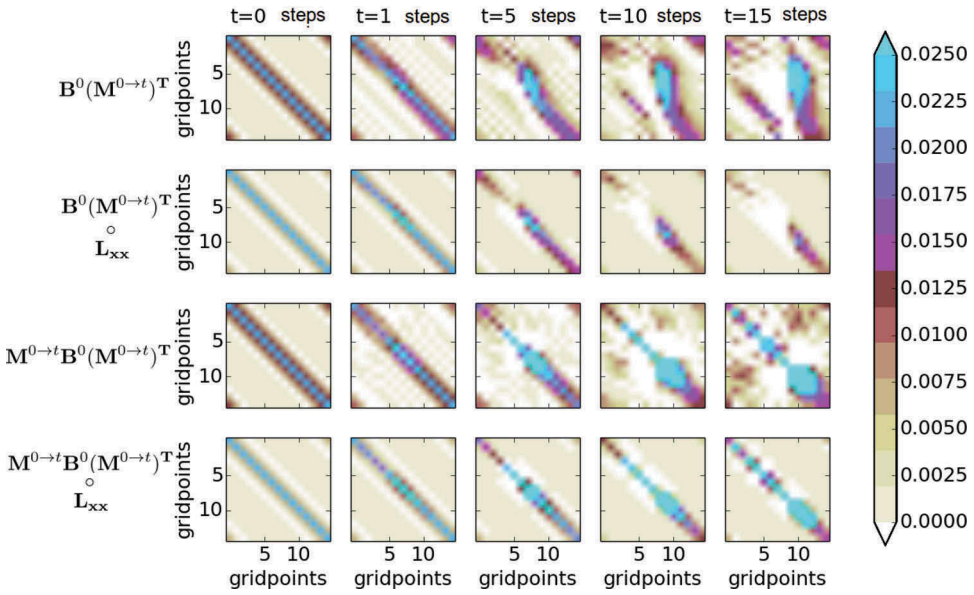

Fig. 3. Analytical evolution of the covariance $\mathbf{B}^t$ and cross-time covariance $Cov(\mathbf{x}^0, \mathbf{x}^t)$ for the KdV model for different times (columns). The horizontal and vertical axes of each panel are the grid point locations. The first row shows $Cov(\mathbf{x}^0, \mathbf{x}^t)$ and the second shows its localised version. The third row shows $\mathbf{B}^t$ and the fourth shows its localised version.

We evaluate both (50) and (49) for different lead times t, and we plot these matrices in Fig. 3. Each individual panel is a covariance matrix, the horizontal and vertical axes are grid point locations. Let us start with the top row, which shows $Cov(\mathbf{x}^0, \mathbf{x}^t)$ for different lead times $t = \{0, 1, 5, 10, 15\}$ (one for every column). The initial covariance is the circulant matrix shown in (48), but as the time increases we notice that off-diagonal features become more prominent. We notice a region of high values (shown in blue) developing above the main diagonal and the appearance of negative values (shown in white) along the main diagonal. The resulting matrices are not symmetric starting from $t = 1$, and this asymmetry grows as time passes. This is particularly evident after 10 and 15 steps.

The third row shows $\mathbf{B}^t$ for different lead times $t = \{0, 1, 5, 10, 15\}$. In this case, all the matrices are symmetric by construction. There is a development of (symmetric) off-diagonal negative elements (shown in white), but most of the information remains concentrated close to the main diagonal. The magnitude of the matrix elements increases as time passes. As expected uncertainty grows in time.

Recall that we computed both $Cov(\mathbf{x}^0, \mathbf{x}^t)$ and $\mathbf{B}^t$ using $\mathbf{B}^0$ and the actual transition matrix $\mathbf{M}^{0\to t}$. The objective of 4DEnVar, however, is to compute these quantities using ensemble estimators, which contain spurious correlations if the ensemble size is small and lead to the need for localisation. The second and fourth rows of figure 3 show the effect of localisation of $Cov(\mathbf{x}^0, \mathbf{x}^t)$ and $\mathbf{B}^t$ respectively, using a localisation matrix $\mathbf{L}^{00}_{\mathbf{xx}}$ which follows a Gaspari and Cohn (1999) function with half-width $\lambda = 1$ grid points (centred in the main diagonal). As we can see in row 2, the localised versions of $Cov(\mathbf{x}^0, \mathbf{x}^t)$ can lose most of the important information, and this problem becomes more pronounced as the time lag increases. The localised versions of $\mathbf{B}^t$, shown in the bottom row, lose less information, since the most distinguishable features were close to the main diagonal.

To appreciate better the effect of localisation with different half-widths, in Fig. 4 we depict the values of the 6^{th} row of the covariance matrices. The left panel of the first row shows $Cov(\mathbf{x}^0, \mathbf{x}^t)[6, :]$. Each one of the different lines shows a different lead-time. Consider we are observing all variables, and look at the red line in the figure. We see a peak corresponding to grid point 10: this means that, for a lead-time of 15 time steps, the most useful information to update grid point 6 (at $t = 0$) comes from the observation at grid point 10. This is no surprise, since the general direction of the flow is from left to right. Actually, for all lead-times we see that the most important information to update grid point 6 comes from observations in the grid points to the right. When localising the covariances this information is lost. This is not too crucial for relative large-scale localisation cases (middle column), but becomes very evident for strict localisation (right column). In both of these panels, the shape of the localisation function is shown in light gray for reference.

