



A 4D-Var method with flow-dependent background covariances for the shallow-water equations

Daniel Paulin¹ · Ajay Jasra² · Alexandros Beskos³ · Dan Crisan⁴

Received: 16 January 2021 / Accepted: 5 June 2022 / Published online: 11 August 2022
© The Author(s) 2022

Abstract

The 4D-Var method for filtering partially observed nonlinear chaotic dynamical systems consists of finding the maximum a-posteriori (MAP) estimator of the initial condition of the system given observations over a time window, and propagating it forward to the current time via the model dynamics. This method forms the basis of most currently operational weather forecasting systems. In practice the optimisation becomes infeasible if the time window is too long due to the non-convexity of the cost function, the effect of model errors, and the limited precision of the ODE solvers. Hence the window has to be kept sufficiently short, and the observations in the previous windows can be taken into account via a Gaussian background (prior) distribution. The choice of the background covariance matrix is an important question that has received much attention in the literature. In this paper, we define the background covariances in a principled manner, based on observations in the previous b assimilation windows, for a parameter $b \geq 1$. The method is at most b times more computationally expensive than using fixed background covariances, requires little tuning, and greatly improves the accuracy of 4D-Var. As a concrete example, we focus on the shallow-water equations. The proposed method is compared against state-of-the-art approaches in data assimilation and is shown to perform favourably on simulated data. We also illustrate our approach on data from the recent tsunami of 2011 in Fukushima, Japan.

Keywords Filtering · Smoothing · Data assimilation · Gauss-Newton method · Shallow-water equations

1 Introduction

Filtering, or data assimilation, is a field of core importance in a wide variety of real applications, such as numerical weather forecasting, climate modelling and finance; see e.g. Asch et al. (2016), Blayo et al. (2014), Crisan (2017), Lahoz et al. (2010), Law et al. (2015) for an introduction. Informally, one is interested in carrying out inference about an unobserved signal process conditionally upon noisy observations. The type of unobserved process considered in this paper is that of a nonlinear chaotic dynamical system, with unknown initial condition. As an application in this paper we consider the case where the unobserved dynamics correspond to the discretised version of the shallow-water equations; see e.g. Salmon (2015). These latter equations are of great practical importance, generating realistic approximations of real-world phenomena, useful in tsunami and flood modelling (see e.g. Bates et al. (2010), Pelinovsky (2006)).

For systems of this type, the filtering problem is notoriously challenging. Firstly, the filter is seldom available in analytic form due to the nonlinearity. Secondly, even if the

✉ Daniel Paulin
dpaulin@ed.ac.uk
Ajay Jasra
ajay.jasra@kaust.edu.sa
Alexandros Beskos
a.beskos@ucl.ac.uk
Dan Crisan
d.crisan@ic.ac.uk

¹ School of Mathematics, University of Edinburgh, James Clerk Maxwell Building, Peter Guthrie Tait Rd, Edinburgh EH9 3FD, UK

² Computer, Electrical and Mathematical Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955-6900, Kingdom of Saudi Arabia

³ Department of Statistical Science, University College London, London WC1E 6BT, UK

⁴ Department of Mathematics, Imperial College London, London SW7 2AZ, UK

given system is solvable, the associated dimension of the object to be filtered is very high (of order of 10^8 or greater), thus posing great computational challenges.

One of the most successful methods capable of handling such high-dimensional datasets is the so-called 4D-Var algorithm (Le Dimet and Talagrand 1986; Talagrand and Courtier 1987): it consists of optimising a loss-functional so that under Gaussian noise it is equivalent to finding the maximum a-posteriori (MAP) estimator of the initial condition. Since its introduction, a lot of further developments in the 4D-Var methodology have appeared in the literature; for an overview of some recent advances, we refer the reader to Bannister (2016), Lorenc (2014), Navon (2009), Park and Xu (2009), Park and Xu (2013), Park and Xu (2017). The main focus of this article is to consider principled improvements of the 4D-Var algorithm.

An important practical issue of the 4D-Var method is that, due to chaotic nature of the systems encountered in weather prediction, the negative log-likelihood (cost function) can become highly non-convex if the assimilation window is too long. The reason for this is that for deterministic dynamical systems, as the assimilation window grows, the smoothing distribution gets more and more concentrated on the stable manifold of the initial position, which is a complicated lower-dimensional set (see Pires et al. 1996, Paulin et al. 2018a for more details). On the one hand, this means that it becomes very difficult to find the MAP estimator. On the other hand, due to the highly non-Gaussian nature of the posterior in this setting, the MAP might be far away from the posterior mean, and have a large mean square error. Moreover, for longer windows, the precision of the tangent linear model/adjoint solvers might decrease. Due to these facts, the performance of 4D-Var deteriorates for many models when the observation window becomes too long (see Kalnay et al. 2007).

The observations in the previous window are taken into account via the background (prior) distribution, which is a Gaussian whose mean is the estimate of the current position based on the previous windows, and has a certain covariance matrix. The choice of this background covariance matrix is an important and difficult problem that has attracted much research. (Fisher 2003) states that in operational weather forecasting systems up to 85% of the information in the smoothing distribution comes from the background (prior) distribution. The main contribution of this paper is an improvement of the 4D-Var methodology by a principled definition of this matrix in a flow-dependent way. This is based on the observations in the previous b assimilation windows (for a parameter $b \geq 1$). Via simulations on the shallow-water model, we show that our method compares favourably in precision with the state-of-the-art Hybrid 4D-Var method (see Lorenc et al. 2015).

The structure of the paper is as follows. In the rest of this section, we briefly review the literature on 4D-Var back-

ground covariances. In Sect. 2, the modelling framework for the shallow-water equations is described in detail. In Sect. 3, we introduce our 4D-Var method with flow-dependent background covariance. In particular, Sect. 1.1 compares our method with other choices of flow-dependent background covariances in the literature. In Sect. 4, we present some simulation results and compare the performance of our method with Hybrid 4D-Var. Finally, in Sect. 5 we state some conclusions for this paper.

1.1 Comparison with the literature

There exist mathematically rigorous techniques to obtain the filter with precision and use the mean of the posterior distribution as the estimate, based upon sequential Monte Carlo methods (e.g. Del Moral et al. (2006),Rebeschini et al. (2015)) which can provably work in high-dimensional systems Beskos et al. (2014). While these approximate the posterior means and hence are optimal in mean square error, and are possibly considerably more accurate than optimisation-based methods, nonetheless such methodology can be practically overly expensive. As a result, one may have to resort to less accurate but more computationally efficient methodologies (see Law et al. (2015) for a review). There are some relatively recent applications of particle filtering methods to high-dimensional data assimilation problems, see, e.g., (van Leeuwen (2009, 2010)). While these algorithms seem to be promising for certain highly nonlinear problems, their theoretical understanding is limited at the moment due to the bias they introduce via various approximations.

Despite the difficulty of solving the nonlinear filtering problem exactly, due to the practical interest in weather prediction, several techniques have been devised and implemented operationally in weather forecasting centres worldwide. These techniques are based on optimisation methods, and hence they scale well to high dimensions and are able to process massive datasets. Although initially such methods were lacking in mathematical foundation, the books (Bengtsson et al. (1981)) and (Kalnay (2003)) are among the first to open up the field of data assimilation to mathematics. Among the earlier works devoted to the efforts of bringing together data assimilation and mathematics, we also mention (Ghil et al. 1981) and (Ghil and Malanotte-Rizzoli (1991)), where a comparison between the Kalman filter (sequential-estimation) and variational methods is presented.

The performance of 4D-Var methods depends very strongly on the choice of background covariances. One of the first principled ways of choosing 4D-Var background covariances was introduced by Parrish and Derber (1992). They have proposed the so-called NMC method for choosing climatological prior error covariances based on a comparison of 24 and 48 hour forecast differences. This method was refined in Derber and Bouttier (1999). (Fisher (2003)) proposed the

use of wavelets for forming background covariances; these retain the computational advantages of spectral methods, while also allow for spatial inhomogeneity. The background covariances are made flow dependent via a suitable modification of the NMC approach. (Lorenz (2003)) reviews some of the practical aspects of modelling 4D-Var error covariances, while (Fisher et al. (2005)) makes a comparison between 4D-Var for long assimilation windows and the Extended Kalman Filter. As we have noted previously, long windows are not always applicable due to the presence of model errors and the non-convexity of the likelihood.

More recently, there have been several methods proposing the use of ensembles combined with localisation methods for modelling the covariances, see, e.g. Zupanski (2005), Auligné et al. (2016), Bannister (2016), Bannister (2008a), Bannister (2008b), Bonavita et al. (2016), Bousserez et al. (2015), Buehner (2005) Clayton et al. (2013), Hamill et al. (2011), Kuhl et al. (2013), Wang et al. (2013). Currently most operational NWP centres use the Hybrid 4D-Var method, which is based on a linear combination of a fixed background covariance matrix (the climatological background error covariance) and an ensemble-based background covariance (see Lorenz et al. (2015), Fairbairn et al. (2014)).

Localisation eliminates spurious correlations between elements of the covariance matrix that are far away from each other, and hence they have little correlation. This means that these long range correlations are set to zero, which allows the sparse storage of the covariance matrix and efficient computations of the matrix-vector products. Bishop et al 2011 proposes an efficient implementation of localisation by introducing some further approximations, using the product structure of the grid, and doing part of the calculations on lower resolution grid points. Such efficient implementations have allowed localised ENKF-based background covariance modelling to provide the state-of-the-art performance in data assimilation, and they form the core of most operational NWP systems at the moment.

Over longer time periods, given sufficient data available, most of the variables become correlated, and imposing a localised structure over them leads to some loss of information vs the benefit of computational efficiency. Our method does not impose such a structure as it writes the precision matrix in a factorised form. Moreover, the localisation structure is assumed to be fixed in time, so even with a considerable amount of tuning for a certain time period of data it is not guaranteed that the same localisation structure will be optimal in the future. Our method does not make such a constant localisation assumption and hence it is able to adapt to different correlation structures automatically.

We use an implicit factorised form of the Hessian and the background precision matrix described in Sects. 3.2–3.3, and thus we only need to store the positions of the system started from the 4D-Var estimate of the previous b windows

at the observation times. This allows us to compute the effect of these matrices on a vector efficiently, without needing to store all the elements of the background precision matrix, which would require too much memory.

