Tellus (2003), 55A, 1–15
Printed in UK. All rights reserved

TELLUS
ISSN 0280–6495

Determination of optimal nudging coefficients

By P. A. VIDARD[1*], F.-X. LE DIMET[2] and A. PIACENTINI[1], [1]*Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique, 42, avenue Coriolis, 31057 Toulouse, France;* [2]*Laboratoire de Modélisation et Calcul., Université Joseph Fourier, BP 53 38041 Grenoble, France*

(Manuscript received 28 May 2001; in final form 31 July 2002)

ABSTRACT

A four-dimensional variational assimilation (4D-Var) scheme is now widely used by meteorological centres in a operational way. However, most of these applications do not take account of model error. Indeed, the classical 4D-Var with imperfect model formulation is unaffordable for current computational means. This paper presents a low-cost method for dealing with model errors in 4D variational assimilation. This method can be formally compared to a Kalman filter. This new scheme is tested on two configurations: first on a Burger equation, which allows one to calibrate the method, and second on a more relevant shallow-water equations model, both in a twin experiment framework. It is shown that, compared to classical 4D-Var results, this method provides a noticeable improvement.

1. Introduction

Data assimilation is a wide class of numerical methods for estimating the state of a system by combining information from observational data with information provided by a numerical model. One of the most important applications of these methods is the best estimation of the state of the atmosphere (or the ocean) at a given time in order to improve the accuracy of the numerical forecast. In recent years, developments of both observational means (remote sensing, buoys, tomography, etc.) and computing resources have permitted a wide improvement in data assimilation methods and their efficiency.

The current operational method in meteorological centres is called 4D-Var (four-dimensional variational data assimilation). It consists of assimilating all the observational and model information from the previous time sequence. The problem is formulated as an optimal control problem in which the criteria measure the misfit between the model predictions and the observations of the system state. One of the main assumptions of 4D-Var is that the model describes exactly the system behaviour. However, in practice the model equations do not represent the exact evolution of the system, and model errors arise because of the lack of resolution, inaccurate representation of small-scale physics or errors in boundary conditions, topography or forcing terms.

The next step in 4D-Var development will be to consider that the model is not exact, i.e., for example, to introduce the model errors correcting term in the control vector (see Jazwinski, 1970; Derber, 1989; Cohn, 1997). The principle of the complete method is to add in the control vector, a residual error correcting term which is added to the model equation at every time step. Due to the size of this new problem [the dimension of the state variable (typically 10^6–10^7) times the number of time steps], this approach is unaffordable for current computational resources. In order to reduce the cost of model error control, one can propose to control only several and well chosen directions of the error (Vidard et al., 2000), or only the time-correlated part (bias) (Nichols and Griffith, 2000; Derber, 1989).

In this paper we will introduce a new technique for treating model errors, based on an idea of Zou

*Corresponding author. Present address: European Centre for Medium-range Weather Forecasting, Shinfield Park, Reading, Berkshire RG2 9AX, UK.
e-mail: nes@ecmwf.int

et al. (1992). This method is an extension of nudging (or Newtonian relaxation), which was the first operational data assimilation method used in oceanography (Lyne et al., 1982; Krishnamurti et al., 1991; Lorenc et al., 1991). It consists of relaxing the model state towards the observations during the assimilation period by adding a non-physical 'nudging' term to the model equation. The nudging terms are defined as the difference between the observation and the model solution weighted by a nudging coefficient.

The nudging can be thought as an approximation of the Kalman filter (KF): the best nudging coefficients are those related to a KF in a linear case (see Lorenc, 1986; Lorenc et al., 1991). However, the KF is very costly in practice (storage and manipulation of $N \times N$ matrices where N is the size of the state vector) compared to the very cheap nudging. Moreover, in order to be applied, a KF needs simplifications that do not take into account the whole time period.

In this paper we will use optimal control methods to estimate both the initial condition (as in 4D-Var) and the nudging coefficients in order to correct model errors in a much more efficient way (from the computational point of view).

First we will recall the equations of 4D-Var and introduce optimal nudging, and we will show that this new method and the Kalman filter are equivalent in a sense to be defined. Section 2 deals with the application of optimal nudging data assimilation on a very simple Burger's equation. Three forms of the nudging matrix will be tested to determine which compromise between the size of control and the quality of the results will be optimal.

Finally we will test this method on a more realistic shallow-water model and compare it with classical 4D-Var data assimilation on the same model. We will show that for a little extra cost this new method provides a quite noticeable improvement.

The notations used in this paper are the unified ones advocated by Ide et al. (1997).

2. Optimal nudging: motivation

The aim of the 4D-Var classical approach of data assimilation is to try to reduce the misfit between the observations and the forecast state by controlling the initial condition of the analysis period.

Given a discretised model M, let $\mathbf{x} \in \mathcal{C} \subset \mathbb{R}^N \cdot \mathbf{x}^{\mathrm{b}}$ is the background state, or first guess of the minimisation; the evolution of the state can be described as

$$\begin{cases} \mathbf{x}_0 = \mathbf{x}^{\mathrm{b}} + \delta\mathbf{x}_0 \\ \mathbf{x}_{i+1} = M_{(t_i,t_{i+1})}(\mathbf{x}_i). \end{cases} \tag{1}$$

The aim of the method is to search for $\delta\mathbf{x}_0$ that minimises the following cost function:

$$J(\delta\mathbf{x}_0) = \overbrace{\frac{1}{2}\sum_{i=0}^{n}\left\langle \mathbf{R}^{-1}\left(H_i(\mathbf{x}_i) - \mathbf{y}_i^{\mathrm{o}}\right), H_i(\mathbf{x}_i) - \mathbf{y}_i^{\mathrm{o}}\right\rangle}^{J_{\mathrm{o}}} + \overbrace{\frac{1}{2}\left\langle \mathbf{B}^{-1}\delta\mathbf{x}_0, \delta\mathbf{x}_0\right\rangle}^{J_{\mathrm{b}}}$$

where $\mathbf{R}$ is the $m \times m$ observational error covariance matrix and $\mathbf{B}$ the $n \times n$ background error covariance matrix, $\mathbf{y}_i^{\mathrm{o}} \in \mathbb{R}^m$ is the observation vector at time t_i and H_i is the observation operator that computes the model equivalent quantities and interpolates them to the observation location at time t_i. The first term of the cost function (called J_{o}) represents the misfit we wish to minimise in an idealised case with true observations and a model describing exactly the evolution of the system. However, due to observation errors and errors induced by the model, this term can not be cancelled. That is why one introduces the second term (called J_{b}) that prevents the solution from being non-physical (trying to to fit the observations exactly) and then generating oscillations to retrieve physical equilibrium. The appropriate equilibrium between the two terms is provided by the inverse of the error covariance matrices $\mathbf{B}^{-1}$ and $\mathbf{R}^{-1}$, which represent the confidence we have in the background and the observations, respectively.

Due to the term J_{b} and to the model errors, the innovation residual $\mathbf{d}_k = H_k(\mathbf{x}_k^{\mathrm{a}}) - \mathbf{y}_k^{\mathrm{o}}$ remains non-zero after the 4D-Var assimilation.

The error sources can be written as:

Prediction (or Forecast) Error:

$$\varepsilon_k^{\mathrm{f}} = \mathbf{x}_k^{\mathrm{f}} - \mathbf{x}_k^{\mathrm{t}}$$

Observation Error:

$$\varepsilon_k^{\mathrm{o}} = \mathbf{y}_k^{\mathrm{o}} - H_k\left(\mathbf{x}_k^{\mathrm{t}}\right)$$

where $\mathbf{x}_k^{\mathrm{f}}$ represents the **f**orecast state at time t_k and $\mathbf{x}_k^{\mathrm{t}}$ the **t**rue state.

Then we note that the innovation residual can be written as:

$$\begin{aligned} \mathbf{d}_k &= H_k\left(\mathbf{x}_k^{\mathrm{f}}\right) - \mathbf{y}_k^{\mathrm{o}} \\ &\approx \mathbf{H}_k\varepsilon_k^{\mathrm{f}} - \varepsilon_k^{\mathrm{o}} \end{aligned} \tag{2}$$

where

$$\mathbf{H}_k = \left.\frac{\partial H_k}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_k^{\mathrm{f}}}$$

is the tangent linear of the observation operator, and ε^{f} is assumed to be negligible with respect to $\mathbf{x}^{\mathrm{f}}$.

Equation (2) shows that important information about model and observation errors remains in this residual. In the following we will try to use it to improve the 4D-Var assimilation scheme.

Formally, in the case of a completely and directly observed state an intuitive approach was to modify the forecast state using a correcting term including the innovation residual. In this case, the inverse of the observation operator $\mathbf{H}^{-1}$ exists and then we could write:

$$\mathbf{x}_k^{\mathrm{a}} = \mathbf{x}_k^{\mathrm{f}} + \mathbf{H}_k^{-1}\left(\mathbf{d}_k - \varepsilon^{\mathrm{o}}\right).$$

Among existing data assimilation methods one can quote the Kalman filter (Kalman, 1960), which provides an analysed state using the innovation vector to correct the forecast by

$$\mathbf{x}_k^{\mathrm{a}} = \mathbf{x}_k^{\mathrm{f}} + \mathbf{K}_k\mathbf{d}_k$$

where $\mathbf{K}$ is called the gain matrix and is computed as follows:

$$\mathbf{K}_k = \mathbf{P}_k^{\mathrm{f}}\mathbf{H}_k\left(\mathbf{H}_k\mathbf{P}_k^{\mathrm{f}}\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}\right)^{-1}$$

with $\mathbf{P}^{\mathrm{f}}$ the forecast error covariance matrix.