The left panel of the second row of Fig. 4 shows the 6^{th} of $\mathbf{B}^t$ (actually we normalise and create a correlation-type matrix for ease of comparison, since the magnitudes in this matrix grow with time), and again different lines show a different

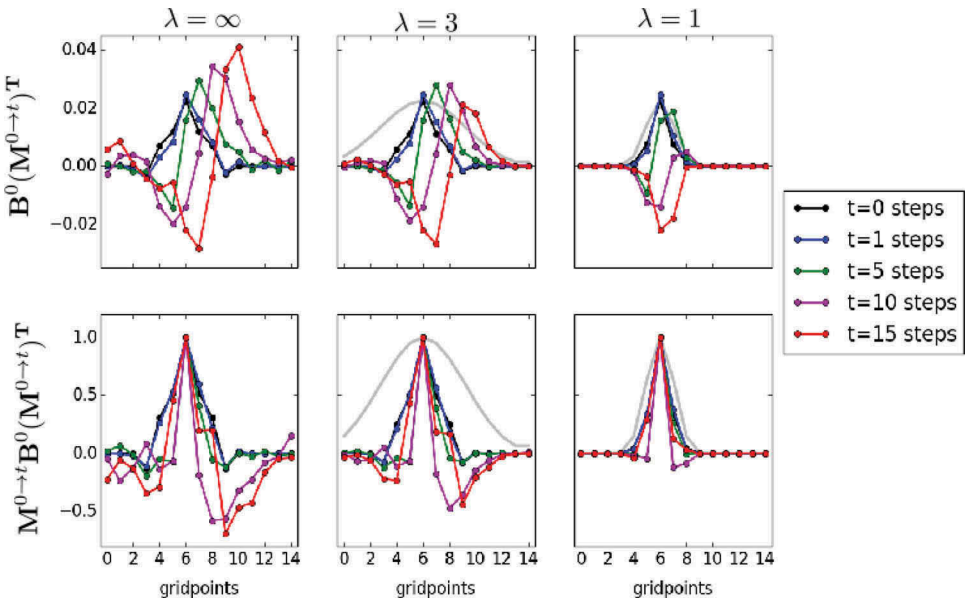

Fig. 4. Elements of the 6th row of the matrices $Cov(\mathbf{x}^0, \mathbf{x}^t)$ (top row) and $\mathbf{B}^t$ (bottom row), for different times (different colors). The first column shows the unaltered elements, while the middle and right columns show these elements after being localised using Gaspari-Cohn functions of different half-widths (gray lines).

lead-times. In all cases the dominant element is that corresponding to grid point 6. New features do develop (with respect to the original $\mathbf{B}^0$, i.e. the black line), but none is dominant. When localising with both a large- (middle column) and small- (right column) scale the information is not lost as crucially as before.

4. DA experiments under a perfect scenario

4.1. Setup

For these experiments, we extend the nature run generated before until $t = 200$ (i.e. we have 800 model steps). We create synthetic observations by taking the value of the variables at all grid points and all model time steps, and adding an uncorrelated random error with zero mean and $\mathbf{R} = \sigma^{o2}\mathbf{I}$, with $\sigma^{o2} = 0.1$. For the experiments we use subsets of these observations, both in time and space.

We observe every 3^{rd} grid point; hence we have 5 observed grid points and 10 unobserved grid points. This sparse observational density (in space) makes the situation challenging, and the quality of the off-diagonal elements in the covariance matrices are quite important to communicate information from observed to unobserved variables. We experiment with several observational periods. We report the results of 3 cases: observations every 2 model steps, every 5 model steps, and every 10 model steps.

We test the following formulations: 3DVar, SC4DVar, ETKS, LETKS, SC4DEnVar and LSC4DEnVar. The L at the beginning of the EnVar methods denote the localised versions. ETKS and LETKS (S for smoother instead of filter) denote the 'no-cost smoother' versions of both ETKF and LETKF (see Kalnay and Yang, 2010 for a description).

Even though the nature run is generated with a perfect model, we also use WC methods for the sake of comparison. We use the *effective* error WC4DVar, as well as WC4DEnVar and LWC4DEnvar. In this case, we model the (fictitious) *effective* model error as a zero-center Gaussian rv with $\mathbf{Q} = 0.01\ \mathbf{B}_c$.

For the ensemble and hybrid methods we use a tiny ensemble size of $N_e = 3$. With this choice we try to mimic realistic situations where $N_e \ll N_x$. This chosen value of N_e should render low-quality sample estimators. The adaptive inflation of Miyoshi (2011) is used, with an initial value of $\rho_0 = 0.05$. Inflation is generally applied in the following manner:

$$\begin{aligned} \hat{\mathbf{X}}^b &\to (1+\rho)\hat{\mathbf{X}}^b \\ \mathbf{P}^b &\to (1+\rho)^2\mathbf{P}^b \end{aligned} \tag{51}$$

but following Miyoshi (2011), the inflation values are different for each grid point and it evolves with time.

For the localised methods, a Gaspari-Cohn function with half-width λ is used. We explored several values and kept the optimal one, which depends on the period of observations in each experiment. For the localised EnVar methods, we truncated the

localisation matrix at 11 eigenvalues. Truncation with 9 to 15 kept eigenvalues was tried and gave similar results (not shown).

For the variational and hybrid methods, we use one observational time per assimilation window. We use the incremental pre-conditioned forms discussed earlier, and we do not use outer loops in any of these methods. We later performed experiments with 2 observational times per assimilation window and this did not change the main results (hence not shown).