Although in this paper we have assumed that the model is perfect, there have been efforts to account for model error in the literature, see (Trémolet 2006). The effect of nonlinearities in the dynamics and the observations can be in some cases so strong that the Gaussian approximations are no longer reasonable, see (Miller et al. 1994; Bocquet et al. 2010; Gejadze et al. 2011) for some examples and suggestions for overcoming these problems.

2 Notations and model

2.1 Notations

In this paper, we will be generally using the unified notations for data assimilation introduced in Ide et al. (1997). In this section, we briefly review the required notations for the 4D-Var data assimilation method.

The state vector at time t will be denoted by $\mathbf{x}(t)$, and it is assumed that it has dimension n . The evolution of the system from time s to time t will be governed by the equation

$$\mathbf{x}(t) = M(t, s)[\mathbf{x}(s)], \tag{2.1}$$

where $M(t, s)$ is the model evolution operator from time s to time t . In practice, this finite-dimensional model is usually obtained by discretisation of the full partial differential equations governing the flow of the system.

Observations are made at times $(t_i)_{i \geq 0}$, and they are of the form

$$\mathbf{y}_i^\circ = H_i[\mathbf{x}(t_i)] + \varepsilon_i, \tag{2.2}$$

where H_i is the observation operator, and ε_i is the random noise. We will denote the dimension \mathbf{y}_i° by n_i° , and assume that $(\varepsilon_i)_{i \geq 0}$ are independent normally distributed random vectors with mean 0 and covariance matrix $(\mathbf{R}_i)_{i \geq 0}$. The Jacobian matrix (i.e. linearisation) of the operator $M(t, s)$ at position $\mathbf{x}(s)$ will be denoted by $\mathbf{M}(t, s)$, and the Jacobian of H_i at $\mathbf{x}(t_i)$ will be denoted by \mathbf{H}_i . The inverse and transpose of a matrix will be denoted by $(\cdot)^{-1}$ and $(\cdot)^T$, respectively.

The 4D-Var method for assimilating the observations in the time interval $[t_0, t_{k-1}]$ consists of minimising the cost functional

$$J[\mathbf{x}(t_0)] = \frac{1}{2}[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)]^T \mathbf{B}_0^{-1}[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)] + \frac{1}{2} \sum_{i=0}^{k-1} [\mathbf{y}_i - \mathbf{y}_i^\circ]^T \mathbf{R}_i^{-1}[\mathbf{y}_i - \mathbf{y}_i^\circ], \tag{2.3}$$

where $y_i := H_i(x(t_i))$, and B_0 denotes the background covariance matrix, and $x^b(t_0)$ denotes the background mean. Minimising this functional is equivalent to maximising the likelihood of the smoothing distribution for $x(t_0)$ given $y_{0:k-1}^\circ := \{y_0^\circ, \dots, y_{k-1}^\circ\}$ and normally distributed prior with mean $x^b(t_0)$ and covariance B_0 . Note that the cost function (2.3) corresponds to the so-called *strong constraint* 4D-Var (i.e., no noise is allowed in the dynamics), there are also *weak constraint* alternatives that account for possible model errors by allowing noise in the dynamics (see e.g. Trémolet 2006).

2.2 The model

We consider the shallow-water equations, e.g. as described in [pg. 105-106]Salmon (2015), but with added diffusion and bottom friction terms, i.e.

$$\frac{\partial u}{\partial t} = \left(-\frac{\partial u}{\partial y} + f\right)v - \frac{\partial}{\partial x} \left(\frac{1}{2}u^2 + gh\right) + v\nabla^2 u - c_b u; \tag{2.4}$$

$$\frac{\partial v}{\partial t} = -\left(\frac{\partial v}{\partial x} + f\right)u - \frac{\partial}{\partial y} \left(\frac{1}{2}v^2 + gh\right) + v\nabla^2 v - c_b v; \tag{2.5}$$

$$\frac{\partial h}{\partial t} = -\frac{\partial}{\partial x}((h + \underline{h})u) - \frac{\partial}{\partial y}((h + \underline{h})v). \tag{2.6}$$

Here, u and v are the velocity fields in the x and y directions, respectively, and h the field for the height of the wave. Also, \underline{h} is the depth of the ocean, g the gravity constant, f the Coriolis parameter, c_b the bottom friction coefficient and ν the viscosity coefficient. Parameters \underline{h} , f , c_b and ν are assumed to be constant in time but in general depend on the location. The total height of the water column is the sum $\underline{h} + h$.

For square grids, under periodic boundary conditions, the equations are discretised as

$$\begin{aligned} \frac{du_{i,j}}{dt} &= f_{i,j}v_{i,j} - \frac{g}{2\Delta}(h_{i+1,j} - h_{i-1,j}) \\ &- c_b u_{i,j} + \frac{\nu}{\Delta^2}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}) \\ &- \frac{1}{2\Delta}[(u_{i,j+1} - u_{i,j-1})v_{i,j} + (u_{i+1,j} - u_{i-1,j})u_{i,j}], \end{aligned} \tag{2.7}$$

$$\begin{aligned} \frac{dv_{i,j}}{dt} &= -f_{i,j}u_{i,j} - \frac{g}{2\Delta}(h_{i,j+1} - h_{i,j-1}) \\ &- c_b v_{i,j} + \frac{\nu}{\Delta^2}(v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1} - 4v_{i,j}) \\ &- \frac{1}{2\Delta}[(v_{i+1,j} - v_{i-1,j})u_{i,j} + (v_{i,j+1} - v_{i,j-1})v_{i,j}], \end{aligned} \tag{2.8}$$

$$\begin{aligned} \frac{dh_{i,j}}{dt} &= \\ &- \frac{1}{2\Delta}(h_{i,j} + \underline{h}_{i,j})(u_{i+1,j} - u_{i-1,j} + v_{i,j+1} - v_{i,j-1}) \\ &- \frac{1}{2\Delta}u_{i,j}(h_{i+1,j} + \underline{h}_{i+1,j} - h_{i-1,j} - \underline{h}_{i-1,j}) \\ &- \frac{1}{2\Delta}v_{i,j}(h_{i,j+1} + \underline{h}_{i,j+1} - h_{i,j-1} - \underline{h}_{i,j-1}), \end{aligned} \tag{2.9}$$

where $1 \leq i, j \leq d$, for a typically large $d \in \mathbb{Z}_+$, with the indices understood modulo d (hence the domain is a torus), and some space-step $\Delta > 0$. Summing up (2.9) over $1 \leq i, j \leq d$, one can see that the discretisation preserves the total mass $h_{\text{tot}} := \sum_{i,j}(h_{i,j} + \underline{h}_{i,j})$. If we assume that the viscosity and bottom friction are negligible, i.e. $\nu = c_b = 0$, then the total energy

$$E_{\text{tot}} := \frac{1}{2} \sum_{i,j} \left((h_{i,j} + \underline{h}_{i,j})u_{i,j}^2 + (h_{i,j} + \underline{h}_{i,j})v_{i,j}^2 + g(h_{i,j}^2 - \underline{h}_{i,j}^2) \right)$$

is also preserved. When the coefficients c_b and ν are not zero, the bottom friction term always decreases the total energy (the sum of the kinetic and potential energy), while the diffusion term tends to smooth the velocity profile. We denote the solution of Eqs. (2.7-2.9) at time $t \geq 0$ as

$$x(t) := \left((u_{i,j}(t))_{1 \leq i,j \leq d}, (v_{i,j}(t))_{1 \leq i,j \leq d}, (h_{i,j}(t))_{1 \leq i,j \leq d} \right).$$

The unknown and random initial condition is denoted by $x(0)$. One can show by standard methods (see Murray and Miller 2013) that the solution of (2.7-2.9) exists up to some time $T_{\text{sol}}(x(0)) > 0$. In order to represent the components of $x(t)$, we introduce a vector index notation. The set $\mathcal{I} := \{u, v, h\} \times \{1, \dots, d\} \times \{1, \dots, d\}$ denotes the possible indices, with the first component referring to one of u, v, h , the second component to coordinate i , and the third to j . A vector index in \mathcal{I} will usually be denoted as m or n , e.g. if $m = (u, 1, 2)$, then $x_m(t) := u_{1,2}(t)$.

We assume that the $n := 3d^2$ -dimensional system is observed at time points $(t_l)_{l \geq 0}$, with observations described as in Section 2.1. The aim of smoothing and filtering is to approximately reconstruct $x(t_0)$ and $x(t_k)$ based on observations $y_{0:k-1}^\circ$. We note that data assimilation for the shallow-water equations has been widely studied in the literature, see (Bengtsson et al. 1981), (Egbert et al. 1994) for the linearised form and (Lyard et al. 2006), (Courtier and Talagrand 1990), (Ghil and Malanotte-Rizzoli 1991) for the full nonlinear form of the equations.

3 4D-Var with flow-dependent covariance

3.1 Method Overview

Assume that observations $y_{0:k-1}^\circ$ are made at time points $t_l = t_0 + lh$ for $l = 0, \dots, k - 1$, and let $T := kh$. The 4D-Var method for assimilating the observations in the time interval $[t_0, t_{k-1}]$ consists of minimising the cost functional (2.3). Under the independent Gaussian observation error assumption, $-J[x(t_0)]$ is the log-likelihood of the smoothing distribution, ignoring the normalising constant. The minimiser of J is the MAP estimator, and is denoted by \hat{x}_0 (if multiple such minimisers exist, then we choose any of them). A careful choice of the background distribution is essential, especially in the case when the total number of observations in the assimilation window is smaller than the dimension of the dynamical system, where without the prior distribution, the likelihood would be singular (see Dashti and Stuart 2016 for a principled method of choosing priors).

To obtain the MAP estimator, we make use of Newton’s method. Starting from some appropriate initial position $x_0 \in \mathbb{R}^n$, the method proceeds via the iterations

$$x_{l+1} = x_l - \left(\frac{\partial^2 J}{\partial x_l^2} \right)^{-1} \frac{\partial J}{\partial x_l}, \quad l \geq 0, \tag{3.1}$$

where $\frac{\partial J}{\partial x_l}$ and $\frac{\partial^2 J}{\partial x_l^2}$ denote the gradient and Hessian of J at x_l , respectively. Due to the high dimensionality of the systems in weather forecasting, typically iterative methods such as the preconditioned conjugate gradient (PCG) are used for evaluating (3.1). The iterations are continued until the step size $\|x_{l+1} - x_l\|$ falls below a threshold $\delta_{\min} > 0$. The final position is denoted by \hat{x}_* , and this is the numerical estimate for \hat{x}_0 - with its push-forward $M(t_k, t_0)[\hat{x}_*]$ then being the numerical estimate for $M(t_k, t_0)[\hat{x}_0]$.