It can be pointed out that in the Kalman filter, if observations are assumed exact ($\mathbf{R} = 0$), we retrieve the gain matrix as: $\mathbf{K}_n = \mathbf{H}_n^{-1}$ and $\varepsilon_n^{\mathrm{f}} = \mathbf{H}_n^{-1}\mathbf{d}_k$.

However, there are two major drawbacks in Kalman filtering: it requires the manipulation of very large matrices and it needs simplifications that do not take into account the whole time period.

That is why we try here to estimate an optimal 'Gain' matrix using variational methods (4D-Var). The following extends the method of optimal nudging introduced by Zou et al. (1992), exploring three kinds of nudging matrices and introducing the smoothed optimal nudging which avoids the shocks induced by nudging corrections in the model.

3. Optimal nudging: the method

In this section we introduce the main point of this paper: the Four-Dimensional Optimal Nudging Data Assimilation scheme (4D-ON). The computation of the gradient of the cost function does not require more tools than for classical 4D-Var, but only a few more operations and storage.

The aim of nudging methods is to relax the model states toward the observations by adding a 'nudging' term. This term is the misfit between observation and forecast $\mathbf{y}_{i+1}^{\mathrm{o}} - H_{i+1}(M_{(t_{i+1},t_i)}(\mathbf{x}_i))$ weighted by the nudging operator $\mathbf{G}_i$. In this part the complete nudging case is considered, i.e. $\mathbf{G}_i$ is a $n \times m$ matrix where n is the dimension of the state vector and m the dimension of the observation vector. The problem is now how to estimate $\mathbf{G}_i$.

Let M be a non-linear discretised model describing the ocean or atmosphere evolution. We focus on the problem:

$$\begin{cases} \mathbf{x}_0 = \mathbf{x}^{\mathrm{b}} + \delta\mathbf{x}_0 \\ \mathbf{x}_{i+1} = M_{(t_{i+1},t_i)}(\mathbf{x}_i) + \mathbf{G}_{i+1} \\ \qquad \times \left(\mathbf{y}_{i+1}^{\mathrm{o}} - H_{i+1}\left(M_{(t_{i+1},t_i)}(\mathbf{x}_i)\right)\right) \end{cases} \tag{3}$$

and we wish to minimise the cost function:

$$J(\delta\mathbf{x}_0, \mathbf{G}) = \frac{1}{2}\sum_{i=0}^{n}\left\langle \mathbf{R}^{-1}H_i(\mathbf{x}_i) - \mathbf{y}_i^{\mathrm{o}},\, H_i(\mathbf{x}_i) - \mathbf{y}_i^{\mathrm{o}}\right\rangle + \frac{1}{2}\left\langle \mathbf{B}^{-1}\delta\mathbf{x}_0,\, \delta\mathbf{x}_0\right\rangle + \overbrace{\frac{1}{2}\sum_{i=1}^{n}\left\langle \mathbf{Q}_i^{-1}\mathbf{G}_i\left(\mathbf{y}_i^{\mathrm{o}} - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right),\, \mathbf{G}_i\left(\mathbf{y}_i^{\mathrm{o}} - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right)\right\rangle}^{J_{\mathrm{nudg}}}. \tag{4}$$

Basically we will consider a 4D-Var cost function where $\mathbf{R}$ is the observation error correlation matrix and $\mathbf{B}$ is the background error covariance matrix, plus a regularisation term J_{nudg} that prevents $\mathbf{G} = \mathbf{G}_1, \ldots, \mathbf{G}_n$ from being too large, where $\mathbf{Q}$ is the forecast error covariance matrix. This formulation can be related to an adjoint parameter estimation where $\mathbf{G}$ are the model parameters to be estimated (see Navon, 1998).

Actually, the regularisation term J_{nudg} comes from the classical formulation of the cost function of the 4D-Var with imperfect model data assimilation (see Jazwinski, 1970; Tikhonov and Arsenin, 1977; Cohn,

1997; Alekseev and Navon, 2001). In this entire formulation of the control of model errors the cost function is

$$\begin{aligned} &J(\delta\mathbf{x}_0, \eta_1, \ldots, \eta_n) \\ &= \frac{1}{2}\sum_{i=0}^{n}\left\langle \mathbf{R}^{-1}H_i(\mathbf{x}_i) - \mathbf{y}_i^{\mathrm{o}}, H_i(\mathbf{x}_i) - \mathbf{y}_i^{\mathrm{o}}\right\rangle \\ &\quad + \frac{1}{2}\sum_{i=1}^{n}\left\langle \mathbf{Q}_i^{-1}\eta_i, \eta_i\right\rangle + \frac{1}{2}\left\langle \mathbf{B}^{-1}\delta\mathbf{x}_0, \delta\mathbf{x}_0\right\rangle \end{aligned} \tag{5}$$

where η_i is the correcting term at time t_i. So η_i has simply been replaced with the nudging correcting term.

In order to compute the gradient of the cost function we will use a classical Lagrangian method under the constraint of eq. (3). Let L, the Lagrangian, being defined by:

$$\begin{aligned} &L\left(\mathbf{G}, \delta\mathbf{x}_0; \mathbf{x}^*\right) \\ &= J(\mathbf{G}, \delta\mathbf{x}_0) + \sum_{i=1}^{n}\left\langle \mathbf{x}_i^*, \mathbf{x}_i - M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right. \\ &\quad \left. - \mathbf{G}_i\left(\mathbf{y}_i^{\mathrm{o}} - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right)\right\rangle \end{aligned}$$

$\mathbf{x}^*$ being the Lagrangian multiplier. Computing the partial derivatives

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{x}^*}\left(\mathbf{G}, \mathbf{v}; h_{\mathbf{x}}^*\right) &= \mathbf{x}_i - M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1}) \\ &\quad - \mathbf{G}_i^{\mathrm{T}}\left(\mathbf{y}_i^{\mathrm{o}} - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right) \end{aligned}$$

we can first retrieve

$$\frac{\partial L}{\partial \mathbf{x}^*} = 0 \Longrightarrow (3).$$

Moreover,

$$\begin{aligned} &\frac{\partial L}{\partial \delta\mathbf{x}_0}\left(\mathbf{G}, h_{\delta\mathbf{x}_0}, \mathbf{x}^*\right) \\ &= \sum_{i=0}^{n}\left\langle \mathbf{H}_i^{\mathrm{T}}\left(H_i(\mathbf{x}_i) - \mathbf{y}_i^{\mathrm{o}}\right), \hat{\mathbf{x}}_i\right\rangle + \left\langle \mathbf{B}^{-1}\delta\mathbf{x}_0, h_{\delta\mathbf{x}_0}\right\rangle \\ &\quad - \sum_{i=1}^{n}\left\langle \mathbf{M}_{(t_i,t_{i-1})}^{\mathrm{T}}\mathbf{H}_i^{\mathrm{T}}\mathbf{G}_i^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{G}_i\right. \\ &\quad \left.\times\left(\mathbf{y}_i^{\mathrm{o}} - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right), \hat{\mathbf{x}}_{i-1}\right\rangle \\ &\quad - \sum_{i=1}^{n}\left\langle \mathbf{x}_i^*, \hat{\mathbf{x}}_i\right\rangle + \left\langle \mathbf{M}_{(t_i,t_{i-1})}^{\mathrm{T}}\mathbf{x}_i^*, \hat{\mathbf{x}}_{i-1}\right\rangle \\ &\quad - \left\langle \mathbf{M}_{(t_i,t_{i-1})}^{\mathrm{T}}\mathbf{H}_i^{\mathrm{T}}\mathbf{G}_i^{\mathrm{T}}\mathbf{x}_i^*, \hat{\mathbf{x}}_{i-1}\right\rangle - \left\langle \mathbf{x}_0^*, h_{\delta\mathbf{x}_0}\right\rangle. \end{aligned}$$

So, if the adjoint state $\mathbf{x}^*$ is defined by:

$$\begin{cases} \mathbf{x}_n^* = \mathbf{H}_n^{\mathrm{T}}\mathbf{R}^{-1}\left[H_n(\mathbf{x}_n) - \mathbf{y}_n^{\mathrm{o}}\right] \\ \mathbf{x}_i^* = \left(\mathbf{M}_{(t_{i+1},t_i)} - \mathbf{G}_{i+1}\mathbf{H}_{i+1}\mathbf{M}_{(t_{i+1},t_i)}\right)^{\mathrm{T}}\mathbf{x}_{i+1}^* \\ \qquad + \mathbf{H}_i^{\mathrm{T}}\mathbf{R}^{-1}H_i(\mathbf{x}_i) - \mathbf{y}_i^{\mathrm{o}} - \mathbf{M}_{(t_{i+1},t_i)}^{\mathrm{T}}\mathbf{H}_{i+1}^{\mathrm{T}}\mathbf{G}_{i+1}^{\mathrm{T}} \\ \qquad \times \mathbf{Q}^{-1}\mathbf{G}_{i+1}\left(\mathbf{y}_{i+1}^{\mathrm{o}} - H_{i+1}\left(M_{(t_{i+1},t_i)}(\mathbf{x}_i)\right)\right) \end{cases} \tag{6}$$

we obtain

$$\begin{aligned} &\frac{\partial L}{\partial \delta\mathbf{x}_0}\left(\mathbf{G}, h_{\delta\mathbf{x}_0}, \mathbf{x}^*\right) = \left\langle \nabla_{\delta\mathbf{x}_0}J, h_{\delta\mathbf{x}_0}\right\rangle \\ &\text{i.e. } \nabla_{\delta\mathbf{x}_0}J = -\mathbf{x}^*(0) + \mathbf{B}^{-1}\delta\mathbf{x}_0; \end{aligned} \tag{7}$$

and finally:

$$\begin{aligned} &\frac{\partial L}{\partial \mathbf{G}_i}\left(\mathbf{v}, h_{\mathbf{G}_i}, \mathbf{x}_i^*\right) \\ &= \left\langle \mathbf{Q}^{-1}\mathbf{G}_i\left(\mathbf{y}_i^{\mathrm{o}} - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right),\right. \\ &\quad h_{\mathbf{G}_i}\left(\mathbf{y}_i^{\mathrm{o}} - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right)\Big\rangle \\ &\quad - \left\langle \mathbf{x}_i^*, h_{\mathbf{G}_i}\left(\mathbf{y}_i^{\mathrm{o}} - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right)\right\rangle \\ &= \langle \nabla_{\mathbf{G}}J, h_{\mathbf{G}}\rangle. \end{aligned} \tag{8}$$

