Several data assimilation methods for geophysical problems

Didier Auroux

Several data assimilation methods for geophysical problems

Didier Auroux

Laboratoire MIP, Université Paul Sabatier Toulouse 3,
31062 Toulouse cedex 9, France.
Tel: +33 5 61 55 76 42 - Fax: +33 5 61 55 83 85
auroux@mip.ups-tlse.fr

March 16, 2006

Abstract. In this paper, we present an overview of various data assimilation methods, in order to identify the initial condition of a geophysical system and reconstruct its evolution in time and space. We first present the well known four dimensional variational adjoint method, the 4D-VAR algorithm, and then the four dimensional variational dual method, the 4D-PSAS algorithm, extended to nonlinear models. We present then an improved sequential data assimilation algorithm, the SEEK filter. We finally introduce a new simple algorithm, the Back and Forth Nudging. Some theoretical and numerical results about the BFN algorithm are finally given.

Keywords: data assimilation, geophysical fluids, 4D-VAR, 4D-PSAS, SEEK filter, nudging

1. Introduction

It is now well established that the quality of weather and ocean circulation forecasts is highly dependent on the quality of the initial conditions. Geophysical fluids (air, atmospheric, oceanic, surface or underground water) are governed by the general equations of fluid dynamics. Geophysical processes are hence non linear because of their fluid component. Such non linearities impose a huge sensitivity to the initial conditions, and then an ultimate limit to deterministic prediction (estimated to be about two weeks for weather prediction for example). This limit is still far from being reached, and substantial gain can still be obtained in the quality of forecasts. This can be obtained through improvement of the observing system itself, but also through improvement of the geophysical models used to modelize the geophysical processes. For example, a major problem comes from the fact that sub-scales processes could be associated with extremely large fluxes of energy. Seeking a numerical solution to the equations requires discretizing the equations, and therefore cutting off in the scales. It

will be crucial to represent the fluxes of energy associated to subgrid processes by some additional terms in the equations (Holland, 1978; Pedlosky, 1979).

Over the past twenty years, observations of ocean and atmosphere circulation have become much more readily available, as a result of new satellite techniques and international field programs (MERCATOR, CLIPPER, GODAE, ...). In the case of the ocean modelling, the use of altimeter measurements has provided extremely valuable information about the sea-surface height, and then has allowed the oceanographic community to study more precisely both the general circulation of the ocean and the local dynamics of some particular regions (the Gulf Stream area, for example, but also the Kuroshio extension, the Antarctic circumpolar current and the tropical oceans). Geostationnary satellites also provide information on the wind by estimating the shifting of clouds considered as lagrangian tracers. Polar orbiting satellites are used for the estimation of the atmospheric vertical temperature profiles. Generally, radiances are measured and then temperatures are estimated as the solution of an inverse problem.

Meteorologic and oceanographic data are currently extremely heterogeneous, both in nature, density and quality, but their number is still smaller than the degree of freedom of the models. The growth of the available computing ressources indeed allows refinements of the grid size of general circulation models.

Environmental scientists are increasingly turning to inverse methods for combining in an optimal manner all the sources of information coming from theory, numerical models and data. Data assimilation (DA) is precisely the domain at the interface between observations and models which makes it possible to identify the global structure of a system from a set of discrete space-time data. DA covers all the mathematical and numerical techniques in which the observed information is accumulated into the model state by taking advantage of consistency constraints with laws of time evolution and physical properties, and which allow us to blend as optimally as possible all the sources of information coming from theory, models and other types of data.

There are two main categories of data assimilation techniques (Talagrand, 1997), variational methods based on the optimal control theory (Lions, 1968) and statistical methods based on the theory of optimal statistical estimation (see, for example, (Bennett, 2002; Kalnay, 2003) for an overview of inverse methods, both for oceanography and meteorology). The first class of methods (3D-VAR, 4D-VAR, 4D-PSAS, ...) was first introduced in meteorology (Lewis et al., 1985; Le Dimet et al., 1986; Talagrand et al., 1987) and more recently for oceanic data (Thacker et al., 1988; Sheinbaum et al., 1990; Moore, 1991; Schröter et

al., 1993; Nechaev et al., 1994; Luong et al., 1998). The statistical (or sequential) methods (optimal interpolation, Kalman filter, SEEK filter, ...) were introduced in oceanography roughly fifteen years ago (Ghil, 1989; Ghil et al., 1991). The Kalman filter was extended to nonlinear cases (Jazwinski, 1970; Gelb, 1974) but it has been mostly applied in oceanography to quasi-linear situations, in particular tropical oceans (Gourdeau et al., 1992; Fukumori et al., 1993; Fukumori, 1995; Cane et al., 1996; Verron et al., 1999).