To apply the iterations (3.1), one needs to compute the gradient and the Hessian of J (or, more precisely, the application of the latter to a vector, which is all that is required for iterative methods such as PCG). An efficient method for doing this is given in the next section. In practice, one cannot apply the above optimisation procedure for arbitrarily large k due to the non-convexity of the smoothing distribution for big enough k (due to the nonlinearity of the system). Therefore, we need to partition the observations into blocks of size k for some reasonably small k , and apply the procedure on them separately. The observations in the previous blocks can be taken into account by appropriately updating the prior distribution. The details of this procedure are explained in Section 3.3. Finally, in Section 1.1 we compare our method with other choices of flow-dependent background covariances in the literature.

3.2 Gradient and Hessian calculation

We can rewrite the gradient and Hessian of the cost function J at a point $x(t_0) \in \mathbb{R}^n$ as

$$\begin{aligned} \frac{\partial J}{\partial x(t_0)} &= B_0^{-1} [x(t_0) - x^b(t_0)] \\ &\quad - \sum_{l=0}^{k-1} M(t_l, t_0)^T H_l^T R_l^{-1} (y_l - y_l^\circ), \\ \frac{\partial^2 J}{\partial x(t_0)^2} &= B_0^{-1} + \sum_{l=0}^{k-1} M(t_l, t_0)^T H_l^T R_l^{-1} H_l M(t_l, t_0) \end{aligned} \tag{3.2}$$

$$\begin{aligned} &\quad - \sum_{l=0}^{k-1} \left(\frac{\partial^2 M(t_l, t_0)}{\partial x_0^2} \right)^T H_l^T R_l^{-1} (y_l - y_l^\circ) \\ &\quad - \sum_{l=0}^{k-1} \left(M(t_l, t_0)^T \right)^2 \left(\frac{\partial^2 H_l}{\partial x(t_l)^2} \right)^T R_l^{-1} (y_l - y_l^\circ). \end{aligned} \tag{3.3}$$

These can be obtained either directly, or by viewing J as a free quadratic function with (2.1) and (2.2) as constraints.

Let $M_l := M(t_l, t_{l-1})$, then $M(t_l, t_0) = M_l \cdot \dots \cdot M_1$, so the sum in the gradient (3.2) can be rewritten as

$$\begin{aligned} &\sum_{l=0}^{k-1} M(t_l, t_0)^T H_l^T R_l^{-1} (y_l - y_l^\circ) \\ &= \sum_{l=0}^{k-1} M_1^T \cdot \dots \cdot M_l^T H_l^T R_l^{-1} (y_l - y_l^\circ) \end{aligned} \tag{3.4}$$

The above summation can be efficiently performed as follows. We consider the sequence of vectors

$$\begin{aligned} g_{k-1} &:= H_{k-1}^T R_{k-1}^{-1} (y_{k-1} - y_{k-1}^\circ); \\ g_l &:= H_l^T R_l^{-1} (y_l - y_l^\circ) + M_{l+1}^T g_{l+1}, \quad k-1 > l \geq 0. \end{aligned}$$

The sum on the right side of (3.4) then equals g_0 . We note that this method of computing the gradients forms the basis of the *adjoint* method, introduced in Talagrand and Courtier (1987), see also (Talagrand 1997).

In the case of the Hessian, in (3.3) there are also second-order Jacobian terms. If $x(t_0)$ is close to the true initial position, then $(y_l - y_l^\circ) \approx \varepsilon_l$. Therefore in the low-noise/high-frequency regime, given a sufficiently precise initial estimator, these second-order terms can be neglected. Using such Hessian corresponds to the so-called Gauss–Newton method, which has been studied in the context of 4D-Var in Gratton et al. (2007). Thus, we use the approximation

$$\frac{\widehat{\partial^2 J}}{\partial \mathbf{x}(t_0)^2} := \mathbf{B}_0^{-1} + \sum_{l=0}^{k-1} \mathbf{M}(t_l, t_0)^T \mathbf{H}_l^T \mathbf{R}_l^{-1} \mathbf{H}_l \mathbf{M}(t_l, t_0) \tag{3.5}$$

A practical advantage of removing the second-order terms is that if the Hessian of the log-likelihood of the prior, \mathbf{B}_0 is positive definite, then the resulting sum is positive definite, so the direction of $-\left(\frac{\widehat{\partial^2 J}}{\partial \mathbf{x}(t_0)^2}\right)^{-1} \cdot \frac{\partial J}{\partial \mathbf{x}(t_0)}$ is always a direction of descent (which is not always true if the second-order terms are included). Note that via the so-called second-order adjoint equations, it is possible to avoid this approximation, and compute the action of the Hessian $\frac{\partial^2 J}{\partial \mathbf{x}(t_0)^2}$ on a vector in $O(n)$ time, see (Le Dimet et al. 2002). However, this can be slightly more computationally expensive, and in our simulations the Gauss-Newton approximation (3.5) worked well.

For the first order terms in the Hessian, for any $\mathbf{w} \in \mathbb{R}^n$, we have

$$\begin{aligned} & \sum_{l=0}^{k-1} \mathbf{M}(t_l, t_0)^T \mathbf{H}_l^T \mathbf{R}_l^{-1} \mathbf{H}_l \mathbf{M}(t_l, t_0) \mathbf{w} \\ &= \sum_{l=0}^{k-1} \mathbf{M}_1^T \mathbf{M}_2^T \dots \mathbf{M}_l^T \mathbf{H}_l^T \mathbf{R}_l^{-1} \mathbf{H}_l \mathbf{M}_l \mathbf{M}_{l-1} \dots \mathbf{M}_1 \mathbf{w}. \end{aligned} \tag{3.6}$$

We define

$$\mathbf{w}_l := \mathbf{M}_l \dots \mathbf{M}_1 \mathbf{w}, \quad l = 0, \dots, k - 1;$$

and consider the sequence of vectors

$$\begin{aligned} \mathbf{h}_{k-1} &= \mathbf{H}_{k-1}^T \mathbf{R}_{k-1}^{-1} \mathbf{H}_{k-1} \mathbf{w}_{k-1}; \\ \mathbf{h}_l &= \mathbf{H}_l^T \mathbf{R}_l^{-1} \mathbf{H}_l \mathbf{w}_l + \mathbf{M}_{l+1}^T \mathbf{h}_{l+1}, \quad k - 1 > l \geq 0. \end{aligned} \tag{3.7}$$

Then the sum on the right side of (3.6) equals \mathbf{h}_0 . The Hessian plays an important role in practical implementations of the 4D-Var method, and several methods have been proposed for its calculation (see Courtier et al. 1994; Le Dimet et al. 2002; Lawless et al. 2005). Due to computational considerations, usually some approximations such as lower resolution models are used when computing Hessian-vector products for Krylov subspace iterative solvers in practice (this is the so-called incremental 4D-Var method, see Courtier et al. 1994). Note that it is also possible to use inner and outer loops, where in the inner loops both the Hessian-vector products and the gradient are run on lower resolution models, while in the outer loops we use the high resolution model for the gradient, and lower resolution model for the Hessian-vector products. (Lawless et al. 2005) has studied the theoretical properties of this approximation. In practice, the speedup from this method can be substantial, but this approximation

can introduce some instability, and hence appropriate tuning is needed to ensure good performance.

At the end of Sect. 3.3, we discuss how can the incremental 4D-Var strategy be combined with the flow-dependent background covariances proposed in this paper.

3.3 4D-Var filtering with flow-dependent covariance

In this section, we describe a 4D-Var-based filtering procedure that can be implemented in an online fashion, with observations $\{y_l^\circ\}_l$ obtained at times $t_l = lh, l = 0, 1, \dots$ (although the method can be also easily adapted to the case when the time between the observations varies). We first fix an assimilation window length $T = kh$, for some $k \in \mathbb{Z}_+$, giving rise to consecutive windows $[0, t_k], [t_k, t_{2k}], \dots$

Let the background distribution on $\mathbf{x}(t_0)$ be Gaussian with mean $\mathbf{x}^b(t_0)$ and covariance matrix \mathbf{B}_0 . In general, let the background distributions for the position of the signal at the beginning of each assimilation window, $\{\mathbf{x}(t_{mk})\}_{m \geq 0}$, have means $\{\mathbf{x}^b(t_{mk})\}_{m \geq 0}$ and covariance matrices $\{\mathbf{B}_{mk}\}_{m \geq 0}$. There are several ways to define these quantities sequentially, as we shall explain later on in this section. Assuming that these are determined with some approach, working on the window $[t_{mk}, t_{(m+1)k}]$ we set our estimator $\hat{\mathbf{x}}(t_{mk})$ of $\mathbf{x}(t_{mk})$ as the MAP of the posterior of $\mathbf{x}(t_{mk})$ given background with mean $\mathbf{x}^b(t_{mk})$ and covariance \mathbf{B}_{mk} , and data $\mathbf{y}_{mk:(m+1)k-1}^\circ$; we also obtain estimates for subsequent times in the window, via push-forward, i.e.

$$\hat{\mathbf{x}}(t_l) := \mathbf{M}(t_l, t_{mk})[\hat{\mathbf{x}}(t_{mk})], \quad mk \leq l < (m + 1)k.$$

Recall that the numerical value of MAP is obtained by the Gauss-Newton method (see (3.1), with the details given in Section 3.1).

We now discuss choices for the specification of the background distributions. A first option is to set these distributions identical to the first one, and set $\mathbf{B}_{mk} := \mathbf{B}_0$ and $\mathbf{x}(t_{mk}) := \mathbf{x}(t_0)$ (i.e. no connection with earlier observations). A second choice (used in the first practical implementations of the 4D-Var method) is to set $\mathbf{B}_{mk} := \mathbf{B}_0$ (the covariance is kept constant) but change the background mean to

$$\mathbf{x}^b(t_{mk}) := \mathbf{M}(t_{mk}, t_{(m-1)k})[\hat{\mathbf{x}}(t_{(m-1)k})], \tag{3.8}$$

i.e. adjusting the prior mean to earlier observations. Finally, one can attempt to update both the mean and the covariance matrix of the background (prior) distribution, and this is the approach we follow here.