Thus the gradient of the cost function with respect to $\mathbf{G}_i$ can be written:

$$\begin{aligned} \nabla_{\mathbf{G}_i}J &= \mathbf{Q}^{-1}\mathbf{G}_i\left(\mathbf{y}_i^{\mathrm{o}} - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right) \\ &\quad \times\left(\mathbf{y}_i^{\mathrm{o}} - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right)^{\mathrm{T}} \\ &\quad - \mathbf{x}^*(t)\left(\mathbf{y}^{\mathrm{o}}(t_i) - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x})_{i-1}\right)\right)^{\mathrm{T}}. \end{aligned} \tag{9}$$

Provided the initial condition contribution to the gradient [eq. (3)] is computed (thanks to one integration of the direct model and of the adjoint model) the computation of the nudging contribution of the gradient does not need extra 'heavy' computations. Indeed $\nabla J_{\mathbf{G}_i}$ is obtained by the product of vectors used in $\nabla J\delta\mathbf{x}_0$ (i.e. $\mathbf{G}_i(\mathbf{y}_i^{\mathrm{o}} - H_i(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})))$, $(\mathbf{y}_i^{\mathrm{o}} - H_i(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})))^{\mathrm{T}}$ and $\mathbf{x}^*$).

It can be proved that this algorithm is formally equivalent to the Kalman filter (see Appendix A).

4. Basic experiment on a Burger equation (1D)

In this section we will apply the optimal nudging data assimilation scheme on a simple one-dimensional Burger equation. It can be an easy validation of the ON scheme and will permit us to find an appropriate choice for the form of $\mathbf{G}_i$.

Let the model be

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} - \nu\dfrac{\partial^2 u}{\partial x^2} = f \\ u(0,t) = u(1,t) = 0 \\ u(x,0) = \sin \pi x \end{cases} \tag{10}$$

with $x \in]0, 1[$, $t > 0$.

Note that a known analytic exact solution is associated with a fixed forcing term (f). For this experiment we impose f to be $f(x,t) = \mathrm{e}^{-t}[\pi u \cos \pi x + (-1 + \nu\pi^2)\sin \pi x]$, and then $u = \mathrm{e}^{-t}\sin \pi x$ is an exact solution of eq. (10).

The 4D-ON is defined by adding the nudging term into the model equations:

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} - \nu\dfrac{\partial^2 u}{\partial x^2} = f + g[u_{\text{obs}} - h(u)] \\ u(0,t) = u(1,t) = 0 \\ u(x,0) = \sin \pi x. \end{cases} \tag{11}$$

We will focus on a sequence [0,1] split into $N + 1$ time steps. The same holds for spatial discretisation of [0,1] in E points. In this experiment $N = 5000$ and $E = 20$.

During this time sequence, we have M observations located at $(x_m, t_m) m = 1 \ldots M$.

Therefore we can rewrite our model as:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = f + \sum_{m=1}^{M} g_m\left(u_m^{\text{obs}} - u_m\right)\delta(x - x_m)\delta(t - t_m).$$

In a discrete formulation:

$$\frac{d\mathbf{u}}{dt}(t) + M(\mathbf{u})(t) = F(t) + \sum_{k=1}^{K} \mathbf{G}_k\left[\mathbf{u}_k^{\text{obs}} - H(\mathbf{u}_k)\right]\delta(t - t_k)$$

where H is the observation operator. In this experiment the minimisation is performed by a home-made conjugate gradient method using the Hessian of J as a preconditioner. The Hessian–vector product is computed using the second order adjoint of the model (see Wang et al., 1992; Le Dimet et al., 1998).

We will apply two types of model error simultaneously on the forcing term f: a 5% white noise and a 10% correlated noise (bias). The observations are obtained from the model free of error; only one point over five in space and one time step over 50 will be retained as observation, and we add a 5% white noise to these observations. Moreover, we add 20% noise to the initial condition in order to simulate background errors, and we set all the control vectors (increment of initial conditions and nudging coefficients) to 0.

In this experiment the number of time steps N is set to 5000 and the number of grid points E is set to 20; therefore the number of observed gridpoints at a given time step, M, is equal to 5.

Now we will analyse the results of three types of nudging coefficient matrices:

1. $\mathbf{G}_k$ is a full rank $E \times M$ matrix, i.e. at each point the correcting term is a linear combination of all forecast/observation misfits at the same time. This is the "dream" case, because each component of the state vector will be corrected and not only observed ones. Even if only a very small size problem allows one this form of $\mathbf{G}_k$, this experiment allows are to compare the efficiency of this ideal case with simpler forms of the nudging matrix and to emphasise the fact that the most is not necessarily the best.
2. $\mathbf{G}_k$ is a "pseudo-diagonal" matrix. In this case the model equations are only corrected at observation locations and the correction coefficient is different for each observation.
3. $\mathbf{G}_k$ is a scalar coefficient (just varying with time). It is the same case as previously, but for a fixed time t_k the correction coefficient is the same at every observation location.

Using one of these three forms of $\mathbf{G}_k$ increases the size of the control vector (in relation to the classical 4D-Var the control size of which is E) by $E \times M \times N$, $M \times N$ and N, respectively.

In Table 1 we compare the level of error RMS and the computational cost for these three forms of 4D-ON. The first column shows the "raw" results of the assimilation and the second one is obtained using a linear time interpolation of correcting terms as temporal smoothing and, for $\mathbf{G}$ diagonal matrix and scalar coefficients, a convolution product with a gaussian function λ as spatial smoothing [eq. (12)]:

$$\mathbf{u}(t) = \mathbf{u}(t) + \mathbf{G}_k\left(\mathbf{u}_k^{\text{obs}} - H(\mathbf{u}(t))\right) \quad \text{if } t = t_k$$

$$\mathbf{u}(t) = \mathbf{u}(t) + \left(\frac{t - t_k}{t_{k+1} - t_k}\mathbf{G}_{k+1} + \frac{t_{k+1} - t}{t_{k+1} - t_k}\lambda \star (\mathbf{G}_k) \times \left(\frac{t - t_k}{t_{k+1} - t_k}\mathbf{u}_{k+1}^{\text{obs}} + \frac{t_{k+1} - t}{t_{k+1} - t_k}\mathbf{u}_k^{\text{obs}} - H(\mathbf{u}(t))\right) \right.$$
$$\left. \text{if } t_k < t < t_{k+1}\right). \tag{12}$$

Table 1. *Norm of the error and CPU time for both raw and smoothed methods according to the form of* $\boldsymbol{G}$

		Raw		Interpolated	
Form of **G**	Size of control	RMS	CPU	RMS	CPU
G is a full matrix	$E \times M \times N$	45.06	57.80s	38.49	57.69s
G is a diagonal matrix	$M \times N$	65.30	6.90s	17.57	7.43s
G is a scalar coefficient	N	64.23	9.74s	17.33	3.63s
4D-Var	E	73.73	5.82s		
Without assimilation		248.21			

The level of RMS of error is computed by comparing the results of the assimilation to the known analytic solution.

In Table 1 the CPU time is not directly related to the size of the control due to the different number of iterations needed for convergence.

As a fair conclusion of this first experiment it can be noticed that even though all the 4D-ON schemes provide better results than classical 4D-Var, the complete 4D-ON ($\mathbf{G}_k$ full matrix) does not really provide better results than other alternatives. Even if in the raw method case the full matrix is slightly more efficient compared to simpler forms of $\mathbf{G}_k$, the difference does not justify the huge computational over-cost induced. (The full matrix control is more than 60 times more expensive than the scalar control.) Moreover, in the results shown for interpolated methods the diagonal matrix and the scalar coefficient provide better results. This can be explained by the excess degrees of freedom that the complete method contains. The simpler the control vector, the better conditioned the optimisation problem, which leads to a much more efficient minimisation. Moreover, we can notice that both the diagonal matrix and the scalar coefficient give almost the same results (different but indistinguishable on the graphs), which could indicate that the misfit vector contains spatial information, i.e. the misfit vector is itself well equilibrated on the whole grid. (The optimal nudging coefficient is nearly the same for all the observed points at a given time.)

Figure 1 shows the shape of one of the $\mathbf{G}_k$ as a full matrix. In this figure the inner contours represent higher values than outer ones. All the $\mathbf{G}_k$ being of the same shape, with slight amplitude variations, it can be noticed that the method gives more weight to the observations located in the centre of the domain.

5. Numerical experiment with a non-linear shallow water model

5.1. The model

For this test experiment we use a non-linear one-layer shallow-water model on a square basin with a flat bottom. (This model is classically used; see for example Adcroft and Marshall, 1998.)

Even if this model is not a very realistic one, the non-linearities included in the equation and the size of the state vector (more than 10^4) make its complexity sufficient to be a relevant test case.