4.2. Results

The results of these experiments are depicted in Figs. 5–7. Each figure has 3 columns and 2 rows. The top row corresponds to results of *observed* grid points, while the bottom row corresponds to results of *unobserved* grid points. Three panels are shown in each case:

i. The left panel shows the time evolution of the truth (black line) over $41 \leq t \leq 49$, as well as the analysis trajectories reconstructed by the different DA methods (shown in different colors). As example of an observed variable we choose grid point 1; observations are shown as gray circles. As example of an unobserved variable we show grid point 3.

ii. The middle panel shows the evolution of the analysis RMSE with respect to the truth (over $41 \leq t \leq 49$) using:

$$RMSE(t) = \left[\sum_{n=1}^{N} \frac{\left(\mathbf{x}_n^a(t) - \mathbf{x}_n^{true}(t)\right)^2}{N}\right]^{\frac{1}{2}} \tag{52}$$

where we use the mean of the analysis ensemble for (L)ETKS, and the unique analysis trajectory for the Var and EnVar methods. The index n runs over observed and unobserved grid points depending on the case. The horizontal gray line shows the standard deviation of the observational error.

iii. The right panel shows summary statistics for the analysis RMSE. We eliminate the period $0 \leq t \leq 10$ (40 time steps) as a transient. For the remaining time steps we compute the 1^{st}, 2^{nd}(median) and 3^{rd} quartiles of the analysis RMSE. These are depicted—for each and every method—using an upward-pointing triangle, a circle, and a downward-pointing triangle respectively. Again, the standard deviation of the observational error is shown with a horizontal gray line.

4.2.1. Observations every 2 model steps. This is a relatively easy DA scenario. The results are shown in Fig. 5. For both observed and unobserved variables, the RMSE values of all DA methods is well under the level of observational error. The best performing methods are SC4DVar and WC4DVar, followed by SC4DEnVar, LSC4DEnVar and LWC4DEnVar. Localisation (slightly) improves the performance of

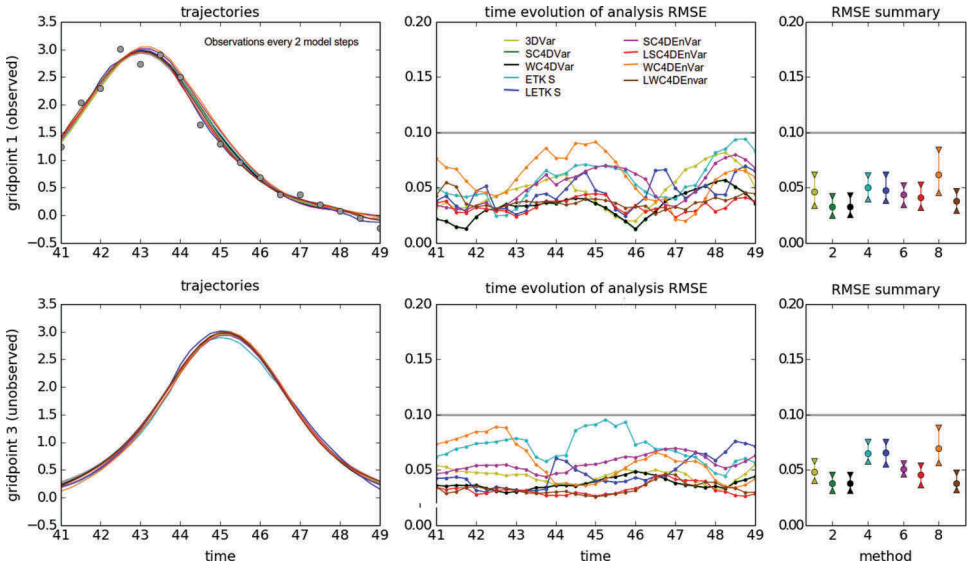

Fig. 5. Using different DA methods in the KdV model for an observation period of 2 model steps. The left panels show the time evolution of the nature run (black line) and the analysis trajectories generated by different DA methods (color lines). The top row shows the case of grid point 1 (observed variable, observations are represented with gray circles), while the bottom row shows the case of grid point 3 (unobserved variable). The middle panels show the time evolution of the analysis RMSE. The top panel corresponds to observed variables, while the bottom panel corresponds to unobserved variables. For both cases, the horizontal gray line is the standard deviation of the observational error. The right panels show summary statistics for observed (top panel) and unobserved variables (bottom panel). We show intervals containing the first, second (median) and third quartiles of the analysis RMSE distribution in time.

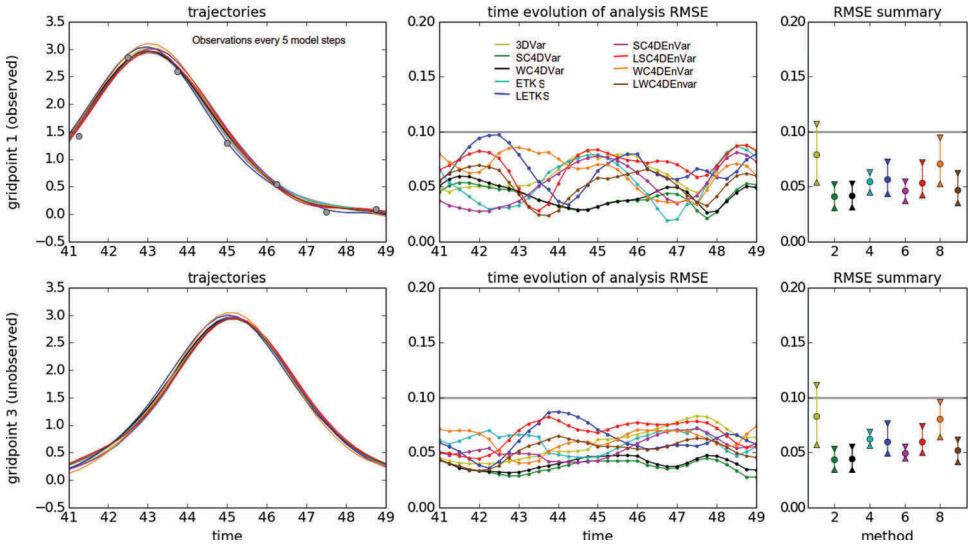

Fig. 6. Same as Fig. 5 but with an observational period of 5 model time steps.