Note that we still define the background means according to (3.8). To obtain data-informed background covariances \mathbf{B}_{mk} , we use Gaussian approximations for a number, say $b \geq 1$, of earlier windows of length T , and appropriately push-forward these to reach the instance of current interest

t_{mk} . There are two reasons why we use a fixed b and do not push-forward all the way from time t_0 . The first is to avoid quadratic costs in time. The total computational cost for our approach up to time mT scales linearly with time for a fixed b , but if we would start from t_0 , then we would incur $O(m^2)$ computational cost (or if it is done by storing the whole covariance matrix directly, then the approach would have $O(d^2)$ memory cost which is prohibitive in practice). The second reason is that a Gaussian distribution propagated through nonlinear dynamics for longer and longer intervals of length bT becomes highly non-Gaussian for large values of b , so the resulting background distribution can lead to poorer results than using smaller values of b . Reminiscent to 4D-Var, at time $t_{(m-b)k}$ we always start off the procedure with the same background covariance \mathbf{B}_0 . In Paulin et al. (2018b) it was shown—under certain assumptions—that for a class of nonlinear dynamical systems, for a fixed observation window T , if $\|\mathbf{R}_i\| = O(\sigma^2)$ and $\sigma\sqrt{h}$ is sufficiently small (h is the observation time step) then the smoothing and filtering distributions can indeed be well approximated by Gaussian laws. Following the ideas behind (3.5), an approximation of the Hessian of J , evaluated at the MAP given data $y_{(m-1)k:m k-1}^\circ$, is given as

$$\mathbf{B}_{(m-1)k}^{-1} + \mathbf{D}_{(m-1)k:m k-1},$$

where we have defined

$$\begin{aligned} \mathbf{D}_{(m-1)k:m k-1} &:= \sum_{l=0}^{k-1} \mathbf{A}_{m-1,l}^T \mathbf{R}_{(m-1)k+l}^{-1} \mathbf{A}_{m-1,l}; \\ \mathbf{A}_{m-1,l} &:= \\ \mathbf{H}_{(m-1)k+l} &\mathbf{M}(t_{(m-1)k+l}, t_{(m-1)k})[\hat{\mathbf{x}}(t_{(m-1)k})]. \end{aligned}$$

If the precision (inverse covariance) of the background were 0, then $\mathbf{D}_{(m-1)k:m k-1}$ would correspond to the Hessian of J at the MAP, and the smoothing distribution could be approximated by a Gaussian with mean $\hat{\mathbf{x}}(t_{(m-1)k})$ and precision matrix $\mathbf{D}_{(m-1)k:m k-1}$. Recall the change of variables formula: if $\mathbf{Z} \sim N(m, \mathbf{P}^{-1})$ in \mathbb{R}^n and $\varphi: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a continuously differentiable function, then $\varphi(\mathbf{Z})$ follows approximately

$$N\left(\varphi(m), \left[\left(\left(\frac{\partial \varphi}{\partial m}\right)^{-1}\right)^T \cdot \mathbf{P} \cdot \left(\frac{\partial \varphi}{\partial m}\right)^{-1}\right]^{-1}\right). \quad (3.9)$$

The quality of this approximation depends on the size of the variance of \mathbf{Z} , and the degree of nonlinearity of φ . A way to consider the effect of the observations in the previous b assimilation windows is therefore by using the recursion

$$\begin{aligned} \mathbf{B}_{(m-b)k}^m &= \mathbf{B}_0; \\ \mathbf{B}_{(m-b+j)k}^m & \end{aligned}$$

$$\begin{aligned} &= \left[\left(\left(\mathbf{M}(t_{(m-b+j)k}, t_{(m-b+j-1)k})[\hat{\mathbf{x}}(t_{(m-b+j-1)k})]^T \right)^{-1} \right. \right. \\ &\quad \cdot \left. \left(\left(\mathbf{B}_{(m-b+j-1)k}^m \right)^{-1} + \mathbf{D}_{(m-b+j-1)k:(m-b+j)k-1} \right) \right. \\ &\quad \left. \left. \cdot \left(\mathbf{M}(t_{(m-b+j)k}, t_{(m-b+j-1)k})[\hat{\mathbf{x}}(t_{(m-b+j-1)k})]^{-1} \right) \right]^{-1}, \end{aligned} \quad (3.10)$$

for $j = 1, \dots, b$, and set $\mathbf{B}_{mk} := \mathbf{B}_{mk}^m$, where we have defined

$$\begin{aligned} \mathbf{D}_{(m-b+j-1)k:(m-b+j)k-1} &:= \\ \sum_{l=0}^{k-1} \mathbf{A}_{m-b+j-1,l}^T \mathbf{R}_{(m-b+j)k+l}^{-1} \mathbf{A}_{m-b+j-1,l}, &\quad j = 1, \dots, b; \\ \mathbf{A}_{m-b+j-1,l} &:= \mathbf{H}_{(m-b+j-1)k+l} \\ \cdot \mathbf{M}(t_{(m-b+j-1)k+l}, t_{(m-b+j-1)k})[\hat{\mathbf{x}}(t_{(m-b+j-1)k})]. \end{aligned}$$

Note that similarly to the idea of variance inflation for the Kalman filter, one could include a multiplication by an inflation factor $(1 + \alpha)$ for some $\alpha > 0$ in the definition of $\mathbf{B}_{(m-b)k}^m$ in (3.10). To simplify the expressions (3.10), we define

$$\begin{aligned} \mathbf{M}_{-j} &:= \mathbf{M}(t_{(m-j+1)k}, t_{(m-j)k})[\hat{\mathbf{x}}(t_{(m-j)k})], \\ j &= 1, 2, \dots, b. \end{aligned} \quad (3.11)$$

The action of \mathbf{B}_{mk}^{-1} on a vector $\mathbf{w} \in \mathbb{R}^n$ can be computed efficiently as follows. Let

$$\mathbf{w}_{-j} := \mathbf{M}_{-j}^{-1} \cdots \mathbf{M}_{-1}^{-1} \mathbf{w}, \quad j = 1, 2, \dots, b. \quad (3.12)$$

We then determine the recursion

$$\begin{aligned} \mathcal{B}_{-b} &:= (\mathbf{M}_{-b}^T)^{-1} (\mathbf{B}_0^{-1} + \mathbf{D}_{(m-b)k:(m-b+1)k-1}) \mathbf{w}_{-b} \\ \mathcal{B}_{-j} &:= (\mathbf{M}_{-j}^T)^{-1} (\mathcal{B}_{-j+1} + \mathbf{D}_{(m-j)k:(m-j+1)k-1} \mathbf{w}_{-j}), \\ j &= b-1, \dots, 1. \end{aligned} \quad (3.13)$$

Then it is easy to see that $\mathbf{B}_{mk}^{-1} \mathbf{w} = \mathcal{B}_{-1}$.

In order to evaluate the quantities in (3.12) and (3.13) for the shallow-water equations (2.7)-(2.9), we need implement the effect of the Jacobians $\mathbf{M}_1, \dots, \mathbf{M}_k$, their inverses $\mathbf{M}_1^{-1}, \dots, \mathbf{M}_k^{-1}$, and their transpose for the previous b assimilation windows. Note that multiplying by $\mathbf{D}_{(m-l)k:(m-l+1)k-1}$ is equivalent to evaluating (3.7) for the appropriate Jacobians; hence, it is also based on multiplication by these Jacobians, their inverses and their transposes.

Matrix-vector products of the form $\mathbf{M}_j \mathbf{v}$ and $\mathbf{M}'_j \mathbf{v}$ can be computed by the tangent linear model, and by the adjoint equations, respectively. When computing matrix-vector products of the form $\mathbf{M}_j^{-1} \mathbf{v}$ and $(\mathbf{M}_j^{-1})' \mathbf{v}$, we need

to run the tangent linear model backward in time, while the adjoint equations forward in time. It is important to note that while normally this would lead to numerical instability if done for a long time period (as the shallow-water equations are dissipative), this is not a problem here as we only run them over short time periods, the time between two observations (even shorter time periods could be possible if needed by breaking the Jacobians into products of Jacobians over shorter intervals). The initial point of these backward runs of the original equation (and forward runs of the adjoint equation) is always based on a forward run of the original equation; hence, the instability is avoided. For the shallow-water equations, the Jacobians M_j can be stored directly in a sparse format; see Appendix for more details (this reduces the need to use the ODE solvers repeatedly during the optimisation steps; however, this is not necessary for the method to work as we can always use the adjoint/tangent linear solvers directly as described above).

Our method is based on the forward and adjoint equations of the model. The computational cost of using these b previous intervals in each iteration of our proposed flow-dependent 4D-Var methodology (requiring the calculation of the gradient of the cost function J , and the and the product of its Hessian with a vector) is at most $O(b)$ times more than just using the observations in the current window. The key idea behind the choice of the precision matrices (3.10) is that we approximate the likelihood terms corresponding to the observations in the previous windows by Gaussian distributions, and then propagate them forward to the current time position via the Jacobians of the dynamics according to the change of variables formula (3.9). This allows us to effectively extend the assimilation time T to $(b + 1)T$, but without the non-convexity issue that would occur if it would be extended directly (this was confirmed in our simulations). Moreover, the choice (3.10) introduces a strong linkage between the successive assimilation windows, and effectively allows the smoothing distribution to rely on two sided information (both from the past and the future), versus one-sided information if one would simply use a longer window of length $(b + 1)T$.

In fact, this was confirmed during our simulations, and we have found that increasing T beyond a certain range did not improve the performance, while increasing b has resulted in an improvement in general up to a certain point.

We note that the incremental 4D-Var strategy of Courtier et al. (1994) can be implemented here as follows. In the inner loops, we compute the gradient $\frac{\partial J}{\partial \mathbf{x}(t_0)}$ using the adjoint equations with lower resolution models (see (3.2)). The Hessian-matrix products required to compute the $\mathbf{B}_0^{-1}[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)]$ term in the gradient from the flow-dependent matrix covariances can also be computed using lower resolution models. For the Hessian-matrix products in the iterative Krylov subspace solvers, we can always use the lower resolution model.