Actually, the framework of this experiment is a twin-experiment one, i.e. the observations do not come from reality but from the model (with a slight difference from the model used in assimilation). In this way two different models are used. The first is used to represent

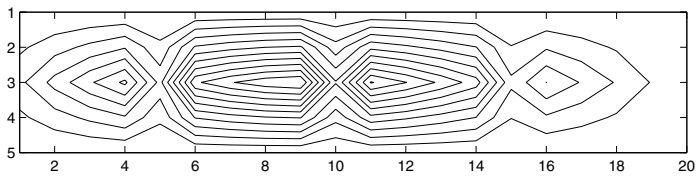

Fig. 1. Transpose of the estimated **G** as a full matrix.

the reality; it provides a simulated 'true' state evolution, and then using the observation operator (and possibly white noise) synthetic observations can be obtained. The second one represents the model (called a forecast model in the following in contrast to the true model or the truth evolution). In this way the presence of model errors is simulated and the evolution of the (pseudo-) reality is known.

Both evolutions of $\mathbf{x}_{\text{true/forecast}} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{h} \end{bmatrix}$ are described by:

$$\begin{cases} \partial_t \mathbf{u} - (f+\zeta)\mathbf{v} + \partial_x B = \dfrac{\tau}{\rho_0 \mathbf{h}} - r\mathbf{u} + \nu\Delta\mathbf{u} \\ \partial_t \mathbf{v} - (f+\zeta)\mathbf{u} + \partial_y B = \dfrac{\tau}{\rho_0 \mathbf{h}} - r\mathbf{v} + \nu\Delta\mathbf{v} \\ \partial_t \mathbf{h} + \partial_x(\mathbf{hu}) + \partial_y(\mathbf{hv}) = 0 \end{cases} \quad (13)$$

where ($\mathbf{u}$,$\mathbf{v}$) represents the current velocity, $\mathbf{h}$ is the height of the layer, $\zeta = \partial_x \mathbf{v} - \partial_y \mathbf{u}$ is the relative vorticity, $B = g^*\mathbf{h} + \frac{1}{2}(\mathbf{u}^2 + \mathbf{v}^2)$ is the Bernoulli potential, g^* is the reduced gravity, r is the linear friction coefficient and ν is the viscosity coefficient.

The forcing terms of the models are:

wind: $\tau_{\text{forecast}} = \tau_0 \dfrac{\sin 2\pi[y - (L/4)]}{L} L\boldsymbol{i}$, and

$$\tau_{\text{true}} = \tau_{\text{forecast}} \times \left[1 + 0.8 \times \sin\left(\frac{2\pi t}{\Delta t \times 480}\right)\right]$$

where L is the basin length.

Coriolis factor: $f = f_0 + \beta y$.

Boundary conditions are:

$\mathbf{v} = 0$ on north–south boundaries and $\mathbf{u} = 0$ on east–west boundaries

non-slippery boundary conditions.

In this experiment the numerical values are: $L = 2000\,\text{km}$, $f_0 = 0.7 \times 10^{-4}\,\text{s}^{-1}$, $\beta = 2 \times 10^{-11}\,\text{m}^{-1}\,\text{s}^{-1}$, $\nu = 15\,\text{m}^2\,\text{s}^{-1}$ (forecast), $\nu = 0.9 \times 15\,\text{m}^2\,\text{s}^{-1}$ (true), $r = 10^{-7}\,\text{s}^{-1}$ (forecast), $r = 0.9 \times 10^{-7}\,\text{s}^{-1}$ (true), $\rho_0 = 10^3\,\text{kg}\,\text{m}^{-3}$, $g^* = 0.02\,\text{m}\,\text{s}^{-2}$ and $\tau_0 = 0.015\,\text{N}\,\text{m}^{-2}$.

For the spatial discretisation a second-order centred scheme is used on an Arakawa C-grid with $\Delta x = \Delta y = 25$, km and for the time discretisation a leap-frog scheme with a time-step size $\Delta t = 30$ min and an Asselin time filter is added in the equations.

The initial condition of our data assimilation window is provided using a 6 yr spin-up of the forecast model, whereas true initial states and observations are computed with a true model spin-up.

5.2. *The assimilation*

The cost function is defined by eq. (4), but in order to improve the minimisation efficiency we operate a change of variable,

$$\mathbf{v} = \mathbf{B}^{-1/2}\delta\mathbf{x}_0 \quad (14)$$

where $\mathbf{B}^{-1/2}$ is defined as $A = A^{1/2}A^{T/2}$. This preconditioning has two advantages. (i) $\mathbf{B}^{-1/2}$ is the square root of the Hessian of J_b (and then a very good preconditioner for the minimisation) (Thepaut and Moll, 1990; Yang et al., 1996; Courtier, 1997; Derber and Bouttier, 1999). (ii) It implies the manipulation of $\mathbf{B}^{1/2}$ instead of its inverse [actually the inverse of eq. (14), $\delta\mathbf{x}_0 = \mathbf{B}^{1/2}\mathbf{v}$, is performed at the beginning of each iteration in order to retrieve a convenient initial condition].

Even in this simplified case (and consequently for more realistic ones) the entire $\mathbf{Q}$ and $\mathbf{B}$ matrices can neither be estimated nor stored explicitly because of their size ($\mathbf{B}$ is an $N \times N$ matrix where N is the size of the state vector). We are therefore forced to see $\mathbf{Q}$ and $\mathbf{B}$ as operators. Roughly (for more explanation see Weaver and Courtier, 2001) following Derber and Bouttier (1999) we can write, for instance:

$$\mathbf{B} = \mathbf{K}\mathbf{B}_\text{u}\mathbf{K}^\text{T}$$

where $\mathbf{K}$ is a balance operator that relates one state variable to the other (here $\mathbf{K}$ uses the geostrophic balance to compute the balanced part of $\mathbf{u}$ and $\mathbf{v}$ from $\mathbf{h}$) and $\mathbf{B}_\text{u}$ is the error covariance matrix for the unbalanced part of variables and assumed to be a block diagonal (i.e. the cross-covariances of the unbalanced part of variables are negligible):

$$\mathbf{B}_\text{u} = \Sigma_\mathbf{B}\mathbf{C}\Sigma_\mathbf{B}.$$

Here $\Sigma_\mathbf{B}$ is the diagonal matrix of background error standard deviation and represents the symmetric matrix of background error correlations for the unbalanced part of the state variables. The $\mathbf{C}$ operator is modelled using a diffusion equation (see Weaver and Courtier, 2001). Therefore the inverse of change of variable can be written as

$$\delta\mathbf{x}_0 = \mathbf{K}\Sigma_\mathbf{B}\mathbf{C}^{1/2}\mathbf{v}.$$

The same approach is used to build $\mathbf{Q}$, modifying standard error deviations ($\Sigma_\mathbf{Q}$) and possibly the parameters of the diffusion equation ($\mathbf{C}$).

In the same way, for this more realistic problem we can not really control the entire $\mathbf{G}$ matrices but

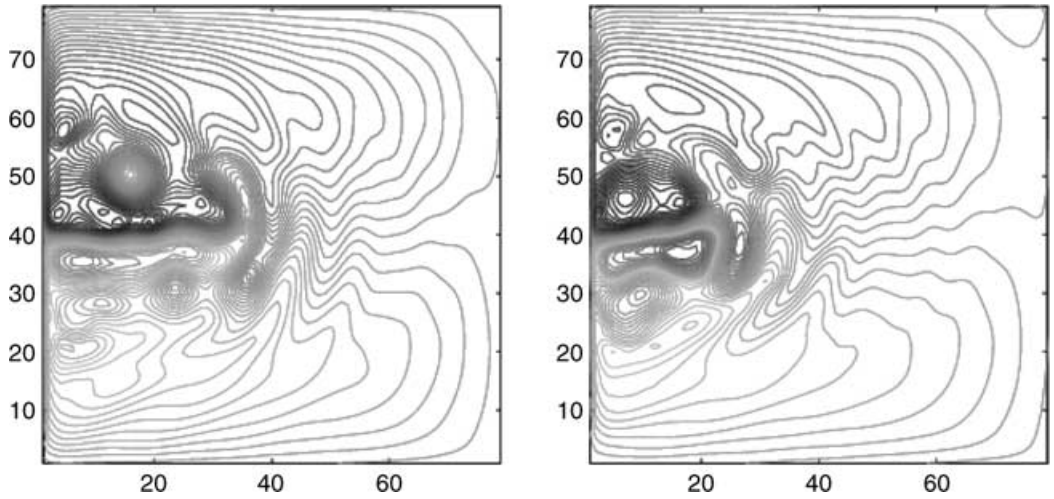

Fig. 2. SSH for forecast model (left) and true state (right) at initial time step.

only a few coefficients. However, this implies that the corrections are only applied on observed locations. In order to correct this drawback we can pre-multiply the correcting term by $\mathbf{Q}^{1/2}$; this will smooth the correction and simplify the computation of the cost function.