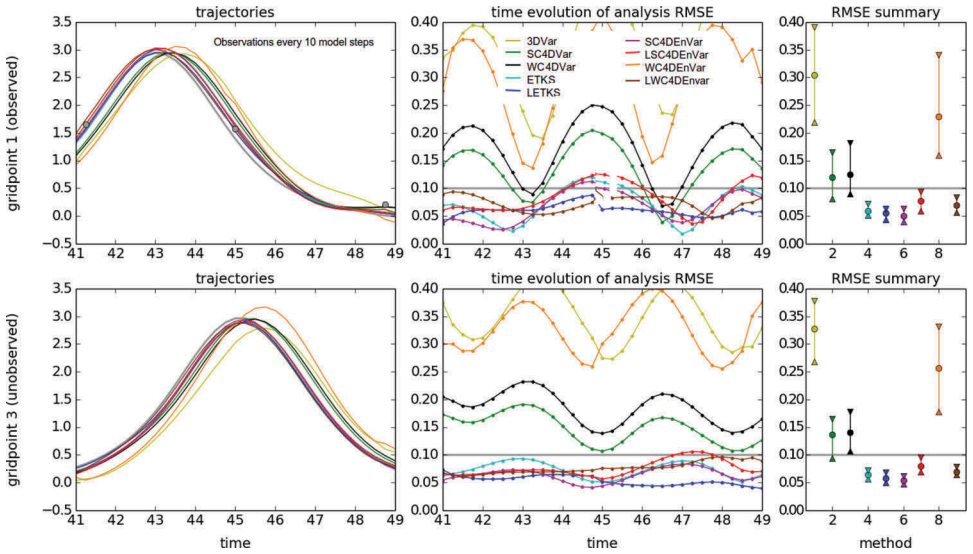

Fig. 7. Same as Fig. 5 but with an observational period of 10 model time steps.

SC4DEnVar, especially for the unobserved variables. 3DVar performs slightly better than 4DVar (both SC and WC), which is understandable since the observations are frequent. Both ETKS and LETKS perform worse than the Var and EnVar methods. Localisation does not help the performance of ETKS. We explored some values of localisation and settled with $\lambda = 1.0$, although we did not perform an exhaustive search. This same value was used in the EnVar methods. The worst performing method is WC4DEnVar (with no localisation), which is understandable since it contains the fictitious (and in this case unnecessary) model error, on top of the sample error from the ensemble part. Nonetheless, its performance is greatly improved by localisation.

4.2.2. Observations every 5 model steps. We increase the observational period to 5 model steps, the results are shown in Fig. 6. We start noticing a distinction in the performance of the DA methods. As expected, 3DVar is now the worst performing method for both observed and unobserved variables. Again, both SC4DVar and WC4DVar are the best methods, with SC4DEnVar and WSC4DEnVar next. Localisation damages the performance of SC4DEnVar, leading to larger RMSE's in general. We started

seeing this phenomenon with an observational period of $\Delta t \geq 4$ model time steps. For WC4DEnVar localisation improves the performance considerably. The performance of ETKS and LETKS is slightly better than that of the Var methods. Again, for (L)ETKS and the EnVar methods we used a localisation half-width of $\lambda = 1.0$.

4.2.3. Observations every 10 model steps. This rather large observational period presents a DA challenge. The results are presented in Fig. 7. 3DVar is the worst-performing method, with an RMSE considerably higher than the observational error for both observed and unobserved variables. SC4DVar and WC4DVar perform considerably worse than for the previous observational frequencies, both observed and unobserved variables. The observations are infrequent and the assimilation window is too long. Hence, the linearisation we perform in the incremental form loses validity and would require the use of outer loops. In this case both ETKS and LETKS are amongst the best performing methods, and localisation helps reduce the RMSE slightly.

SC4DEnVar is still the best method, and the use of the ensemble information allows the variational solution to find a global minimum with no need for outer loops. Nonetheless, we see that the inclusion of localisation damages the performance of the method. WC4DEnVar has a very bad performance when not localised. One could think that having more degrees of freedom (control variables) in the fitting could always help, but this is not the case when they provide wrong information. After localising, the performance improves considerably for both observed and unobserved variables.

For these experiments, the half-width used in the localisation of both (L)ETKS and the hybrid methods is $\lambda = 3$.

5. Conclusion and summary

In this paper we have explored the ensemble-variational DA formulation. As a motivation, we have discussed the problem of localising time cross-covariances using static localisation covariances, and we have illustrated this effect with the help of the KdV model. These time cross-covariances develop off-diagonal features that are erroneously eliminated when using localisation matrices that do not contain information of the direction of the flow.

Our main contribution is the introduction of an ensemble-variational method in a weak-constraint scenario, i.e. considering the effect of model error. For simplicity, we have chosen as control variable the *effective model error* at the time of observation.

We have discussed the implementation of the model-error term in the cost function using ensemble information. Our formulation leads to updates at the beginning of the assimilation window and at the observation times within the assimilation window. We only require one ensemble initialised at the beginning of the window, and that the size of the problem does not scale with the number of model steps. However, we require computing the correct statistics of this effective model error. We offer a modest approach for this effect. Our experiments also suggest that having updates at times different than the start of the window helps ameliorate (but does not solve) the problem of having badly-localised time cross-covariances.

The results presented in this part I encourage us to explore the performance of our methods in a more thorough manner in more appropriate models. This is the subject of part II.

Acknowledgements

JA and PJvL acknowledge the support of the National Environment Research Centre (NERC) via the National Centre for Earth Observation (NCEO). MG and PJvL acknowledge the support of NERC via the DIAMET project (NE/I0051G6/1). PJvL received funding from the European Research Council (ERC) under the European Union Horizon 2020 research and innovation programme.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Environment Research Centre (NERC); NERC [NE/I0051G6/1]; European Research Council (ERC) and National Centre for Earth Observation (NCEO).

References

Bannister, R. N. 2008. A review of forecast error covariance statistics in atmospheric variational data assimilation. II: Modelling the forecast error covariance statistics. *Q.J.R. Meteorol. Soc.* **134**, 1971–1996. DOI:10.1002/qj.v134:637.