In contrast with this, in the outer loops, when computing the gradient $\frac{\partial J}{\partial \mathbf{x}(t_0)}$ we always need to use the highest resolution model (including in Hessian-matrix products required to compute the $\mathbf{B}_0^{-1}[\mathbf{x}(t_0) - \mathbf{x}^b(t_0)]$ term in the gradient). The Hessian-matrix products for the iterative solvers can still be computed on lower resolution models.

4 Simulations

In this section, we are going to illustrate the performance of our proposed method through simulation results. As a comparison, we also apply the Hybrid 4D-Var method on the same datasets as these form the basis of most currently used data assimilation systems (see Clayton et al. 2013, Lorenc et al. 2015). Section 5 of Kalnay (2003), (Evensen 2009) and Sections 7-8 of Reich and Cotter (2015) offer excellent introductions to standard data assimilation methods such as 4D-Var and ENKF and its variants.

We consider two linear observation scenarios. In both of them, it is assumed that the observations happen in every h time units and that the linear observation operators H_i are the same each time, represented by a matrix $\mathbf{H} \in \mathbb{R}^{n^o \times n}$. The scenarios are as follows.

1. We observe the height h at every gridpoint $1 \leq i, j \leq d$, and the velocities u and v at selected locations with spatial frequency r in both directions for a positive integer r . All of the observation errors are i.i.d. $N(0, \sigma^2)$ random variables.
2. We observe the height h at selected locations with spatial frequency r in both directions for a positive integer r . All of the observation errors are i.i.d. $N(0, \sigma^2)$ random variables.

The motivation of using these scenarios is that the heights are in general easier to observe than the velocities (for example, by satellite altimetry). In the following experiments, we are going to compare the performance of our proposed 4D-Var method using flow-dependent background covariances with the Hybrid 4D-Var method. In Sect. 4.1, we use synthetic data, while in Sect. 4.2 we use data from the tsunami waves after the 2011 Japan earthquake. Note that we did not have access to observations over multiple locations at multiple time points, only an estimate of the initial position of the ocean surface after the earthquake. We ran our shallow-water model initiated from this estimate and then generated observations from the model to be fed into the data assimilation systems.

Data assimilation methods are often evaluated in the literature over long time periods using real data, and their performance is reaching some sort of stationary. While using longer time periods is natural, the shallow-water equations

are unstable and so are their discretised version. The solution blows up after a while, and we are not able to use these equations directly for periods longer than 10 days. A standard practice in the literature is to modify the dynamics using a filter that smooths out the high-frequency components and removes the instability from the system (see Laible and Lillys 1997). We have tried to add a similar filter for our discretisation scheme, and this managed to stabilise the process, so it could run for a longer period. Nevertheless, this somewhat arbitrary modification means that all 4D-Var-based approaches that rely on the likelihood using the original discretisation of the shallow-water equations (without the filtering part) seemed to be significantly less accurate when applied to the data generated by the filtered equations. We believe that this is due to the fact that the model is misspecified in this case, so we cannot expect the same level of performance as before. None of the methods for generating flow-dependent covariances seemed to have outperformed simple fixed diagonal background covariances in these experiments. The arbitrary filtering step dramatically impacts the results. Due to this, in the absence of real data, we believe that using the original dynamics over shorter periods (where it can be evaluated without resorting to filters) is a more appropriate way to evaluate the performance of data assimilation methods on the shallow-water equations. We have used this approach in the paper.

Our flow-dependent 4D-Var method is fully deterministic; hence, the simulation results would be exactly the same if we ran them again on the same dataset. The Hybrid 4D-Var method is using at least 100 particles in our implementation, which is sufficient to ensure good stability, and we did not detect significant variability in the results over multiple runs.

4.1 Comparison based on synthetic data

First, we compare the performance of various methods using synthetic data. The shallow-water equations were solved on the torus $[0, L]^2$ with $L = 210\text{km}$. The initial condition $U(0) := (u(0), v(0), h(0))$, ocean depth H and other ODE parameters were chosen as follows:

$$\begin{aligned} u(0) &= 0.5 + 0.5 \sin\left(\frac{2\pi(x+y)}{L}\right), \\ v(0) &= 0.5 - 0.5 \cos\left(\frac{2\pi(x-y)}{L}\right), \\ h(0) &= 2 \sin\left(\frac{2\pi x}{L}\right) \cos\left(\frac{2\pi y}{L}\right); \\ H &= 100 + 100(1 + 0.5 \sin\left(\frac{2\pi x}{L}\right))(1 + 0.5 \sin\left(\frac{2\pi y}{L}\right)); \\ v &= 10^{-3}, \quad c_b = 10^{-5}, \quad g = 9.81, \quad f = 10^{-4}. \end{aligned}$$

The discretised versions of the initial condition and the ocean depth were obtained under the choices $d = 21$, $\Delta = 10\text{km}$. In the first observation scenario, we have chosen

the spatial frequency of the velocity observations as $r = 3$ (giving 49 velocity observations). All of the heights are also observed. Observations are made every 10 seconds, the total observation time is 1 day, and the observation errors had standard deviation $\sigma = 10^{-2}$.

In the second observation scenario, we have chosen the spatial frequency of the height observations as $r = 3$ (giving 49 height observations). Velocities are not observed. Observations are made every 60 seconds, the total observation time is 10 days, and the observation errors had standard deviation $\sigma = 10^{-2}$.

For the Hybrid 4D-Var, we have used localisation, as described in Section 8.3 of Reich and Cotter (2015). This localisation matrix was chosen as $(C)_{kl} = \rho\left(\frac{r_{k,l}}{r_{\text{loc}}}\right)$, where $r_{k,l}$ denotes the spacial distance on the torus between two gridpoints k and l , and ρ is the filter function describing the decay of correlations, and r_{loc} is the localisation radius. The filter function in the localisation was chosen according to equation (8.29) of Reich and Cotter (2015) as

$$\rho(s) = \begin{cases} 1 - \frac{5}{3}s^2 + \frac{5}{8}s^3 + \frac{1}{2}s^4 - \frac{1}{4}s^5 & \text{for } 0 \leq s \leq 1 \\ -\frac{2}{3}s^{-1} + 4 - 5s + \frac{5}{3}s^2 + \frac{5}{8}s^3 - \frac{1}{2}s^4 + \frac{1}{12}s^5 & \text{for } 1 \leq s \leq 2 \\ 0 & \\ \text{otherwise.} & \end{cases} \tag{4.1}$$

We have also used multiplicative ensemble inflation, as described in Section 8.2 of Reich and Cotter (2015). This consists of rescaling the ensemble members around their mean by a factor $1 + c_{\text{inf}}$ for some $c_{\text{inf}} > 0$. Finally, the climatological covariance B_0 and mean were estimated from the true unobserved path of the system during the total assimilation time (1 day in our first experiment, and 10 days in our second). This is typically estimated from past data, or by running the model over longer time periods (see Fairbairn et al. 2014), but this was not possible as the nonlinear shallow-water equations suffer from numerical instabilities over long time periods (due to the breaking waves phenomenon). The initial ensemble was sampled from a Gaussian distribution corresponding to the estimated climatological mean, and climatological covariance. We have used hybridisation, so the ensemble-based flow-dependent covariances were combined with the climatological covariances according to a hybridisation parameter $c_{\text{hyb}} \in [0, 1]$.

Table 1 states the values of the localisation radius, multiplicative inflation, and hybridisation parameters that we tested in a grid search for two experiments. In total, $3 \times 3 \times 6 = 54$ different values were tested, with $c_{\text{hyb}} = 0$ corresponding to 4D-EnVar.

Table 1 Tested parameter values for Hybrid 4D-Var

Parameter & Experiment	Synthetic 1 day	Synthetic 10 days	Tsunami
r_{loc} (localisation)	2,3,4	2,3,4	2,3,4
c_{inf} (inflation)	$2.5 \cdot 10^{-4}$, $5 \cdot 10^{-4}$, 10^{-3}	$5 \cdot 10^{-4}$, 10^{-3} , $2 \cdot 10^{-3}$	0.01, 0.02, 0.03
c_{hyb} (hybridisation)	0, 0.1, 0.3, 0.5, 0.7, 0.9	0, 0.1, 0.3, 0.5, 0.7, 0.9	0, 0.1, 0.3, 0.5, 0.7, 0.9
Ensemble size	200	200	100

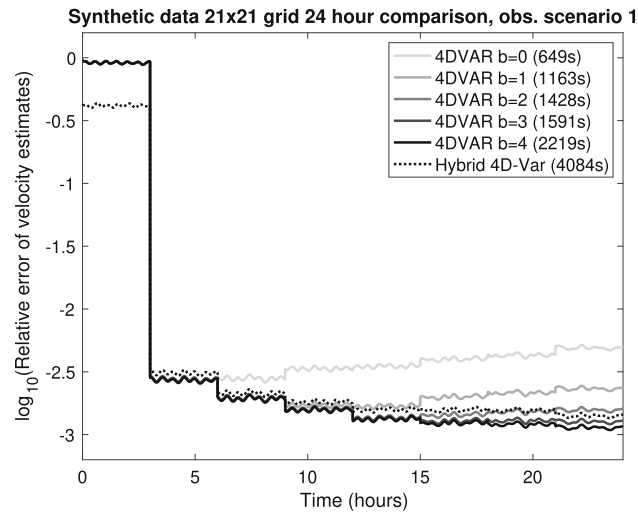


Fig. 1 Relative errors of velocity estimates in the case of synthetic data for all methods. Setting: $d = 21$, $k = 1080$, $T = 3h$, $\sigma = 10^{-2}$, $\Delta = 10^4$, 10 seconds between observations. The 4D-Var and Hybrid 4D-Var methods process the data in batches of size k ; hence, the filtering accuracy is poor until we have reached the end of the first observation window. The best parameter values for Hybrid 4D-Var were $r_{loc} = 4$, $c_{hyb} = 0$, $c_{inf} = 2.5 \cdot 10^{-4}$ (see Table 1 for all of the parameter values that we tested).