Now we can rewrite eq. (19):

$$\begin{cases} \mathbf{x}_0 = \mathbf{x}^{\mathrm{b}} + \mathbf{B}^{1/2}\mathbf{v} \\ \mathbf{x}_{i+1} = M_{(t_{i+1},t_i)}(\mathbf{x}_i) + \mathbf{Q}^{1/2}\mathbf{G}_{i+1} \\ \qquad \times \left(\mathbf{y}^{\mathrm{o}}_{i+1} - H_{i+1}\left(M_{(t_{i+1},t_i)}(\mathbf{x}_i)\right)\right) \end{cases} \tag{15}$$

and the cost function

$$\begin{aligned} J(\mathbf{v},\mathbf{G}) = {} & \frac{1}{2}\sum_{i=0}^{n}\left\langle \mathbf{R}^{-1}H_i(\mathbf{x}_i) - \mathbf{y}^{\mathrm{o}}_i,\, H_i(\mathbf{x}_i) - \mathbf{y}^{\mathrm{o}}_i\right\rangle \\ & + \frac{1}{2}\sum_{i=1}^{n}\Big\langle \mathbf{G}_i\left(\mathbf{y}^{\mathrm{o}}_i - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right), \\ & \quad \mathbf{G}_i\left(\mathbf{y}^{\mathrm{o}}_i - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right)\Big\rangle \\ & + \frac{1}{2}\langle \mathbf{v},\mathbf{v}\rangle. \end{aligned} \tag{16}$$

This change of variable modifies the adjoint equations

$$\begin{cases} \mathbf{x}^*_n = \mathbf{H}^{\mathrm{T}}_n\mathbf{R}^{-1}\left(H_n(\mathbf{x}_n) - \mathbf{y}^{\mathrm{o}}_n\right) \\ \mathbf{x}^*_i = \left(\mathbf{M}_{(t_{i+1},t_i)} - \mathbf{Q}^{1/2}\mathbf{G}_{i+1}\mathbf{H}_{i+1}\mathbf{M}_{(t_{i+1},t_i)}\right)^{\mathrm{T}}\mathbf{x}^*_{i+1} \\ \qquad + \mathbf{H}^{\mathrm{T}}_i\mathbf{R}^{-1}H_i(\mathbf{x}_i) - \mathbf{y}^{\mathrm{o}}_i\big) - \mathbf{M}^{\mathrm{T}}_{(t_{i+1},t_i)}\mathbf{H}^{\mathrm{T}}_{i+1}\mathbf{G}^{\mathrm{T}}_{i+1} \\ \qquad \times \mathbf{G}_{i+1}\big(\mathbf{y}^{\mathrm{o}}_{i+1} - H_{i+1}\left(M_{(t_{i+1},t_i)}(\mathbf{x}_i)\right)\big). \end{cases} \tag{17}$$

The gradients become

$$\begin{cases} \nabla_{\mathbf{v}} J = \mathbf{B}^{\mathrm{T}/2}\mathbf{x}^*_0 + \mathbf{v} \\ \nabla_{\mathbf{G_i}} J = \mathbf{Q}^{\mathrm{T}/2}\mathbf{x}^*_i\left(\mathbf{y}^{\mathrm{o}}_i - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right)^{\mathrm{T}} \\ \qquad + \mathbf{G}_i\left(\mathbf{y}^{\mathrm{o}}_i - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right) \\ \qquad \times\left(\mathbf{y}^{\mathrm{o}}_i - H_i\left(M_{(t_i,t_{i-1})}(\mathbf{x}_{i-1})\right)\right)^{\mathrm{T}} \end{cases} \tag{18}$$

Note that this second change of variable cannot be considered as a good preconditioner because $\mathbf{Q}^{1/2}$ is not the square root of the inverse of the Hessian of J_{nudg}. Thus the minimisation efficiency can be improved by considering $J_{\mathrm{nudg}} = \frac{1}{2}\sum_{i=1}^{n}\langle\tilde{\mathbf{Q}}^{-1}\mathbf{G}_i, \mathbf{G}_i\rangle$ (before the change of variable), where $\mathbf{G}_i$ is a state dimension vector.

Although the optimisation problem is modified and the Kalman equivalence (see Appendix A) is lost, the assimilation provides roughly the same quality of results compared with the preceding J_{nudg}, and the number of needed iterations is highly reduced (about 1/10).

In the following, in both **B** and **Q** operators the correlation scales are about 250 km.

The minimisation method used in the following experiment is M1QN3 Quasi-Newton with limited memory developed at INRIA by Gilbert and Lemarechal, (1989). Both classical 4D-Var and 4D-ON methods have been implemented with the PALM modular data assimilation system. This software package allows a full modularity by splitting data assimilation algorithms into elementary units (see Appendix B and/or

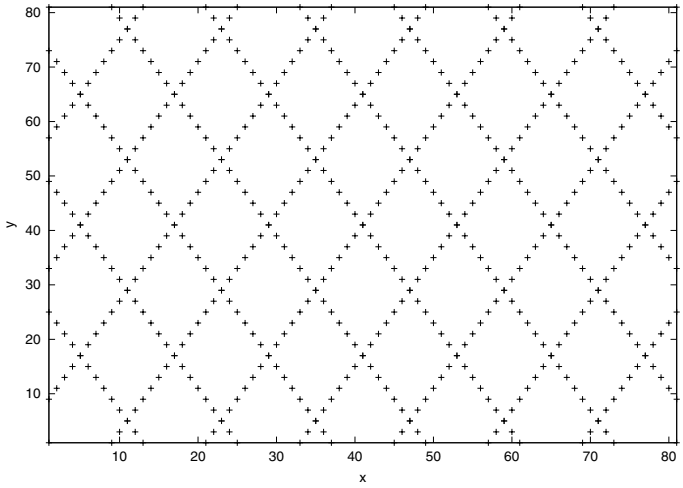

Fig. 3. Location of the observations (ground track of satellites).

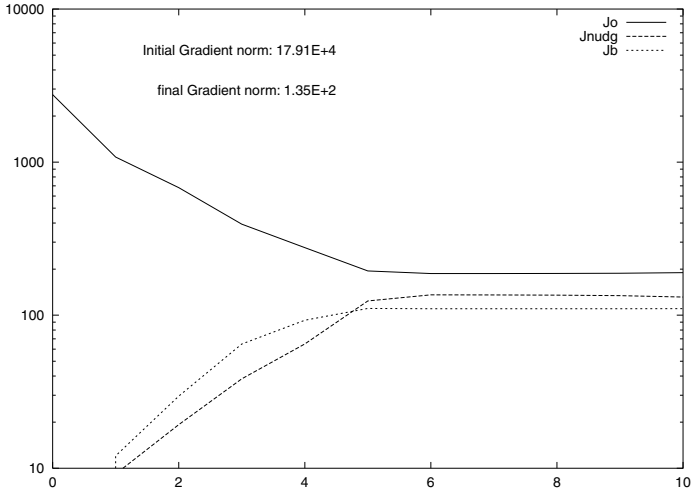

Fig. 4. Evolution of the value of the different terms of the cost function.

Piacentini et al., 2000); therefore it has been of a great help in the construction and the handling of these two methods.

5.3. Numerical experiment

In this section we will compare the 4D-ON results with classical 4D-Var ones. In both cases, the control vectors is set to 0 at the beginning of the assimilation. The assimilation is carried out for 30 d. This time window represents a relevant time scale of the ocean model, furthermore the tangent linear hypothesis,[1] which ensures the validity of the adjoint computation of the gradient, is still valid. Following the results of the Burger experiment we only focus on nudging coefficient as time-dependant scalar coefficients. The observations are simulated as satellite tracks (Fig. 3) with one track every 3 d. This is not a favourable case for nudging because of the lack of information provided by the observations; indeed only 500 grid points are observed, whereas the state vector size is about 20 000. In Fig. 4 values of the different terms of the

[1] $M_{(t_0,t_N)}(\mathbf{x}^{\mathrm{b}} + \delta\mathbf{x}_0) = M_{(t_0,t_N)}(\mathbf{x}^{\mathrm{b}}) + \mathbf{M}_{(t_0,t_N)}\delta\mathbf{x}_0$.

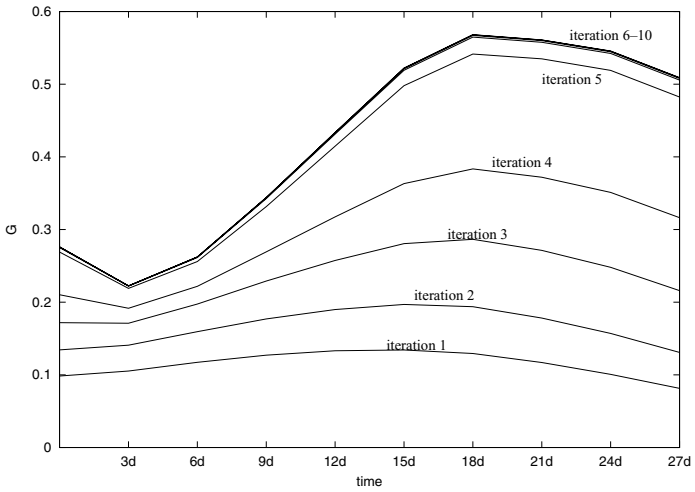

Fig. 5. Evolutions of the nudging coefficient according to the number of iterations.

cost function and the initial and the final values of the gradient are shown. The norm of the gradient decreases by six orders of magnitude during the minimisation. We note a rapid decrease in the J_o term of the cost function during the first few iterations. After six iterations the algorithm has already converged. Figure 5 shows the corresponding evolution of the nudging coefficient $\mathbf{G}_h$ according to time at several steps of the minimisation, and confirms the convergence after six iterations. The 4D-Var (not shown) requires almost the same number of iterations.

Due to the undersampling of the observation, comparing the different levels of RMS (Figs. 6–9) (computed thanks to the knowledge of the "true" trajectory), it can be seen that the former version of 4D-ON (long-dashed line) gives slightly better results compared to 4D-Var (full line) and poor results compared to smoothed 4D-ON (with the change of variable) (short-dashed line, Fig. 7). The non-smoothed 4D-ON corrects the model only on observed locations, therefore it is not consistent with the other coordinates, and then gravity waves can be created to return to a more

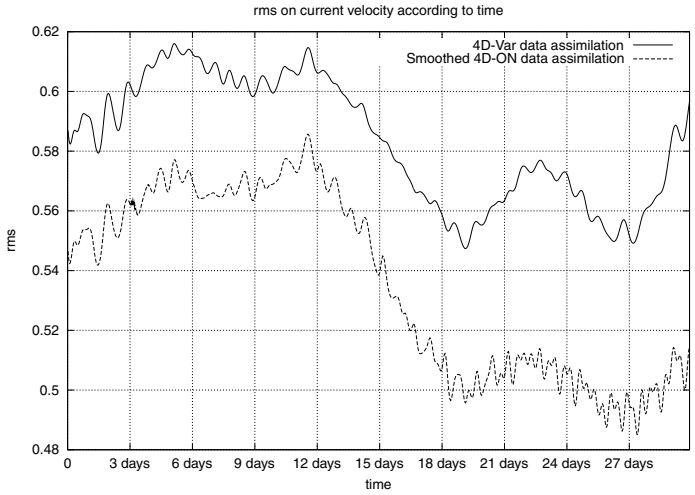

Fig. 6. RMS of the error on the current velocity for smoothed version of the 4D-ON compared to 4D-Var.