In practice, all data assimilation techniques encounter major difficulties due to computational reasons. The full Kalman filter would, in principle, require the manipulation of matrices with a dimension of typically 10^7 or 10^8 in an oceanic problem. The optimal control adjoint method often requires several hundred iterations of the minimization process to converge, thus implying an equivalent number of model runs. In this context, it is important to find new data assimilation algorithms allowing in particular a reduction of the problem dimension.

In this paper, we focus our interest on various data assimilation algorithms in order to identify the initial condition of a geophysical system and reconstruct its evolution in time and space.

We first study in Section 2 the four dimensional variational adjoint method (named 4D-VAR), using a strong constraint hypothesis (the ocean circulation model is assumed to be exact). The use of a cost function, measuring the mean-square difference between the observations and the corresponding model variables, allows us to carry out the assimilation process by an identification of the initial state of the ocean which minimizes the cost function.

Then, in Section 3, it is then possible to consider the model as a weak constraint, the adjoint state being interpreted as the Lagrange multiplier of the model equations. The linear theory of duality, extended to a nonlinear case, allows us to consider the minimization problem in a dual way. The minimization of the dual cost function is then performed in the observation space, which is smaller than the state space. Moreover, there is no more need to assume that the ocean model is exact. This dual variational method is called 4D-PSAS (Physical Space Assimilation System).

We will then study in Section 4 one of the most recent and powerful sequential DA methods, the SEEK (Singular Extended Evolutive Kalman) filter. Sequential methods are mostly based on the Kalman filtering theory, which consists in a forecast step and an analysis (or correction) step. The dimension of the error covariance matrices is a main issue, and the idea of the SEEK filter is to use low rank error covariance matrices. This allows to reduce considerably the compu-

tational cost of the filter, but this filter is also known to reduce the propagation of errors from one step to the next.

We finally introduce in Section 5 the Back and Forth Nudging (BFN) algorithm, which is a prototype of a new class of data assimilation methods, although the standard nudging algorithm is known for a couple of decades. It consists in adding a feedback term in the model equations, measuring the difference between the observations and the corresponding space states. The idea is to apply the standard nudging algorithm to the backward (in time) nonlinear model in order to stabilize it. The BFN algorithm is an iterative sequence of forward and backward resolutions, all of them being performed with an additional nudging feedback term in the model equations. As the BFN algorithm is completely new, we will give at the end of Section 5 some numerical results illustrating the convergence of the algorithm and the reconstruction of the initial condition.

2. Four dimensional variational adjoint method: 4D-VAR

Variational methods consider the equations governing the geophysical flow as constraints, and the problem is closed by using a variational principle, e.g. the minimization of the discrepancy between the model and the observations. We will first assume in this section that the model is a strong constraint.

2.1. Model and observations

Every DA method needs both a model describing the evolution of the fluid, basically a system of non linear partial differential equations, and a set of discrete observations. Firstly, we assume that the model can be written:

$$\begin{cases} \dfrac{dX}{dt} = F(X,U), \quad 0 < t < T, \\ X(0) = V, \end{cases} \tag{1}$$

where X is the state variable which describes the evolution of the system at each grid point. X depends on time, and is for operational models of large dimension (10^7 to 10^8). F is a non linear differential operator, describing the dynamics of the system. U corresponds to some internal variables of the model (parameters or boundary conditions) and may be time dependent. Finally, V is the initial condition of the system state, which is unknown. In order to use optimal control techniques, we have to define a control variable that should be identified.

Most of the time, the control is (U, V), the initial condition and the model parameters.

Secondly, we suppose that we have an observation vector X_{obs} which gathers all the data we want to assimilate. These observartions are discrete in time and space, distributed all over the assimilation period $[0, T]$, and are not in the same space as the state variable, from a geographical or a physical point of view. Therefore, we will need an observation operator C mapping the space of state into the space of observations. This operator can be non linear in some cases.