In the case of the proposed 4D-Var method with flow-dependent background covariances, the initial covariance \mathbf{B}_0 was chosen as a diagonal matrix. The assimilation window T was chosen as 3 hours (which offered the best performance for fixed background covariance). Figure 1 illustrates the performance of the various methods in the first observation scenario for the 1 day run case.

The 4D-Var method was optimised based on the Gauss-Newton method with preconditioned conjugate gradient (PCG)-based linear solver. We did not use any preconditioner, and the maximum number of iterations per PCG step was set to 100 (which was sufficient for reducing the relative residual below 0.01 in most cases). In the Hybrid 4D-Var method, we used a hybrid version of the ENKF based on fixed covariances (i.e. a linear combination of them), and the optimisation was done in a similar way as for the 4D-Var method. All of the methods were implemented in Matlab and ran on a single node of the Oxford ARC Arcus-B HPC cluster (16 cores per node). The measure of performance is

the relative error of the unobserved component at a certain time t , i.e. if $\mathbf{w}(t) \in \mathbb{R}^{n-n^o}$ denotes the true value of the unobserved component, and $\hat{\mathbf{w}}(t) \in \mathbb{R}^{n-n^o}$ is the estimator, then $\|\hat{\mathbf{w}}(t) - \mathbf{w}(t)\| / \|\mathbf{w}(t)\|$ is the relative error ($\|\cdot\|$ refers to the Euclidean norm).

We have also repeated the experiment in the more challenging second observation scenario. Figure 2a shows the performance of 4D-Var with a fixed background covariance matrix with varying window sizes $T = 3h, 6h, 9h, 12h$ and $18h$. In the case of $12h$ and $18h$, we have used the idea of Pires et al. (1996) to first find the optimum for shorter windows and then gradually extend the window length to T to avoid issues with non-convexity. This has resulted in a better optimum at the cost of longer computational time (it did not make a difference at shorter window lengths). Overall, we can see that the $T = 12h$ has the best performance, but the computational time is longer than for $T = 9h$ (as we in fact first find the optimum based on the first half of the observations in the window, and then continue with the other half). At $T = 18h$, the performance diminishes due to the non-convexity of the likelihood, and even the gradual extension of the window length fails to overcome this problem.

Figure 2b compares the performance of our method (based on $T = 9h$, but with choices of b from 1 to 5) with 4D-Var with fixed window length ($T = 12h$), ENKF and Hybrid 4D-Var. For this synthetic dataset, our 4D-Var-based method with $b = 3$ offered the best performance. $b = 4$ was similar but with higher computational cost, and $b = 5$ resulted in worse performance, likely due to the nonlinearity of the system. We can see that using observations in earlier assimilation windows to update the background covariance matrix in a flow-dependent way is very beneficial, with relative errors reduced by as much as 70-90% compared to using a fixed background matrix.

To better understand the reason for this improvement in performance, in Figure 3 we have plotted the average correlations in background covariances between the components at a given distance, in the cases $b = 1, 2, 3$, for the first observation scenario computed at the last observation window (after 24 hours). As we can see, as b increases, the background covariance matrix changes and becomes less-and-less localised.

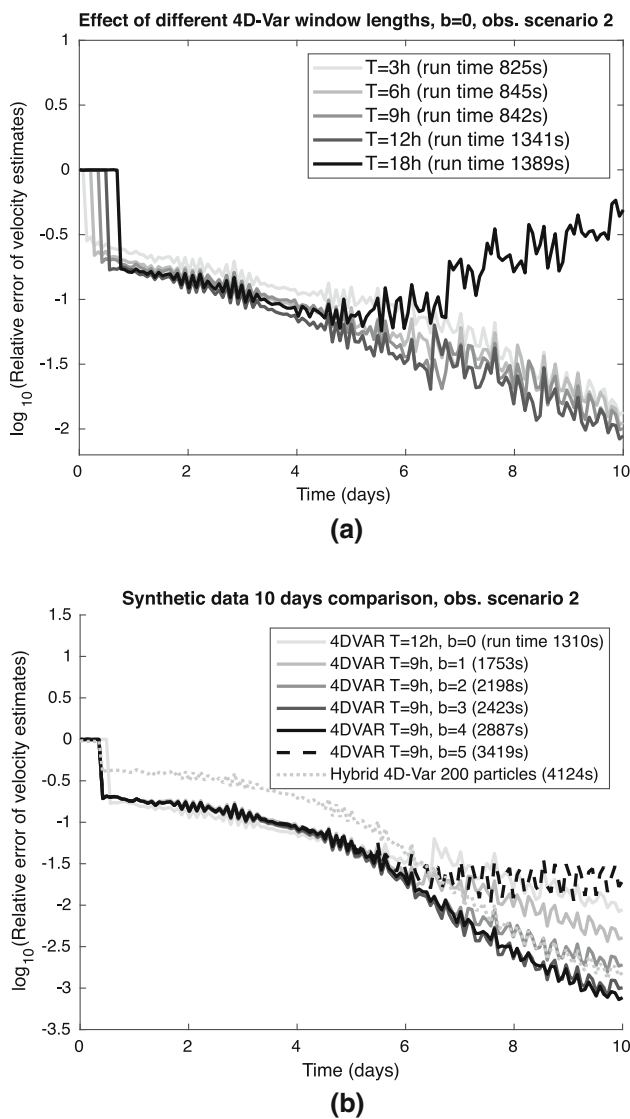


Fig. 2 Relative errors of velocity estimates for synthetic data with observation scenario 3. Setting: $d = 21$, $\sigma = 10^{-2}$, $\Delta = 10^4$, 60 seconds between observations. Figure 2a shows the performance of 4D-Var with fixed covariance matrix for different time lengths. Figure 2b compares the performance of our method with Hybrid 4D-Var. The best parameter values for Hybrid 4D-Var were $r_{loc} = 2$, $c_{hyb} = 0$, $c_{inf} = 5 \cdot 10^{-4}$ (see Table 1 for all of the parameter values that we tested).

The performance of the Hybrid 4D-Var was quite good, and it considerably improved upon using a fixed background covariance matrix, but nevertheless our method still had significantly better accuracy, especially in second observation scenario which involved data assimilation over a longer time period (10 days) with less frequent observations. We believe that this increase in accuracy is due to the more accurate modelling of background covariances, which become less-and-less localised over longer time periods.

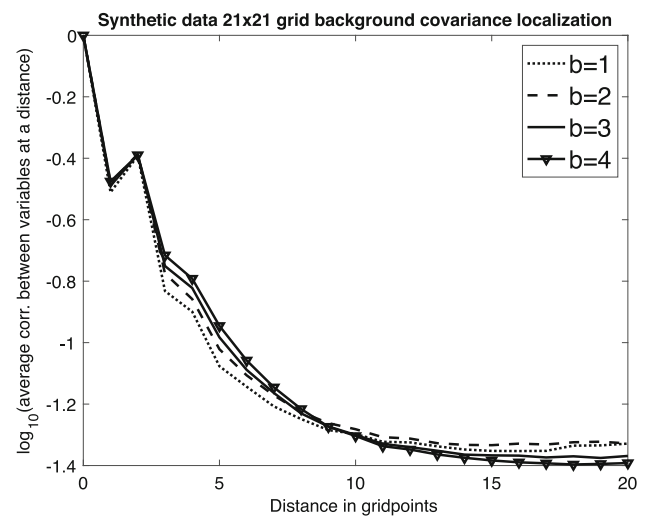


Fig. 3 Correlation decay in flow-dependent background covariances for different values of b . Setting: $d = 21$, $k = 540$, $T = 9h$, $\sigma = 10^{-2}$, $\Delta = 10^4$, observation scenario 2.

4.2 Comparison based on tsunami data

The shallow-water equations are applied in tsunami modelling. (Saito et al. 2011) estimate the initial distribution of the tsunami waves after the 2011 Japan earthquake. They use data from 17 locations in the ocean, where the wave heights were observed continuously in time. We have used these estimates as our initial condition for the heights and set the initial velocities to zero (as they are unknown). Using publicly available bathymetry data for \bar{h} , and the above described initial condition, we have run a simulation of 40 minutes for our model, see Fig. 4. We have tested the efficiency of the data assimilation methods also on this simulated dataset, considering a time interval from 10 to 40 minutes (thus the initial condition corresponds to the value of the model after 10 minutes and is shown in Fig. 4b). Due to the somewhat rough nature of the tsunami waves, in this example we have found that setting the background precision (inverse covariance) matrix B_0^{-1} as zero offered the best performance for the proposed 4D-Var method, while we used a diagonal matrix for the Hybrid 4D-Var method. The localisation and ensemble inflation was implemented as described in Sect. 4.1, with the tested parameter values shown in Table 1. The 4D-Var method was optimised based on the Gauss-Newton method with preconditioned conjugate gradient (PCG)-based linear solver without any preconditioner, and the maximum number of iterations per PCG step was set to 500.

Figure 5 compares the performance of the methods for this synthetic dataset implemented for grid size $d = 336$ (so the dimension on the dynamical system is $n = 3d^2 = 338, 688$) in the first observation scenario, where the spatial frequency of the velocity observations was chosen as $r = 48$ (i.e. $7 \cdot 7 = 49$ velocity observations in total).

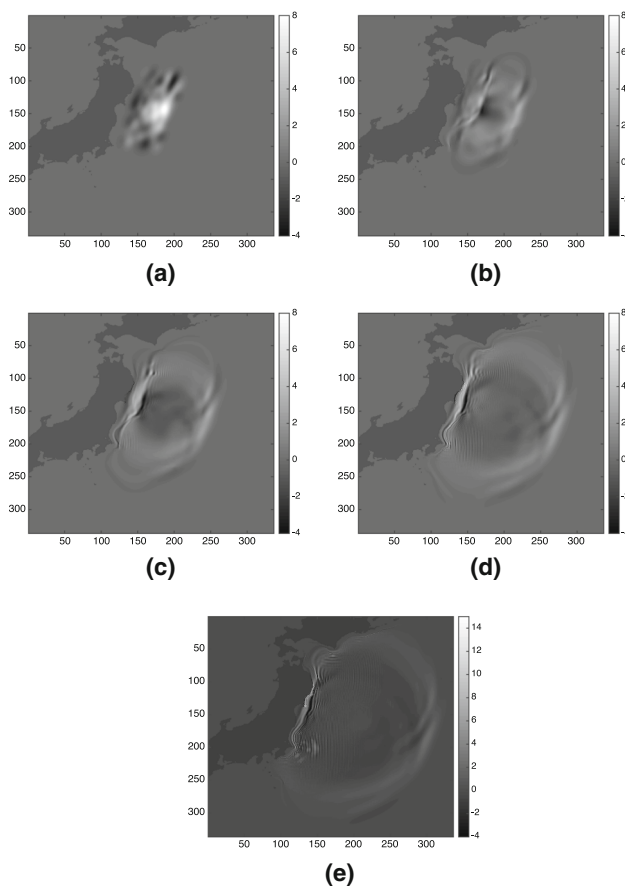


Fig. 4 Evolution of the height of the tsunami waves (in meters) at 0, 10, 20, 30, and 40 mins (for grid size $d = 336$).