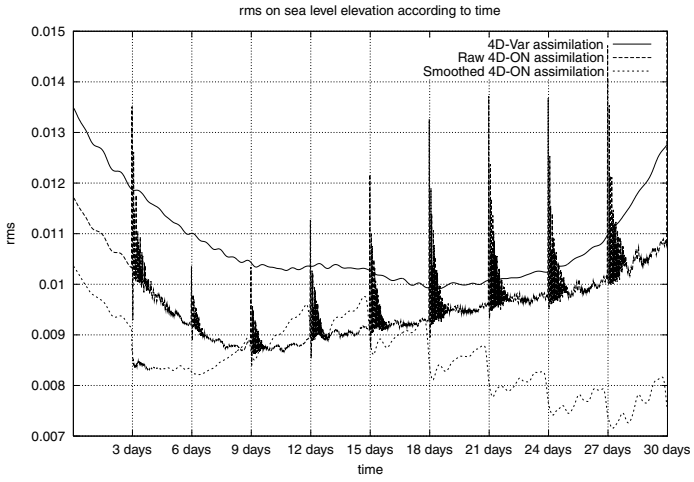

Fig. 7. RMS of the error of SSH for smoothed version of the 4D-ON compared to 4D-Var and raw 4D-ON.

coherent state (extreme fluctuations in Fig. 7). Moreover, the J_o term is only computed by the corrected terms, so the cost function does not really reflect the gap between reality and model state. Indeed the corrected state can be a smooth field with some Dirac functions on observed locations, and the J_o term will be very small.

The smoothed version of 4D-ON allows undersampling of the observation, replacing a Dirac correction by a Gaussian correction (with help from $\mathbf{Q}^{1/2}$). In addition to spatial smoothing, a temporal smoothing is performed multiplying the correcting term by a time-decreasing coefficient. The time-smoothing distance is 6 h; this parameter has been tuned by hand, and a more generalistic estimation may be explored as a further development of this method. If this temporal smoothing is disabled, the level of RMS remains similar to the full smoothing one, but some light oscillations reappear. This may due to the fact that the nudging correction is not enough balanced for the model and therefore some gravity waves may be induced.

All these developments permit 4D-ON to give noticeable improvement to classical 4D-Var for the same number of iterations. On the observed component (sea surface elevation) the improvement compared to 4D-Var is very significant (RMS of the error is up to 50% less, see Fig. 7). On the other components (current velocity, Fig. 6) thanks to **K**, the balance operator included in **Q**, the gain in RMS error remains about 10%.

The introduction of smoothing implies, for this shallow-water model, that the computational cost of one iteration of 4D-ON is about 10% more expensive than a 4D-Var one, and may be smaller for bigger models. Compared to the gain provided by the 4D-ON this overhead is not a limitation.

Due to the risk of shocks induced by nudging corrections, we want to check if the analysed state obtained at the end of the 4D-ON assimilation window is consistent with the dynamics and will provide a quality forecast. Figures 8 and 9 show that there is no particular problem: after the assimilation on the left part of the graph (under the grid) a one month forecast is performed starting from analysed state at the end of the assimilation window. Due to model errors the RMS of error becomes more and more important, but the deviation is roughly the same for 4D-Var as for smoothed 4D-ON, i.e. the analysed state obtained by optimal nudging is as consistent as that obtained by classical 4D-Var (but closer to the true state).

6. Conclusion

This paper presents the 4D-ON, a new data assimilation method, starting from 4D-Var and nudging. This methods allows are to correct the model errors through the information contained in the innovation residual (the remaining misfit between analysed state and observation). A direct application of this method improves the results of the 4D-Var but may induce gravity waves. Even if the analysed state provided by the 4D-ON at the end of the assimilation window is better the the classical 4D-Var one, these gravity waves are not desirable in an operational data-assimilation

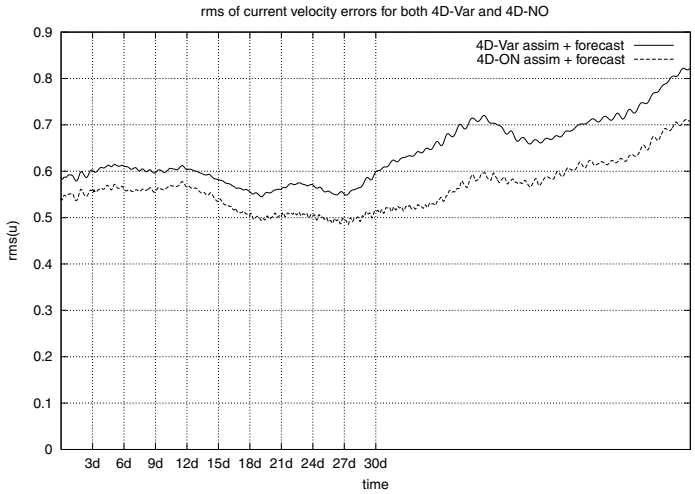

Fig. 8. RMS of the error on current velocity for smoothed version of the 4D-ON compared to 4D-Var on assimilation window and one month forecast.

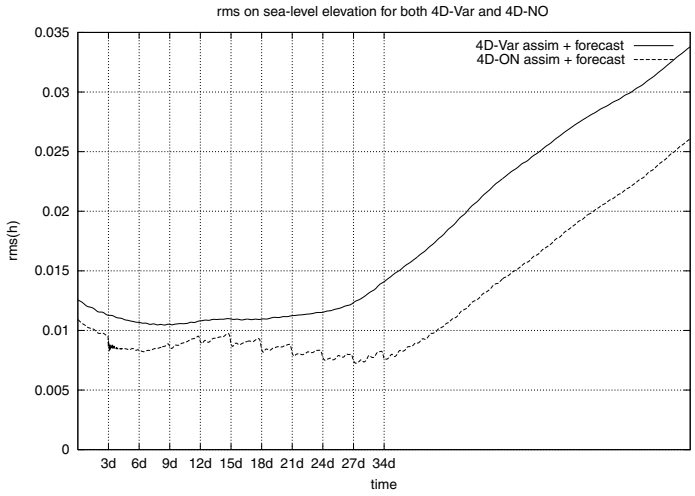

Fig. 9. RMS of SSH for smoothed version of the 4D-ON compared to 4D-Var on assimilation window and one month forecast.

scheme. It is shown that, provided the change of variable introduced in Section 5.2 is done, these waves can be avoided. Indeed, owing to spatial and temporal smoothing, the (smoothed) 4D-ON provides a balanced state of the system during the whole assimilation window.

The experiment on Burger's equation (Section 4) shows that control of the entire **G** matrix is not necessary and even harmful. Therefore the control can be restricted to a few scalar coefficients without a loss of quality. This allows us to use a configuration of the 4D-ON that is dramatically cheaper than the 4D-Var with an imperfect model. Moreover, this new scheme is very easy to implement starting from an existing 4D-Var (little algorithm modification, since all new terms are already calculated), even if the theory is quite complicated.

The run-time extra computational cost induced by the introduction of the control of nudging terms in addition to initial conditions is about 10%. Thanks to the

current growth of computational means, this extra cost is not really a limitation to practical applications. To find an adequate balance between both regularisation terms J_b an J_{nudg} requires care, and the time spent for this research may be added to the extra computational cost. However, this balance only depends on the kind of the observation dataset and on the model, therefore it needs to be set only once. For these reasons, 4D-ON may be successfully implemented in the near future on a variety of meteorological or oceanographic problems.

In parallel to its implementation in realistic models, further development can be carried out. In the considered case $\mathbf{C_Q}$ the diffusion operator included in operator $\mathbf{Q}$ is the same as the background diffusion operator $\mathbf{C_B}$. However, there is no reason for this: the model error and the background error do not necessarily have the same correlation structures. Finding the appropriate balance could be a natural future development of this method.

Appendix A: Links between the Kalman filter and optimal nudging

In this section we will point out that in a linear case, the Kalman filter and the optimal nudging method introduced in previous sections, will provide the same results. This equivalence is based on a demonstration provided by Li and Navon (2001) for a more classical 4D-Var.