Bishop, C., Etherton, B. and Majumdar, S. 2001. Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Mon. Weather Rev.* **129**, 420–436. DOI:10.1175/1520-0493(2001)129<0420:ASWTET>2.0.CO;2.

Bloom, S., Takacs, L., DaSilva, A. and Ledvina, D. 1996. Data assimilation using incremental analysis updates. *Mon. Wea. Rev.* **124**(6), 1256–1271. DOI:10.1175/1520-0493(1996)124<1256:DAUIAU>2.0.CO;2.

Bonavita, M., Isaksen, L. and Holm, E. 2012. On the use of EDA background error variances in the ECMWF 4D-Var. *Q. J. R. Meteorol. Soc.* **138**, 1540–1559. DOI:10.1002/qj.1899.

Buehner, M. 2005. Ensemble-derived stationary and flow- dependent background error covariances: Evaluation in a quasi-operational NWP setting. *Q. J. R. MeteoR. Soc.* **131**, 1013–1043. DOI:10.1256/qj.04.15.

Clayton, A. M., Lorenc, A. C. and Barker, D. M. 2013. Operational Implementation of a Hybrid Ensemble/4D-Var Global Data Assimilation System at the Met Office. *Quart. J. Roy. Meteor. Soc.* DOI:10.1002/qj.2054.

Evensen, G. and Van Leeuwen, P. J. 2000. An ensemble Kalman smoother for nonlinear dynamics. *Mon. Wea. Rev.* **128**, 1852–1867. DOI:10.1175/1520-0493(2000)128<1852:AEKSFN>2.0.CO;2.

Fairbarn, D., Pring, S., Lorenc, A. and Roulsone, I. 2014. A comparison of 4DVAR with ensemble data assimilation methods. *Q.J.R. Meteorol. Soc.* **140**, 281–294. DOI:10.1002/qj.2135.

Gaspari, G. and Cohn, S. E. 1999. Construction of correlation functions in two and three dimensions. *Q. J. R. Meteorol. Soc.* **125**, 723–757. DOI:10.1002/qj.49712555417.

Goodliff, M., Amezcua, J. and Van Leeuwen, P. J. 2015. Comparing hybrid data assimilation methods on the Lorenz 1963 model with increasing non-linearity. *Tellus A* **67**, 26928. DOI:10.3402/tellusa.v67.26928.

Hunt, B. R., Kostelich, E. J. and Szunyogh, I. 2007. Efficient data assimilation for spatiotemporal chaos: a local ensemble transform Kalman filter. *Phys. D* **230**, 112–126. DOI:10.1016/j.physd.2006.11.008.

Kalnay, E. and Yang, S.-C. 2010. Accelerating the spin-up of Ensemble Kalman Filtering. *Quart. J. Roy. Meteor. Soc.* **136**, 1644–1651. DOI:10.1002/qj.652.

Lawson, W. G. and Hansen, J. A. 2005. Alignment Error Models and Ensemble-Based Data Assimilation. *Mon. Wea. Rev.* **133**, 1687–1709. DOI:10.1175/MWR2945.1.

Le Dimet, F.-X. and Talagrand, O. 1986. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Q. J. R. Meteorol. Soc.* **38A**, 97–110.

Liu, C., Xiao, Q. and Wang, B. 2008. An ensemble-based four-dimensional variational data assimilation scheme. Part I: Technical formulation and preliminary test. *Mon. Wea. Rev.* **136**(9), 3363–3373. DOI:10.1175/2008MWR2312.1.

Lorenc, A. 2003. The potential of the ensemble Kalman filter for NWP – a comparison with 4D-Var. *Q. J Roy. Meteorol. Soc.* **129** (595B), 3183–3203. DOI:10.1256/qj.02.132.

Lorenc, A. C., Bowler, N. E., Clayton, A. M., Pring, S. R., and Fairbairn, D. 2015. Comparison of hybrid-4DEnVar and hybrid-4DVar data assimilation methods for global NWP. *Q J. Roy. Met. Soc.* **143**. DOI:10.1175/MWR-D-14-00195.1.

Lorenz, E. 1963. Deterministic Non-periodic Flow. *J. Atmos. Sci.* **20**, 130–141.

Lorenz, E. N. 1996. Predictability: A problem partly solved. In: *Proc. ECMWF Seminar on Predictability*, 4–8 September 1995, Reading, UK, pp. 1–18.

Lorenz, E. N. 2005. Designing chaotic models. *J. Atmos. Sci.* **62**, 1574–1587. DOI:10.1175/JAS3430.1.

Miyoshi, T. 2011. The Gaussian approach to adaptive covariance inflation and its implementation with the local ensemble transform Kalman filter. *Mon. Weather Rev.* **139**, 1519–1535.

Parrish, D. F. and Derber, J. C. 1981. The National Meteorological Center's spectral statistical interpolation analysis system. *Mon. Weather Rev.* **120**, 1747–1763. DOI:10.1175/1520-0493(1992)120<1747:TNMCSS>2.0.CO;2.

Shu, -J.-J. 1987. The proper analytical solution of the Korteweg-de Vries-Burgers equation. *J. Phys. A Math. Gen.* **20**(2), 49–56. DOI:10.1088/0305-4470/20/2/002.

Simon, D. 2006. *Optimal State Estimation*. *John Wiley & Sons*, New York, 552 p.

Talagrand, O. and Courtier, P. 1987. Variational Assimilation of Meteorological Observations With the Adjoint Vorticity Equation. I: Theory. *Q.J.R. Meteorol. Soc.* **113**, 1311–1328. DOI:10.1002/qj.49711347812.

Tremolet, Y. 2006. Accounting for an imperfect model in 4D-Var. *Q. J. R. Meteorol. Soc.* **132**, 2483–2504. DOI:10.1256/qj.05.224.