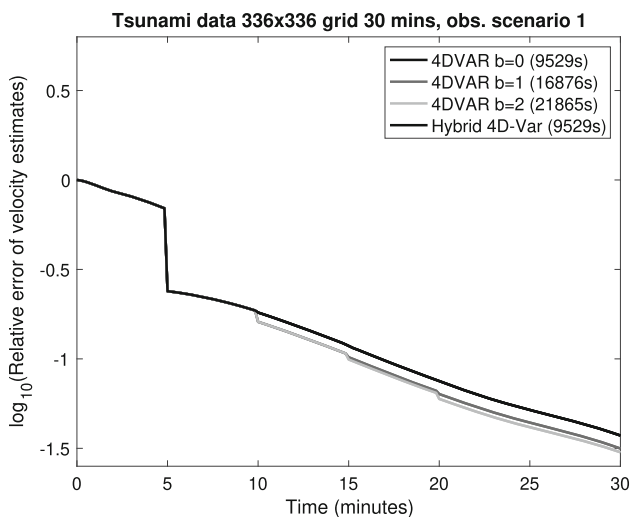


Fig. 5 Relative error of estimates of velocities for tsunami data in the first observation scenario, all methods. Setting: $d = 336$, $k = 30$, $T = 30$ mins, $\sigma = 10^{-2}$. The best performance for hybrid Hybrid 4D-Var was obtained by only using the fixed background covariance matrix; hence, the performance and running time is equivalent to 4D-Var with $b = 0$ (see Table 1 for all of the parameter values that we tested).

As in the previous synthetic example, the proposed 4D-Var-based method offers the best performance. Note that the best performance of the Hybrid 4D-Var was achieved when we have used only a fixed background covariance matrix, and coefficient for the ENKF-based background covariance is set to zero in the hybridisation (see Table 1 for the tested parameter values for hybridisation, inflation and localisation). Hence in this complex highly unstable situation the ENKF-based background covariances did not help, while our proposed flow-dependent covariances improved the precision of the velocity estimates when using $b = 1$ and $b = 2$.

5 Conclusion

In this work, we have presented a new method for updating the background covariances in 4D-Var filtering and applied it to the shallow-water equations. Our method finds the MAP estimator of the initial position using the Gauss-Newton's method with the Hessian matrix stored and the background covariances obtained in a factorised form. Our method is computationally efficient and has memory and computational costs that scale nearly linearly with the size of the grid.

4D-Var-based methods are less directly parallelisable compared to ENKF as the optimisation steps and the ODE solver steps are indeed inherently serial. Hence in a parallel environment, it is likely that the background covariance part of the Hybrid 4D-Var can be computed significantly faster using the ENKF-based methods compared to the proposed method. However, we have shown in the experiments that the method proposed by this paper can be significantly more accurate in the perfect model scenario for the shallow-water equations. Moreover, both Hybrid 4D-Var and the proposed method use the same computations based on adjoint equations and tangent linear model for the data in the current assimilation window (they only differ in formulation of the flow-dependent background covariances). The total computational time of the background covariances takes typically at most factor of b times longer than the computations for the data in the current window, and thus the proposed method is not overly computationally expensive.

It remains to be seen if the improvements in accuracy for the proposed method also hold in more complex weather forecasting models, in the presence of some model error. Our hope is that the proposed method could yield significant improvements in accuracy in some challenging data assimilation scenarios where modelling covariance localisation is difficult, without the need of extensive tuning.

The Matlab code for the experiments is available at <https://github.com/paulindani/shallowwater>.

Acknowledgements The authors thank Joe Wallwork for providing us the tsunami data set and for our correspondence related to the shallow-

water equations. All authors were supported by an AcRF tier 2 grant: R-155-000-161-112. AJ was also supported by KAUST baseline funding. This material is based upon work supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office, and by the U.K. Ministry of Defence (MoD) and the U.K. Engineering and Physical Research Council (EPSRC) under grant number EP/R013616/1. DC was partially supported by the EPSRC grant: EP/N023781/1. AB was supported by a Leverhulme Trust Prize.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

A sparse storage of Jacobians for shallow-water dynamics

In this section, we explain a possible method for the computation and storage of the Jacobians M_i , $1 \leq i \leq k$ specifically for the case of the shallow-water Eqs. (2.7-2.9). For other equations, it might be the case that storing $(M_i)_{1 \leq i \leq k}$ directly as follows is not practical because the interaction between the components is not local and the Jacobian matrix is not sparse. In such cases, we can still apply the tangent-linear and adjoint equations for computing the matrix-vector products $M_i v$ and $M_i^T v$, as explained in Sect. 3.3.

One can observe that time derivatives at a grid position only depends on its grid neighbours. Moreover, the shallow-water equations are of the general form $\frac{dx}{dt} = -Ax - B(x, x) + f$, where A is an $n \times n$ matrix, B is a $n \times n \times n$ array, and f is a constant vector in \mathbb{R}^n (note that for the shallow-water Eqs. (2.7-2.9), we have $f = 0$). For equations of this form, there is an efficient way of calculating the time derivatives and their Jacobians, stated in equations (3.14) and (3.16) of Paulin et al. (2018b). Based on these, one can use Taylor’s expansion to compute the Jacobian $M(t, s)[x(s)]$, that is

$$M(t, s)[x(s)] \approx I_n + \sum_{l=1}^{l_{\max}} \frac{\partial \left(\frac{d^l}{dt^l} M(t, s)[x(s)] \Big|_{t=s} \right)}{\partial x(s)} \cdot \frac{(t-s)^l}{l!}, \tag{A.1}$$

for some $l_{\max} > 0$. Due to the fact that the first derivatives only contain terms from neighbouring gridpoints, it is easy to see that the above approximation only has nonzero elements for gridpoints that are no more than l_{\max} steps away. This means that as long as $t - s$ is sufficiently small, the Jacobian

$M(t, s)[x(s)]$ can be stored as a sparse matrix with $\mathcal{O}(n)$ nonzero elements. If the time interval between the observations is sufficiently small, then each of M_1, \dots, M_k can be stored as a single sparse matrix defined by (A.1). The inverse of the Jacobian satisfies that $(M(t, s)[x(s)])^{-1} = M(s, t)[x(t)]$, so it can be calculated by (A.1) with terms $(s - t)^l$ instead of $(t - s)^l$ and $x(t)$ instead of $x(s)$.

At this point, we note that one could attempt to use the Jacobians $M(t_l, t_0)$ directly. However, for $l \ll n$, storing the Jacobians M_1, \dots, M_l separately requires $\mathcal{O}(nl)$ memory, and the effect of $M_l M_{l-1} \dots M_1$ on a vector can be evaluated in $\mathcal{O}(nl)$ time, while for 2D lattices, the product $M_l \dots M_1$ would require $\mathcal{O}(nl^2)$ memory, and its effect on a vector would require $\mathcal{O}(nl^2)$ time to evaluate (for 3D lattices, it would incur up to $\mathcal{O}(nl^3)$ memory and computational cost). For the same reason, for longer time intervals between observations, it is more effective to break the interval into $r > 1$ smaller blocks of equal size, and store the Jacobians corresponding to each of them. In this case, when applying the Jacobian M_l on a vector, the result can computed as the product of the Jacobians for the shorter intervals.

References

Asch M, Bocquet M, Nodet M.: Data assimilation: methods, algorithms, and applications. SIAM (2016)

Auligné, T., Ménétrier, B., Lorenc, A.C., Buehner, M.: Ensemble-variational integrated localized data assimilation. *Mon. Weather Rev.* **144**(10), 3677–3696 (2016)

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

Bannister, R.N.: A review of forecast error covariance statistics in atmospheric variational data assimilation i: characteristics and measurements of forecast error covariances. *Q. J. Royal Meteorol. Soc.* **134**(637), 1951–1970 (2008)

Bannister, R.N.: A review of forecast error covariance statistics in atmospheric variational data assimilation ii: modelling the forecast error covariance statistics. *Q. J. Royal Meteorol. Soc.* **134**(637), 1971–1996 (2008)

Bates, P.D., Horritt, M.S., Fewtrell, T.J.: A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *J. Hydrol.* **387**(1), 33–45 (2010)

Bengtsson, L., Ghil, M., Källén, E.: *Dynamic meteorology: data assimilation methods*, vol. 36. Springer (1981)

Beskos, A., Crisan, D., Jasra, A.: On the stability of sequential monte carlo methods in high dimensions. *Ann Appl Probab* **24**, 1396–1445 (2014)

Blayo, E., Bocquet, M., Cosme, E., Cugliandolo, L.F.: Advanced data assimilation for geosciences. *International Summer School-Advanced Data Assimilation for Geosciences* (2014)

Bocquet, M., Pires, C.A., Wu, L.: Beyond Gaussian statistical modeling in geophysical data assimilation. *Mon. Weather Rev.* **138**(8), 2997–3023 (2010)

Bonavita, M., Hólm, E., Isaksen, L., Fisher, M.: The evolution of the ECMWF hybrid data assimilation system. *Q. J. R. Meteorol. Soc.* **142**(694), 287–303 (2016)

Bousserez, N., Henze, D., Perkins, A., Bowman, K., Lee, M., Liu, J., Deng, F., Jones, D.: Improved analysis-error covariance matrix for