Let now consider optimal nudging for a discrete linear model where the $\mathbf{G}_i$ are considered to be $n \times m$ matrices:

$$\begin{cases} \mathbf{x}_0 = \mathbf{x}^{\text{b}} + \delta\mathbf{x}_0 \\ \mathbf{x}_{i+1} = \mathbf{M}_{(t_i,t_{i+1})}\mathbf{x}_i \\ \qquad + \mathbf{G}_{i+1}\left(\mathbf{y}^{\text{o}}_{i+1} - \mathbf{H}_{i+1}\left(\mathbf{M}_{(t_i,t_{i+1})}\mathbf{x}_i\right)\right). \end{cases} \tag{19}$$

For the sake of simplicity, we note $\mathbf{d}_i = \mathbf{y}^{\text{o}}_i - \mathbf{H}_i(\mathbf{M}_{(t_i,t_i)}\mathbf{x}_{i-1})$ and $\mathbf{c} = (\delta\mathbf{x}_0, \mathbf{G}_1, \ldots, \mathbf{G}_n)$. We would like to minimise the cost function:

$$J(\mathbf{c}) = \frac{1}{2}\sum_{i=0}^{n}\left\langle \mathbf{R}^{-1}\left(\mathbf{H}_i(\mathbf{x}_i) - \mathbf{y}^{\text{o}}_i\right), \mathbf{H}_i(\mathbf{x}_i) - \mathbf{y}^{o}_i\right\rangle + \frac{1}{2}\sum_{i=0}^{n}\left\langle \mathbf{Q}_i^{-1}\mathbf{G}_i\mathbf{d}_i, \mathbf{G}_i\mathbf{d}_i\right\rangle + \left\langle \mathbf{B}^{-1}\delta\mathbf{x}_0, \delta\mathbf{x}_0\right\rangle$$

where $\langle .\,, .\rangle$ denotes the "classical" Euclidean inner product, and $(.\,, .)$ the Frobenius inner product between two matrices:

$$(A,\, B) = \sum_j \sum_i a_{ij} \sum_i a_{b_{ij}}.$$

First of all, we can note that there exists a relationship between those two inner products. If $\mathbf{X}$ is an n vector, $\mathbf{Y}$ an m vector and B an $n \times m$ matrix then

$$\langle \mathbf{X},\, B\mathbf{Y}\rangle = (\mathbf{X}\cdot\boldsymbol{Y}^{\text{T}},\, B).$$

If the model errors are assumed to be Gaussian, Li and Navon (2001) have shown the additive properties of a 4D-Var, i.e. solving the optimisation problem on the whole data assimilation period provides the same results as splitting our data assimilation period in two subspaces $1 \ldots n_1$ and $n_1 \ldots n$ and defining two cost functions as:

$$J_1(\mathbf{c}) = \frac{1}{2}\left\langle \mathbf{B}^{-1}\delta\mathbf{x}_0,\, \delta\mathbf{x}_0\right\rangle + \frac{1}{2}\sum_{i=0}^{n_1}(\ldots)$$

and

$$J_2(\mathbf{c}) = \frac{1}{2}\left\langle \left(\mathbf{P}^{a}_{\mathbf{x}_{n_1}}\right)^{-1}\left(\mathbf{x}_{n_1} - \hat{\mathbf{x}}_{n_1}\right), \mathbf{x}_{n_1} - \hat{\mathbf{x}}_{n_1}\right\rangle + \frac{1}{2}\sum_{i=n_1+1}^{n}(\ldots)$$

where $\mathbf{P}^{a}_{\mathbf{x}_{n_1}}$ is the analysis error covariance matrix of first data assimilation subsequence (which can be obtained by the computation of the Hessian matrix of J_1 (Rabier and Courtier, 1992) and $\hat{\mathbf{x}}_{n_1}$ is the analysed state (the result of this assimilation).

In order to compare optimal nudging with the Kalman filter, we assume that we have already performed the data assimilation on a former subspace of observations$\{\mathbf{y}_1, \ldots, \mathbf{y}_{k-1}\}$, and so we know $\mathbf{P}^{\text{a}}_{\mathbf{x}_{n_1}}$ and $\hat{\mathbf{x}}_{n_1}$.

We wish to minimize the following cost function:

$$J_k(\mathbf{c}_k) = \frac{1}{2}\left\langle \left(\mathbf{P}^{\text{a}}_{\mathbf{x}_{k-1}}\right)^{-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}), \mathbf{x}_{n_1} - \hat{\mathbf{x}}_{n_1}\right\rangle + \frac{1}{2}\left\langle \mathbf{R}^{-1}\left(\mathbf{H}_k(\mathbf{x}_k) - \mathbf{y}^{\text{o}}_k\right), \mathbf{H}_k(\mathbf{x}_k) - \mathbf{y}^{\text{o}}_k\right\rangle + \left\langle \mathbf{Q}_k^{-1}\mathbf{G}_k\mathbf{d}_k, \mathbf{G}_k\mathbf{d}_k\right\rangle. \tag{20}$$

Using

$$\mathbf{x}_{k-1} = \mathbf{M}^{-1}_{k,k-1}(\mathbf{x}_k - \mathbf{G}_k\mathbf{d}_k)$$

the cost function could be rewritten as:

$$J_k = \frac{1}{2}\left\langle \left(\mathbf{P}^{\mathrm{a}}_{\mathbf{x}_{k-1}}\right)^{-1}\left(\mathbf{M}^{-1}_{k,k-1}(\mathbf{x}_k - \mathbf{G}_k\mathbf{d}_k) - \hat{\mathbf{x}}_{k-1}\right), \right.$$
$$\left. \mathbf{M}^{-1}_{k,k-1}(\mathbf{x}_k - \mathbf{G}_k\mathbf{d}_k) - \hat{\mathbf{x}}_{k-1}\right\rangle$$
$$+ \frac{1}{2}\left\langle \mathbf{R}^{-1}\mathbf{H}_k(\mathbf{x}_k) - \mathbf{y}^{\mathrm{o}}_k, \mathbf{H}_k(\mathbf{x}_k) - \mathbf{y}^{\mathrm{o}}_k\right\rangle$$
$$+ \frac{1}{2}\left\langle \mathbf{Q}^{-1}_k\mathbf{G}_k\mathbf{d}_k, \mathbf{G}_k\mathbf{d}_k\right\rangle. \tag{21}$$

Then our problem is to find $(\hat{\mathbf{x}}_k, \hat{\mathbf{G}}_k)$ as

$$\begin{cases} \dfrac{\partial J_k}{\partial \mathbf{x}_k} = 0 \\ \dfrac{\partial J_k}{\partial \mathbf{G}_k} = 0. \end{cases} \tag{22}$$

Using (20), eq. (22) becomes:

$$\begin{cases} 0 = \mathbf{H}^{\mathrm{T}}_k\mathbf{R}^{-1}\left(\mathbf{H}_k(\hat{\mathbf{x}}_k) - \mathbf{y}^{\mathrm{o}}_k\right) + \mathbf{M}^{-\mathrm{T}}_{k,k-1}\left(\mathbf{P}^{\mathrm{a}}_{\mathbf{x}_{k-1}}\right)^{-1} \\ \qquad \times\left[\mathbf{M}^{-1}_{k,k-1}(\hat{\mathbf{x}}_k - \hat{\mathbf{G}}_k\mathbf{d}_k) - \hat{\mathbf{x}}_{k-1}\right] \\ 0 = \mathbf{Q}^{-1}_k\hat{\mathbf{G}}_k\mathbf{d}_k\mathbf{d}^{\mathrm{T}}_k + \mathbf{M}^{-\mathrm{T}}_{k,k-1}\left(\mathbf{P}^{\mathrm{a}}_{\mathbf{x}_{k-1}}\right)^{-1} \\ \qquad \times\mathbf{M}^{-1}_{k,k-1}\hat{\mathbf{G}}_k\mathbf{d}_k\mathbf{d}^{\mathrm{T}}_k + \mathbf{M}^{-\mathrm{T}}_{k,k-1}\left(\mathbf{P}^{\mathrm{a}}_{\mathbf{x}_{k-1}}\right)^{-1} \\ \qquad \times\left[\hat{\mathbf{x}}_{k-1} - \mathbf{M}^{-1}_{k,k-1}\hat{\mathbf{x}}_k\right]. \end{cases} \tag{23}$$

According to eq. (23) we obtain

$$\hat{\mathbf{G}}_k\mathbf{d}_k = \mathcal{H}^{-1}_{\mathbf{Q}_k}\mathbf{M}^{-\mathrm{T}}_{k,k-1}\left(\mathbf{P}^{\mathrm{a}}_{\mathbf{x}_{k-1}}\right)^{-1}\left[\hat{\mathbf{x}}_{k-1} - \mathbf{M}^{-1}_{k,k-1}\hat{\mathbf{x}}_k\right] \tag{24}$$

where $\mathcal{H}_{\mathbf{Q}_k} = \mathbf{Q}^{-1}_k + \mathbf{M}^{-\mathrm{T}}_{k,k-1}(\mathbf{P}^{\mathrm{a}}_{\mathbf{x}_{k-1}})^{-1}\mathbf{M}^{-1}_{k,k-1}$

Using eq. (24) in eq. (23) leads to

$$0 = \mathbf{H}^{\mathrm{T}}_k\mathbf{R}^{-1}\left(\mathbf{y}^{\mathrm{o}}_k - \mathbf{H}_k(\hat{\mathbf{x}}_k)\right) + \mathbf{M}^{-\mathrm{T}}_{k,k-1}\left(\mathbf{P}^{\mathrm{a}}_{\mathbf{x}_{k-1}}\right)^{-1}$$
$$\times\left(I + \mathbf{M}^{-1}_{k,k-1}\mathcal{H}^{-1}_{\mathbf{Q}_k}\mathbf{M}^{-\mathrm{T}}_{k,k-1}\left(\mathbf{P}^{a}_{\mathbf{x}_{k-1}}\right)^{-1}\right)$$
$$\times\left[\hat{\mathbf{x}}_{k-1} - \mathbf{M}^{-1}_{k,k-1}\hat{\mathbf{x}}_k\right].$$

We can rewrite it as

$$\left[\mathbf{H}^{\mathrm{T}}_k\mathbf{R}^{-1}\mathbf{H}_k + \mathbf{M}^{-\mathrm{T}}_{k,k-1}\left(\mathbf{P}^{\mathrm{a}}_{\mathbf{x}_{k-1}}\right)^{-1}\mathbf{M}^{-1}_{k,k-1} + \mathbf{M}^{-T}_{k,k-1}\right.$$
$$\left.\times\left(\mathbf{P}^{\mathrm{a}}_{\mathbf{x}_{k-1}}\right)^{-1}\mathbf{M}^{-1}_{k,k-1}\mathcal{H}^{-1}_{\mathbf{Q}_k}\mathbf{M}^{-\mathrm{T}}_{k,k-1}\left(\mathbf{P}^{\mathrm{a}}_{\mathbf{x}_{k-1}}\right)^{-1}\mathbf{M}^{-1}_{k,k-1}\right]$$
$$\times\left[\mathbf{M}_{k,k-1}\hat{\mathbf{x}}_{k-1} - \hat{\mathbf{x}}_k\right] = \mathbf{H}^{T}_k\mathbf{R}^{-1}\left[\mathbf{M}_{k,k-1}\hat{\mathbf{x}}_{k-1} - \mathbf{y}^{o}_k\right]. \tag{25}$$