2.2. Cost function

It is now possible to define a cost function $\mathcal{J}$ measuring the discrepancy of the solution of the model associated to the control vector (U, V) and the observations X_{obs}:

$$\begin{aligned} \mathcal{J}(U,V) &= \frac{1}{2}\int_0^T \langle R^{-1}(CX - X_{obs}), CX - X_{obs}\rangle dt \\ &+ \frac{1}{2}\langle P_0^{-1}V, V\rangle + \frac{1}{2}\int_0^T \langle QU, U\rangle dt \end{aligned} \tag{2}$$

where X is the solution of (1). P_0, R and Q are covariance matrices, allowing us to introduce some a priori information about the statistics of the fields X_{obs}, V and U respectively. $\langle ., . \rangle$ is most of the time the canonical real scalar product.

The first part of the cost function quantifies the difference between the observations and the state function, and the two others act like a regularization term in the sense of Tykhonov. It is sometimes replaced by the so-called background term, which is the quadratic (with respect to the covariance matrix norm) difference between the initial optimal variable and the last prediction (Luong et al., 1998).

The inverse problem which consists in the minimization of the cost function $\mathcal{J}$ is then generally well-posed. The variational formulation of our DA problem can then be written as:

$$\begin{cases} \text{Find } (U^*, V^*) \text{ such that} \\ \mathcal{J}(U^*, V^*) = \inf\limits_{(U,V)} \mathcal{J}(U, V). \end{cases} \tag{3}$$

2.3. Gradient step

In order to minimize the cost function, we need its gradient $\nabla\mathcal{J}$. Because of the large dimension of the model state vector (usually more than 10^7), it is not possible to compute directly the gradient by using

finite difference methods. The gradient vector of the functional is then obtained by the adjoint method (Le Dimet et al., 1986; Courtier et al., 1987). Let $\hat{X}$ be the derivative of X with respect to (U,V) in the direction (u,v). Then $\hat{X}$ is solution of the following set of partial differential equations, known as the linear tangent model:

$$\begin{cases} \dfrac{d\hat{X}}{dt} = \dfrac{\partial F}{\partial X}\hat{X} + \dfrac{\partial F}{\partial U}u, \\ \hat{X}(0) = v, \end{cases} \tag{4}$$

where $\dfrac{\partial F}{\partial X}$ and $\dfrac{\partial F}{\partial U}$ represent the jacobian of the model with respect to the state variable and the model parameters respectively.

The derivative of $\mathcal{J}$ with respect to (U,V) in the direction (u,v) is then

$$\langle \hat{\mathcal{J}}(U,V),(u,v)\rangle = \int_0^T \langle R^{-1}(CX - X_{obs}), C\hat{X}\rangle dt$$

$$+\langle P_0^{-1}V, v\rangle + \int_0^T \langle Q^{-1}U, u\rangle dt.$$

If we assume that the operator C is linear (otherwise, we have to linearize it), we can introduce the so called adjoint state P (which lives in the same space as X), solution of the adjoint model (Le Dimet et al., 1986):

$$\begin{cases} -\dfrac{dP}{dt} = \left(\dfrac{\partial F}{\partial X}\right)^T P - C^T R^{-1}(CX - X_{obs}), \\ P(T) = 0. \end{cases} \tag{5}$$

We have then:

$$\langle \hat{\mathcal{J}}(U,V),(u,v)\rangle = \int_0^T \langle \frac{dP}{dt} + \left(\frac{\partial F}{\partial X}\right)^T P, \hat{X}\rangle dt$$

$$+\langle P_0^{-1}V, v\rangle + \int_0^T \langle Q^{-1}U, u\rangle dt$$

and an integration by part shows that, using (4):

$$\langle \hat{\mathcal{J}}(U,V),(u,v)\rangle = \int_0^T \langle -P, \frac{\partial F}{\partial U}u\rangle dt - \langle P(0), v\rangle$$

$$+\langle P_0^{-1}V, v\rangle + \int_0^T \langle Q^{-1}U, u\rangle dt.$$

Finally, the gradient of $\mathcal{J}$ is given by:

$$\nabla\mathcal{J}(U,V) = \begin{pmatrix} -\left(\dfrac{\partial F}{\partial U}\right)^T P + Q^{-1}U \\ -P(0) + P_0^{-1}V \end{pmatrix}. \tag{6}$$

Therefore, the gradient is obtained by a backward integration of the adjoint model, which has the same computational cost as one evaluation of $\mathcal{J}$.