Van Leeuwen, P. J. 2003. A Variance-Minimizing Filter for Large-Scale Applications. *Mon. Wea. Rev.* **131**, 2071–2084. DOI:10.1175/1520-0493(2003)131<2071:AVFFLA>2.0.CO;2.

Van Leeuwen, P. J. and Evensen, G. 1996. Data assimilation and inverse methods in terms of a probabilistic formulation. *Mon. Wea. Rev.* **124**, 2898–2914. DOI:10.1175/1520-0493(1996)124<2898:DAAIMI>2.0.CO;2.

Wang, X., Bishop, C. and Julier, S. 2004. Which is better, an ensemble of positive-negative pairs or a centered spherical simplex ensemble?. *Mon. Wea. Rev.* **132**, 1590–1605. DOI:10.1175/1520-0493(2004)132<1590:WIBAEO>2.0.CO;2.

Wang, X., Snyder, C. and Hamill, T. M. 2007. On the Theoretical Equivalence of Differently Proposed Ensemble 3DVAR Hybrid Analysis Schemes. *Mon. Wea. Rev.* **135**, 222–227. DOI:10.1175/MWR3282.1.

Whitaker, J. S. and Hamill, T. M. 2002. Ensemble data assimilation without perturbed observations. *Mon. Weather Rev.* **130**, 1913–1927. DOI:10.1175/1520-0493(2002)130<1913:EDAWPO>2.0.CO;2.

Wursch, M. and Craig, G. C. 2014. A simple dynamical model of cumulus convection for data assimilation research. *Meteorologische Zeitschrift* **23**(5), 483–490.

Yang, S.-C., Baker, D., Li, H., Cordes, K., Huff, M.et al. 2006. Data assimilation as synchronization of truth and model: experiments with the three-variable Lorenz system. *J. Atmos. Sci.* **63**(9), 2340–2354. DOI:10.1175/JAS3739.1.

Zakharov, V. E. and Faddeev, L. D. 1971. Korteweg-de Vries equation: A completely integrable Hamiltonian system. *Funct. Anal. Appl.* **5**(4), 280–287. DOI:10.1007/BF01086739.

Zupanski, M. 2005. Maximum likelihood ensemble filter: theoretical aspects. *Mon. Wea. Rev.* **133**(6), 171–1726. DOI:10.1175/MWR2946.1.

Appendix 1. Localisation in 4DEnVar

There are two issues associated with localisation in 4DEnVar.

i. The way we have written our formulation in terms of diagonalisations and matrix products –following Buehner (2005)– is very clear from an algorithmic point of view. However, the computational implementation is much more efficient using weighted linear combinations of ensemble members. We can write the observational contribution from time t to the cost function as:

$$(\hat{\mathbf{Y}}^{b,t})^{\mathbf{T}}\mathbf{R}^{-1}(\mathbf{d}^t - \hat{\mathbf{Y}}^{b,t}\mathbf{v}^0) = \begin{bmatrix} (\mathbf{L}_{\mathbf{y}}^{1/2})^{\mathbf{T}}(\hat{\mathbf{y}}_1^{b,t} \circ \mathbf{z}_{\mathbf{y}}) \\ (\mathbf{L}_{\mathbf{y}}^{1/2})^{\mathbf{T}}(\hat{\mathbf{y}}_2^{b,t} \circ \mathbf{z}_{\mathbf{y}}) \\ \vdots \\ (\mathbf{L}_{\mathbf{y}}^{1/2})^{\mathbf{T}}(\hat{\mathbf{y}}_{N_e}^{b,t} \circ \mathbf{z}_{\mathbf{y}}) \end{bmatrix} \tag{53}$$

where $\mathbf{z}_{\mathbf{y}} \in \mathcal{R}^{rN_e}$ is defined as:

$$\mathbf{z}_{\mathbf{y}} = \mathbf{R}^{-1}\left(\mathbf{d}^t - \sum_{n_e=1}^{N_e} \hat{\mathbf{y}}_{n_e}^t \circ (\mathbf{L}_{\mathbf{y}}^{1/2}\mathbf{v}_{n_e}^0)\right) \tag{54}$$

We can implement a localised version of WC4DEnVar if we use the same matrices $\hat{\hat{\mathbf{X}}}^{b,t}$ and $\hat{\hat{\mathbf{Y}}}^{b,t}$ that we defined for the SC case.

The observational contributions in the gradient of the cost function can be efficiently computed as in (53). For the model error contributions at time t, we compute the contributions as:

$$(\hat{\hat{\mathbf{X}}}^{b,t})^{\mathbf{T}}(\mathbf{Q}^t)^{-1}\hat{\hat{\mathbf{X}}}^{b,t}\mathbf{v}^t = \begin{bmatrix} (\mathbf{L_x}^{1/2})^{\mathbf{T}}(\mathfrak{X}_1^t \circ \mathbf{z_x}) \\ (\mathbf{L_x}^{1/2})^{\mathbf{T}}(\mathfrak{X}_2^t \circ \mathbf{z_x}) \\ \vdots \\ (\mathbf{L_x}^{1/2})^{\mathbf{T}}(\mathfrak{X}_{N_e}^t \circ \mathbf{z_x}) \end{bmatrix} \tag{55}$$

where $\mathbf{z_x} \in \mathcal{R}^{rN_e}$ can be found as:

$$\mathbf{z_x} = (\mathbf{Q}^t)^{-1}\sum_{n_e=1}^{N_e}\mathbf{x}_{n_e}^t \circ (\mathbf{L_x}^{1/2}\mathbf{v}_{n_e}^t) \tag{56}$$

ii. Localisation increases the size of the EnVar problem. The localised version of (17) is written as:

$$\nabla_{\mathbf{v}^0}J(\mathbf{v}^0) = \mathbf{v}^0 - \sum_{t=1}^{\tau}\rho^t(\hat{\hat{\mathbf{Y}}}^{b,t})^{\mathbf{T}}\mathbf{R}^{-1}\left(\mathbf{d}^t - \hat{\hat{\mathbf{Y}}}^{b,t}\mathbf{v}^0\right) \tag{57}$$

where the size of the control variable $\mathbf{v}^0$ has now increased to $\mathbf{v}^0 \in \mathcal{R}^{N_eN_g}$, assuming that $rank(\mathbf{L_{xx}}) = N_g$ and writing $\mathbf{v}^0 = [\mathbf{v}_1^0, \cdots, \mathbf{v}_{N_e}^0]^{\mathbf{T}}$, with $\mathbf{v}_{n_e}^0 \in \mathcal{R}^{N_g}$. The increase in size is not a serious problem, however, if we realise that N_eN_g is an only upper bound, and in fact it is often easier to work with a truncated matrix square root.

Recall that

$$\mathbf{L_{xx}} = \mathbf{C}\boldsymbol{\Gamma}\mathbf{C}^{\mathbf{T}} \rightarrow \mathbf{L_x}^{1/2} = \mathbf{C}\boldsymbol{\Gamma}^{\boldsymbol{1/2}}$$

and let us discuss how fast the eigenvalues $\{\gamma_n\}$ decay. For this, let us consider the half-width values in three cases:

a. The case $\lambda \rightarrow \infty$ corresponds to no localisation, where $\mathbf{L_{xx}}$ becomes a matrix of ones. The rank of this matrix is 1. There is only a non-zero eigenvalue $\gamma_1 = 1$ and only one eigenvector, corresponding to the vector of ones. This is clearly equal to doing no localisation at all.

b. The case $\lambda = 0$ corresponds to a strict localisation where variables in a grid point are only affected by observation at that very grid point. Then $\mathbf{L_{xx}}$ becomes the identity matrix. The rank of this matrix is N_x. There are N_x non-zero eigenvalues, all of them with a value of 1, and N_x eigenvectors.

c. Localisation matrices with finite $\lambda \neq 0$ fall somewhere in the middle. For the type of localisation used in this paper –where only the distance amongst grid points determines the localisation value– $\mathbf{L_{xx}}$is a circulant matrix, and it follows that in this case the columns of $\mathbf{C}$ are Fourier modes: column 1 corresponds to a constant (zeroth harmonic), columns 2 and 3 correspond to the first harmonic (one column for sine and one for cosine), columns 4 and 5 correspond to the second harmonic, and so forth. The associated eigenvalues in Γ decrease as the order of the harmonic increases. Then, one can retain only the first $1+2k$ columns of $\mathbf{C}$, and the first $1+2k$ eigenvalues of $\boldsymbol{\Gamma}$, rendering $\mathbf{v}^0 \in \mathcal{R}^{(1+2k)N_e}$.

Let us illustrate this in Fig. 8 with an example. For a system with $N_x = 100$ grid points, we generate localisation matrices $\mathbf{L_{xx}}$ using different half-widths $\lambda = 1, 2, 3, 4, 5$ (shown with different line styles in the figure), and we perform their spectral decompositions. The top panel shows the eigenvalues, in decreasing order of magnitude, for each one of the different half-widths. The decay in magnitude is slow for $\lambda = 1$, but it becomes considerably faster as the half-width increases. The bottom panel shows the cumulative sum of these eigenvalues. For λ>1 it quickly approaches its maximum value $\left(\sum_{n=1}^{N_x}\lambda_n = N_x\right)$, and many eigenvalues contribute to this sum only marginally, especially for larger localisation half-widths.

Small eigenvalues can be discarded in favour of speeding up the computation time, creating a simpler minimisation problem. The minimisation becomes easier for two reasons. First the size of the control vector decreases. Second, considering small grid point-per-grid point variations (a consequence of keeping all eigenvalues) can result in a more difficult minimisation problem.

To determine the number of eigenvalues to keep, a simple idea is to chose the value n that fulfils:

$$\sum_{j=1}^{n}\gamma_j = cN_x \tag{58}$$

where the eigenvalues γ_j are ordered by decreasing magnitude, and 0<c<1 is a fraction of our choice. The closer to one this fraction is, the closer our cropped localisation matrix will be to the original one. The N_x in the right hand side of (58) comes from the fact that $\sum_{j=1}^{N_x}\gamma_j = N_x$.

In Fig. 9 we have illustrated this choice for two fractions: 0.5in the left panel and 0.9 in the right panel. For different total number of grid points N_x (lines with different colors), we show the ratio $\frac{n}{N_x}$ required as a function of the localisation radius λ (half-width of the Gaspari Cohn function). As expected this normalized quantity depends only on the localisation radius, and it is a decreasing function. This corresponds perfectly with our previous discussion of the two limiting cases.

An example of the reduction in computational cost can be seen, for example, at $\lambda = 4$. If our desired fraction is 0.9, we would need to keep around 11% of the total number of eigenvalues. If our desired fraction were 0.5 the required fraction would be reduced to 5%.