If we pose

$$\mathbf{P}^{f}_k = \mathbf{M}_{k,k-1}\mathbf{P}^{\mathrm{a}}_{\mathbf{x}_{k-1}}\mathbf{M}^{\mathrm{T}}_{k,k-1} + \mathbf{Q}_k \tag{26}$$

using a matrix inversion lemma[2]:

$$\mathcal{H}^{-1}_{\mathbf{Q}_k} = \mathbf{Q}_k - \mathbf{Q}_k\left(\mathbf{P}^{\mathrm{f}}_k\right)^{-1}\mathbf{Q}_k$$
$$= \mathbf{M}_{k,k-1}\mathbf{P}^{\mathrm{a}}_{\mathbf{x}_{k-1}}\mathbf{M}^{\mathrm{T}}_{k,k-1}\left(\mathbf{P}^{\mathrm{f}}_k\right)^{-1}\mathbf{Q}_k. \tag{27}$$

Introducing eq. (27) in eqs. (25) and (26) $\times\ (\mathbf{P}^{\mathrm{f}}_k)^{-1}$

$$\left[\mathbf{H}^{\mathrm{T}}_k\mathbf{R}^{-1}\mathbf{H}_k + \left(\mathbf{P}^{\mathrm{f}}_k\right)^{-1}\right]\left[\mathbf{M}_{k,k-1}\hat{\mathbf{x}}_{k-1} - \hat{\mathbf{x}}_k\right]$$
$$= \mathbf{H}^{\mathrm{T}}_k\mathbf{R}^{-1}\left[\mathbf{M}_{k,k-1}\hat{\mathbf{x}}_{k-1} - \mathbf{y}^{\mathrm{o}}_k\right] \tag{28}$$

we can retrieve, using another matrix formula[3]:

$$\hat{\mathbf{x}}_k = \mathbf{M}_{k,k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{P}^{\mathrm{f}}_k\mathbf{H}^{\mathrm{T}}_k\left[\mathbf{H}_k\mathbf{P}^{\mathrm{f}}_k\mathbf{H}^{\mathrm{T}}_k + \mathbf{R}\right]^{-1}$$
$$\times\left[\mathbf{y}^{\mathrm{o}}_k - \mathbf{M}_{k,k-1}\hat{\mathbf{x}}_{k-1}\right].$$

This is exactly the expression of $\hat{\mathbf{x}}_k$ in the Kalman filter. It shows that, in theory and when the model is linear, optimal nudging leads to the same results as the Kalman filter, which is optimal.

Appendix B: The PALM assimilation coupler

PALM aims to provide a general structure for the modular implementation of a data assimilation suite. In this system, a data assimilation algorithm is described as a set of appropriate sequences of elementary "units" such as the forecast model, the observation operator, the adjoint model, the operators approximating the error correlation matrices, etc. This approach allow one to separate the physical part of the problem from the algebraic part.

PALM ensures the scheduling and the synchronisation of the units, drives the communication of the fields exchanged by the units and performs the algebra. This goal is achieved without a significant loss of performances if compared to a standard implementation.

This approach proves to be useful for any kind of dynamic coupling, not only for data assimilation suites.

[2] $(\mathbf{A}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{A} + \mathbf{C}^{-1})^{-1} = \mathbf{C} - \mathbf{C}\mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{C}\mathbf{A}^{\mathrm{T}} + \mathbf{B})^{-1}\mathbf{A}\mathbf{C}$.

[3] $(\mathbf{A}^{-1} + \mathbf{B}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{C}^{-1} = \mathbf{A}\mathbf{B}^{\mathrm{T}}(\mathbf{C} + \mathbf{B}\mathbf{A}\mathbf{B}^{\mathrm{T}})^{-1}$ whenever the inverses exist (see Wunsch, 1996).

REFERENCES

Alekseev, K. Q. and Navon, I. M. 2001. The analysis of an ill posed problem using multiscale resolution and second order adjoint techniques. *Comput. Meth. Appl. Mechan. Eng.* **190**, 15–17; 1937–1953.

Adcroft, A. and Marshall, D. 1998. How slippery are piecewise-constant coastlines in numerical ocean models. *Tellus* **50A**, 95–108.

Cohn, S. E. 1997. An Introduction to Estimation Theory *J. Meteorol. Soc. Jpn.* **75**, 1B, 257–288.

Courtier, P. 1997. Dual formulation of four dimensional variational assimilation. *Q. J. R. Meteorol. Soc.* **123**, 2449–2462.

Derber, J. 1989. A variational continuous assimilation technique. *Mon. Wea. Rev.* **117**, 2437–2446.

Derber, J. and Bouttier, F. 1999. A reformulation of the background error covariance in the ECMWF global data assimilation system. *Tellus* **51A**, 195–221.

Gilbert, J.-C. and Lemarechal, C. 1989. Some numerical experiment with variable storage quasi-Newton algorythms. *Math. Prog.* **B25**, 407–435.

Griffith, A. K. and Nichols, N. K. 2001. Adjoint techniques in data assimilation for estimating model error. *J. Flow, Turbulence Combust.* **65**, 469–488.

Ide, K., Courtier, P., Ghill, M. and Lorenc, A. C. 1997. Unified notation for data assimilation: operational, sequential and variational. *J. Meterol. Soc. Jpn.* **75-B**, 181–189.

Jazwinski, A. H. 1970. *Stochastic processes and filtering theory.* Academic Press, New York.

Kalman, R. E. 1960. A new approach to linear filtering and prediction problems *Trans. ASME: J. Basic Eng.* **35–45**, (March 1960).

Krishnamurti, T. N., Jishan, X., Bedi, H. S., Ingles, K. and Oosterhof, D. 1991. Physical initialization for numerical weather prediction over the tropics. *Tellus* **43A**, 53–81.

Le Dimet, F.-X. and Charpentier, I. 1998. Méthodes du second ordre en assimilation de données. In: *Equations aux dérivées partielles et applications*. Gauthier-Vilars, Paris, 107–125.

Lellouche, J.-M., Devenon, J.-L. and Dekeyser, I. 1994. Boundary control of Burger's equation. A numerical approach. *Comput. Math. Appl.* **28**, 33–44.

Li, Z. and Navon, I. M. 2001. Optimality of variational data assimilation and its relationship with the Kalman filter and smoother. *Q. J. R. Meteorol. Soc.* **127B**, 661–684.

Lorenc, A. C. 1986. Analysis methods for numerical weather prediction. *Q. J. R. Meteorol. Soc.* **112**, 1177–1194.

Lorenc, A. C., Bell, R. S. and Macpherson, B. 1991. The Meteorological Office analysis correction data assimilation scheme. *Q. J. R. Meteorol. Soc.* **117**, 59–89.

Lyne, W. H., Swinbank, R. and Birch, N. T. 1982. A data assimilation experiment and the global circulation during the FGGE special observing periods. *Q. J. R. Meteorol. Soc.* **108**, 575–594.

Ménard, R. 1993. Kalman filtering of Burger's equation and its application to atmospheric data assimilation. Ph.D. Thesis, McGill University, Montréal, Canada.

Navon, I. M. 1997. Practical and theoretical aspects of adjoint parameter estimation and identifiability in meteorology and oceanography. *Dynam. Atmos. Oceans.* **27**, 55–79.

Piacentini, A. and PALM group 1999. PALM: A modular data assimilation system. *Proc. Third WMO Symp. on Data Assimilation*, *7–11 June 1999, Québec City, Canada*. WMO/TD-986, 321–323.

Rabier, F. and Courtier, P. 1992. Four-dimensional assimilation in presence of baroclinic instability. *Q. J. R. Meteorol. Soc.* **118**, 649–672.

Thepaut, J. N. and Moll, P. 1990. Variational Inversion of simulated TOVS radiances using the adjoint technique. *Quart. J. R. Meteorol. Soc.* **116**, 1425–1448.

Tikhonov, A. V. and Arserin, V. 1997. *Solution of ill posed problems.* Winston & Sons, Washington DC. 224 pp.

Vidard, P. A., Blayo, E., Le Dimet, F.-X. and Piacentini, A. 2000. 4D-Variational data analysis with imperfect model. reduction of the size of control. *J. Flow, Turbulence Combust*, **65**, 489–504.

Wang, Z., Navon, I. M., Le Dimet, F.-X. and Zou, X. 1992. The second order analysis: Theory and applications. *Meteorol. Atmos. Phys.* **50**, 3–21.

Weaver, A. and Courtier, P. 2001. Correlation modeling on the sphere using a generalized diffusion equation. *Q. J. R. Meteorol. Soc.* **127**, 1815–1846.

Wunsch, C. 1996. *The ocean circulation inverse problem.* (Cambridge Univesity Press, Cambridge), 99 pp.

Yang, W., Navon, I. M. and Courtier, P. 1996. A new Hessian preconditioning method applied to variational data assimilation experiment using adiabatic version of NASA/GEOS-1G CM. *Mon. Wea. Rev.* **124**(5), 1000–1017.

Zou, X., Navon, I. M. and Le Dimet, F.-X. 1992. An optimal nudging data assimilation scheme using parameter estimation. *Q. J. R. Meteorol. Soc.* **118**, 1163–1186.