2.4. Optimality system

The minimization problem (3) is then equivalent to the following optimality system:

$$\begin{cases} \dfrac{dX}{dt} = F(X,U^*), \\ X(0) = V^*, \end{cases}$$

$$\begin{cases} -\dfrac{dP}{dt} = \left(\dfrac{\partial F}{\partial X}\right)^T P - C^T R^{-1}(CX - X_{obs}), \\ P(T) = 0, \end{cases} \tag{7}$$

$$\left(\frac{\partial F}{\partial U}\right)^T P = Q^{-1}U^*,$$

$$P(0) = P_0^{-1}V^*.$$

2.5. 4D-Var algorithm computation

The determination of (U^*, V^*), solution of (3) and (7), is carried out by running a descent-type optimization method. We may use as a first guess (U_0, V_0) the result of the minimization process at the last prediction. Then, given the first guess, we use an iterative algorithm (Gilbert et al., 1989):

$$(U_n, V_n) = (U_{n-1}, V_{n-1}) + \rho_n D_n$$

where D_n is a gradient descent direction, and ρ_n is the step size.

The knowledge of (U_{n-1}, V_{n-1}) allows us to compute the corresponding solution X_{n-1} of the direct model (1), and consequently to evaluate the cost function $\mathcal{J}(U_{n-1}, V_{n-1})$. Then we solve the adjoint model (5) and compute the adjoint solution P_{n-1}, and using (6), the gradient of the cost function $\nabla\mathcal{J}(U_{n-1}, V_{n-1})$. The computation of the descent direction D_n is usually performed using conjugate gradient or Newton

type methods. Finally, the step size ρ_n is choosen to be the step size which minimizes

$$\mathcal{J}\left((U_{n-1}, V_{n-1}) + \rho D_n\right)$$

with respect to ρ. This is a one-dimensional minimization, but in case the problem is non linear, we can get a high computational cost because it will require several evaluations of $\mathcal{J}$, and hence several integrations of the model (1) (Broyden, 1969; Gilbert et al., 1989; Liu et al., 1989; Veerse et al., 2000).

2.6. Computational issues

One of the most difficult steps in the 4D-Var algorithm is the implementation of the adjoint model. Numerically, the goal is to solve the discrete optimality system, which gives the solution of the discrete direct problem, and the discrete gradient is given by the discrete adjoint model, which has to be derived from the discrete direct model, and not from the continuous adjoint model. A bad solution would be to derive the adjoint model from the continuous direct model, and then to discretize it. The good solution is to first derive the tangent linear model from the direct model. This can be done by differentiating the direct code line by line. And then one has to transpose the linear tangent model in order to get the adjoint of the discrete direct model. To carry out the transposition, one should start from the last statement of the linear tangent code and transpose each statement. The derivation of the adjoint model can be long. Sometimes, it is possible to use some automatic differentiation codes (the direct differentiation gives the tangent linear model, and the inverse differentiation provides the adjoint model) (Rostaing et al., 1994; Mohammadi et al., 2001).

Another issue is the relative ill-posedness of the problem when the model is non linear. The cost function $\mathcal{J}$ is hence not convex, and may have plenty of local minima. The optimization algorithm may then converge toward a local minimum and not the global minimum. For this reason, the choice of the initial guess is extremely important, because if it is located in the vicinity of the global minimum, one can expect a convergence toward the global minimum. Another solution is to increase the weight of the two last terms of $\mathcal{J}$ in (2), which correspond to two regularization terms with respect to the two control variables. This has to be done carefully because it can provide a physically incorrect solution: if P_0 and Q are too small, the regularization of $\mathcal{J}$ is indeed a penalization. But usually, these regularization terms are used to force the model to verify some additional physical constraints or/and to take into account some statistical information on model/observation/background errors.

2.7. Reduced-order 4D-Var

If in equation (1) the model parameters U are time dependent, the numerical implementation of the 4D-Var algorithm will consist in identifying the control vector (U, V), where V has typically a dimension of $10^7 - 10^8$ and U might have the same dimension at each time step. If there are one thousand time steps in the numerical scheme, the size of the control vector can reach $10^{10} - 10^{11}$. This is not computationally realistic. It is clearly not possible to take into account the model errors in such a way. Even it can be very costly to minimize the cost function in the entire space state.

The main idea of the reduced-order 4D-Var is to find a vector X^* which minimizes the cost function $\mathcal{J}$ in a smaller space. X^* is defined as follows:

$$X^* = X_{background} + \sum_{i=1}^{r} \lambda_i L_i, \tag{8}$$

where λ_i are chosen so that $\mathcal{J}$ is minimum, and (L_i) are orthogonal vectors of the state space. These vectors are supposed to modelize as well as possible the variability of the system. Most of the time, one uses empirical orthogonal functions (EOFs) for the choice of such vectors. Then, the minimization of the cost function takes place in a space of dimension r (Durbiano, 2001).