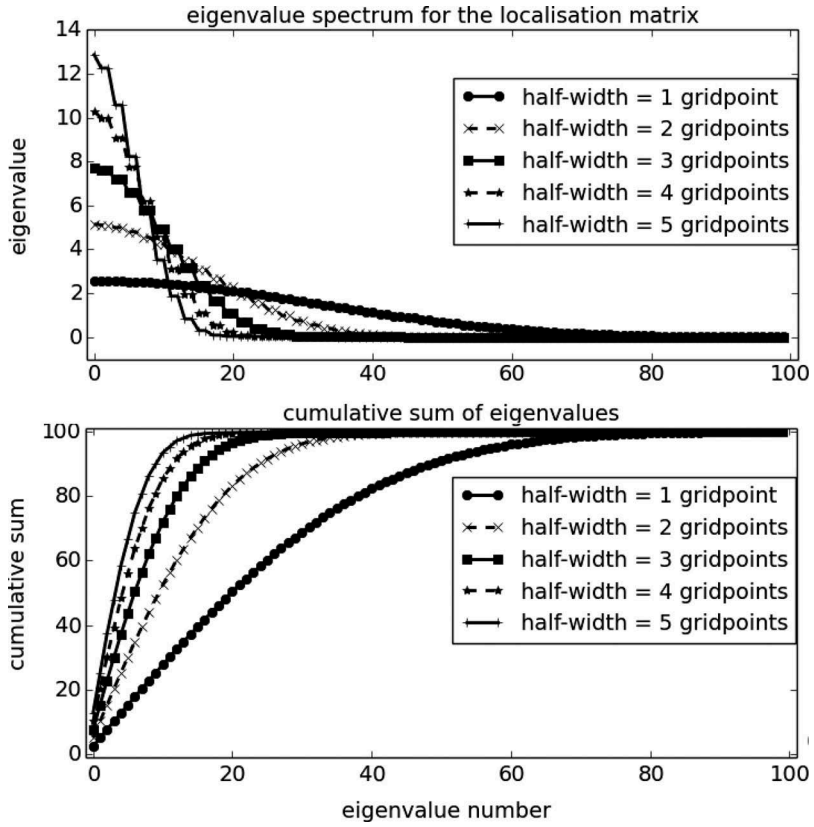

Fig. 8. Eigenvalue spectrum of a localisation matrix $\mathbf{L_{xx}}$ for $N_x = 100$ grid points, with different localisation half-widths (different lines). The top panel shows the eigenvalues in descending order, while the bottom panel shows their cumulative sum.

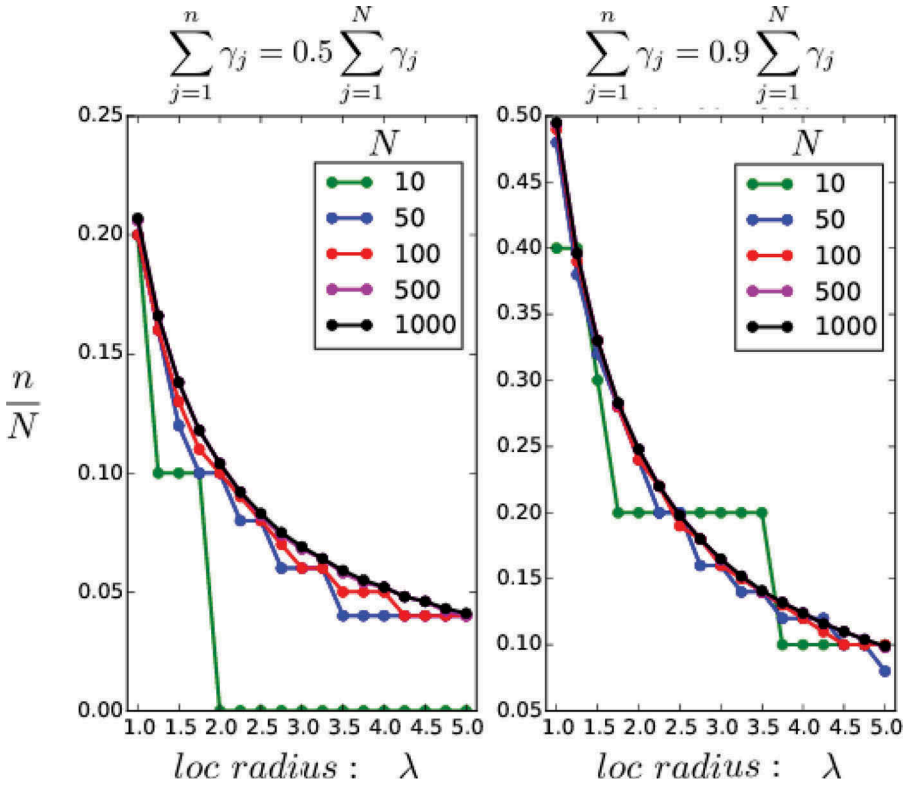

Fig. 9. Construction of $\mathbf{L_{xx}}$ in SC and WC4DEnVar. For different number of grid points N and different localisation half widths (horizontal axis), the vertical axis shows the ratio of retained eigenvalues n over N, in order to cover a given fraction of the total sum of eigenvalues. For the left panel this fraction is 0.5, in the right panel it is 0.9.

Appendix 2. Generating $\mathbf{B}^c$

The method used in Yang et al. (2006) is is outlined next. The purpose is to create a climatological background error covariance matrix in an iterative manner.

1. The process starts by proposing a guess matrix $\mathbf{B}^{c0}$. This can be as simple as the identity matrix. However, we start with a circulant matrix, with the j^{th} row being $\mathbf{B}^{c0}[j,:] = [\cdots, 0, 0.25, 0.5, 1, 0.5, 0.25, \cdots]$.

2. We perform a 3DVar assimilation experiment using observations every 5 model time steps, with all grid points observed being observed. This is done for 100 assimilation cycles, and we use the proposed $\mathbf{B}^{c0}$.

3. At this point we have 100 forecast values (valid at the assimilation instants) which can be compared to the nature run. We use these values to find a full-rank sample estimator $\mathbf{P}^b \rightarrow \mathbf{B}^c$.

4. We replace our previous value of $\mathbf{B}^c$ with the one just computed from sample statistics and repeat steps 2 and 3; we perform 10 iterations (in fact, we found convergence after 5).

5. We repeat the whole procedure 20 times, each with a different set of pseudo-observations. We average the 20 estimators and that yields the final estimator of $\mathbf{B}^c$.