- high-dimensional variational inversions: application to source estimation using a 3d atmospheric transport model. *Q. J. R. Meteorol. Soc.* **141**(690), 1906–1921 (2015)
- Buehner, M.: Ensemble-derived stationary and flow-dependent background-error covariances: Evaluation in a quasi-operational NWP setting. *Q. J. R. Meteorol. Soc.* **131**(607), 1013–1043 (2005)
- Clayton, A., Lorenc, A.C., Barker, D.M.: Operational implementation of a hybrid ensemble/4D-Var global data assimilation system at the Met Office. *Q. J. R. Meteorol. Soc.* **139**(675), 1445–1461 (2013)
- Courtier, P., Talagrand, O.: Variational assimilation of meteorological observations with the direct and adjoint shallow-water equations. *Tellus A* **42**(5), 531–549 (1990)
- Courtier, P., Thépaut, J.N., Hollingsworth, A.: A strategy for operational implementation of 4D-Var, using an incremental approach. *Q. J. R. Meteorol. Soc.* **120**(519), 1367–1387 (1994)
- Crisan, D. (ed.): *Mathematics Of Planet Earth: A Primer*. Advanced Textbooks In Mathematics, World Scientific Europe (2017)
- Dashti, M., Stuart, A.: The bayesian approach to inverse problems. *Handbook of Uncertainty Quantification* pp 311–428 (2016)
- Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. *J. R. Stat. Soc. Ser. B Stat Methodol.* **68**(3), 411–436 (2006)
- Derber, J., Bouttier, F.: A reformulation of the background error covariance in the ecmwf global data assimilation system. *Tellus A* **51**(2), 195–221 (1999)
- Egbert, G.D., Bennett, A.F., Foreman, M.G.: TOPEX/POSEIDON tides estimated using a global inverse model. *J. Geophys. Res: Oceans* **99**(C12), 24821–24852 (1994)
- Evensen, G.: *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media (2009)
- Fairbairn, D., Pring, S.R., Lorenc, A.C., Roulstone, I.: A comparison of 4dvar with ensemble data assimilation methods. *Q. J. R. Meteorol. Soc.* **140**(678), 281–294 (2014)
- Fisher M (2003) Background error covariance modelling. In: *Seminar on Recent Development in Data Assimilation for Atmosphere and Ocean*, pp 45–63
- Fisher, M., Leutbecher, M., Kelly, G.: On the equivalence between Kalman smoothing and weak-constraint four-dimensional variational data assimilation. *Q. J. R. Meteorol. Soc.* **131**(613), 3235–3246 (2005)
- Gejadze, I.Y., Copeland, G., Le Dimet, F.X., Shutyayev, V.: Computation of the analysis error covariance in variational data assimilation problems with nonlinear dynamics. *J. Comput. Phys.* **230**(22), 7923–7943 (2011)
- Ghil, M., Malanotte-Rizzoli, P.: Data assimilation in meteorology and oceanography. *Adv. Geophys.* **33**, 141–266 (1991)
- Ghil, M., Cohn, S., Tavantzis, J., Bube, K., Isaacson, E.: Applications of estimation theory to numerical weather prediction. In: *Dynamic meteorology: Data assimilation methods*, Springer, pp 139–224 (1981)
- Gratton, S., Lawless, A.S., Nichols, N.K.: Approximate Gauss-Newton methods for nonlinear least squares problems. *SIAM J. Optim.* **18**(1), 106–132 (2007)
- Hamill, T.M., Whitaker, J.S., Kleist, D.T., Fiorino, M., Benjamin, S.G.: Predictions of 2010's tropical cyclones using the gfs and ensemble-based data assimilation methods. *Mon. Weather Rev.* **139**(10), 3243–3247 (2011)
- Ide, K., Courtier, P., Ghil, M., Lorenc, A.C.: Unified notation for data assimilation: operational, sequential and variational (special issue, data assimilation in meteorology and oceanography: Theory and practice). *J. Meteorol. Soc. Japan Ser II* **75**(1B), 181–189 (1997)
- Kalnay, E.: *Atmospheric Modeling*. Cambridge University Press, Cambridge, *Data Assimilation and Predictability* (2003)
- Kalnay, E., Li, H., Miyoshi, T., Yang, S.C., Ballabrera-Poy, J.: 4-d-var or ensemble kalman filter? *Tellus A: Dyn. Meteorol. Oceanography* **59**(5), 758–773 (2007)
- Kuhl, D.D., Rosmond, T.E., Bishop, C.H., McLay, J., Baker, N.L.: Comparison of hybrid ensemble/4dvar and 4dvar within the navdas-ar data assimilation framework. *Mon. Weather Rev.* **141**(8), 2740–2758 (2013)
- Lahoz, W., Khattatov, B., Menard, R.: *Data assimilation: making sense of observations*. Springer Science & Business Media (2010)
- Laible, J.P., Lillys, T.P.: A filtered solution of the primitive shallow-water equations. *Adv. Water Resour.* **20**(1), 23–35 (1997)
- Law, K., Stuart, A., Zygalakis, K.: *Data assimilation, Texts in Applied Mathematics*, vol 62. Springer, Cham, a mathematical introduction (2015)
- Lawless, A., Gratton, S., Nichols, N.: An investigation of incremental 4D-Var using non-tangent linear models. *Q. J. R. Meteorol. Soc.* **131**(606), 459–476 (2005)
- Le Dimet, F.X., Talagrand, O.: Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A: Dyn. Meteorol. Oceanography* **38**(2), 97–110 (1986)
- Le Dimet, F.X., Navon, I.M., Daescu, D.N.: Second-order information in data assimilation. *Mon. Weather Rev.* **130**(3), 629–648 (2002)
- van Leeuwen, P.J.: Particle filtering in geophysical systems. *Mon. Weather Rev.* **137**(12), 4089–4114 (2009)
- van Leeuwen, P.J.: Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Q. J. R. Meteorol. Soc.* **136**(653), 1991–1999 (2010)
- Lorenc, A.: Four-dimensional variational data assimilation. *Advanced Data Assimilation for Geosciences: Lecture Notes of the Les Houches School of Physics: Special Issue, June 2012* p 31 (2014)
- Lorenc, A.C.: Modelling of error covariances by 4D-Var data assimilation. *Q. J. R. Meteorol. Soc.* **129**(595), 3167–3182 (2003)
- Lorenc, A.C., Bowler, N.E., Clayton, A.M., Pring, S.R., Fairbairn, D.: Comparison of hybrid-4DENVar and hybrid-4DVar data assimilation methods for global NWP. *Mon. Weather Rev.* **143**(1), 212–229 (2015)
- Lyard, F., Lefevre, F., Letellier, T., Francis, O.: Modelling the global ocean tides: modern insights from FES2004. *Ocean Dyn.* **56**(5–6), 394–415 (2006)
- Miller, R.N., Ghil, M., Gauthiez, F.: Advanced data assimilation in strongly nonlinear dynamical systems. *J. Atmos. Sci.* **51**(8), 1037–1056 (1994)
- Murray, F.J., Miller, K.S.: *Existence theorems for ordinary differential equations*. Courier Corporation (2013)
- Navon, I.M.: Data assimilation for numerical weather prediction: a review. In: *Data assimilation for atmospheric, oceanic and hydrologic applications*, Springer, pp 21–65 (2009)
- Park, S.K., Xu, L. (eds) *Data assimilation for atmospheric, oceanic and hydrologic applications*, vol 1. Springer (2009)
- Park, S.K., Xu, L. (eds) *Data assimilation for atmospheric, oceanic and hydrologic applications*, vol 2. Springer Science & Business Media (2013)
- Park, S.K., Xu, L. (eds) *Data assimilation for atmospheric, oceanic and hydrologic applications*, vol 3. Springer Science & Business Media (2017)
- Parrish, D.F., Derber, J.C.: The national meteorological center's spectral statistical-interpolation analysis system. *Mon. Weather Rev.* **120**(8), 1747–1763 (1992)
- Paulin, D., Jasra, A., Crisan, D., Beskos, A.: On concentration properties of partially observed chaotic systems. *Adv. Appl. Probab.* **50**(2), 440–479 (2018)
- Paulin, D., Jasra, A., Crisan, D., Beskos, A.: Optimization based methods for partially observed chaotic systems. *Foundations of Computational Mathematics* (2018). <https://doi.org/10.1007/s10208-018-9388-x>
- Pelinovsky, E., *Hydrodynamics of tsunami waves*. In: *Waves in Geophysical Fluids*, Springer, pp 1–48 (2006)

- Pires, C., Vautard, R., Talagrand, O.: On extending the limits of variational assimilation in nonlinear chaotic systems. *Tellus A* **48**(1), 96–121 (1996)
- Rebeschini, P., Van Handel, R., et al.: Can local particle filters beat the curse of dimensionality? *Ann. Appl. Probab.* **25**(5), 2809–2866 (2015)
- Reich, S., Cotter, C.: *Probabilistic forecasting and Bayesian data assimilation*. Cambridge University Press, New York (2015)
- Saito, T., Ito, Y., Inazu, D., Hino, R.: Tsunami source of the 2011 Tohoku-Oki earthquake, Japan: Inversion analysis based on dispersive tsunami simulations. *Geophysical Research Letters* **38**(7) (2011)
- Salmon, R.: *Introduction to Ocean Waves*. Scripps Institution of Oceanography, University of California, San Diego, available at <http://pordlabs.ucsd.edu/rsalmon/111.textbook.pdf> (2015)
- Talagrand, O.: Assimilation of observations, an introduction. *J. Meteorol. Soc. Jpn* **75**(1B), 191–209 (1997)
- Talagrand, O., Courtier, P.: Variational assimilation of meteorological observations with the adjoint vorticity equation i: Theory. *Q. J. Royal Meteorol. Soc.* **113**(478), 1311–1328 (1987)
- Trémolet, Y.: Accounting for an imperfect model in 4D-Var. *Q. J. R. Meteorol. Soc.* **132**(621), 2483–2504 (2006)
- Wang, X., Parrish, D., Kleist, D., Whitaker, J.: Gsi 3dvar-based ensemble-variational hybrid data assimilation for ncep global forecast system: Single-resolution experiments. *Mon. Weather Rev.* **141**(11), 4098–4117 (2013)
- Zupanski, M.: Maximum likelihood ensemble filter: Theoretical aspects. *Mon. Weather Rev.* **133**(6), 1710–1726 (2005)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.