The same idea is used for the model parameters:

$$U = \bar{U} + \sum_{i=1}^{s} \alpha_i u_i, \tag{9}$$

where $\bar{U}$ is an estimation of the parameters, (α_i) are the new scalar control variables (instead of the vector U) and (u_i) are orthogonal vectors.

This allows to take into account the unknown terms of the model for a reasonable computational cost (Durbiano, 2001; Vidard, 2001).

3. Four dimensional dual method: 4D-PSAS

In this section, we will now consider the model as a weak constraint. It is then possible to introduce a Lagrange multiplier for this constraint (Bennett, 1992; Amodei, 1995; Courtier, 1997; Bennett, 2002).

3.1. Lagrangian

Let us consider a model operator called $\mathcal{M}$ defined on the space of the control vector by

$$\mathcal{M}(U,V) = X \tag{10}$$

where X is the solution of (1). We would like that $CX = X_{obs}$ and we will hence impose (as a weak constraint) that $C\mathcal{M}(U,V) = X_{obs}$. Let m be the Lagrange multiplier for this constraint, m lives in the same space as the observation vector X_{obs}, and we can define the following Lagrangian:

$$\mathcal{L}((U,V),m) = \mathcal{J}(U,V) + \langle m, C\mathcal{M}(U,V) - X_{obs}\rangle. \tag{11}$$

If the model $\mathcal{M}$ and the observation operator C are linear, then we have the following well known duality result for convex functions:

$$\min_{(U,V)} \mathcal{J} = \min_{(U,V)} \max_{m} \mathcal{L} = \max_{m} \min_{(U,V)} \mathcal{L}$$

3.2. Dual cost function

It is then possible to define on the observation space a dual cost function $\mathcal{J}_D$ in the following way:

$$\mathcal{J}_D(m) = -\min_{(U,V)} \mathcal{L}\left((U,V),m\right). \tag{12}$$

We have then the following result:

$$\min_{(U,V)} \mathcal{J}(U,V) = \max_{m}(-\mathcal{J}_D(m)) = -\min_{m} \mathcal{J}_D(m). \tag{13}$$

Mathematically, the minimization of $\mathcal{J}$, and hence the resolution of problems (3) and (7), is strictly equivalent to the minimization of the dual cost function $\mathcal{J}_D$. Numerically, the minimization of the dual cost function should be faster because the size of the observation space is usually $10^5 - 10^6$ whereas the state space has a dimension of $10^7 - 10^8$. The minimization of $\mathcal{J}_D$ is then performed on a space of much smaller dimension.

By minimizing the Lagrangian $\mathcal{L}$ with respect to (U,V), it is quite easy to obtain an explicit definition of $\mathcal{J}_D$:

$$\mathcal{J}_D(m) = \frac{1}{2}\langle(\mathcal{D}+R)m, m\rangle - \langle d, m\rangle \tag{14}$$

where d is the innovation vector: $d = X_{obs} - CX_{background}$. The matrix $\mathcal{D}$ is called the representers' matrix (Bennett, 1992; Bennett, 2002), and has the following definition:

$$\mathcal{D} = C\mathcal{M}B\mathcal{M}^T C^T + CQC^T \tag{15}$$

where $\mathcal{M}$ is still the direct model, and $\mathcal{M}^T$ represents the adjoint model. The matrix $\mathcal{D}$ quantifies the impact of each specific observation on the others. The minimization of $\mathcal{J}_D$ can be performed in the same way as the minimization of $\mathcal{J}$, using an iterative descent algorithm. In each iteration, one has to first compute the solution of the adjoint model, and then the solution of the direct model in order to evaluate the dual cost function and its gradient.

3.3. Nonlinear extended 4D-PSAS algorithm computation

When the model (and/or the observation operator) is non linear, it is possible to extend the previous duality results in an empirical way (Auroux et al., 2004). Let m be a vector of the observation space, we first have to solve an adjoint (backward) model:

$$\begin{cases} -\dfrac{dP}{dt} = \left(\dfrac{\partial F}{\partial X}\right)^T P - C^T R^{-1}(m - X_{obs}), \\ P(T) = 0, \end{cases} \tag{16}$$

and then the direct model, forced by the adjoint state:

$$\begin{cases} \dfrac{dX}{dt} = F(X, P), \\ X(0) = X_{background} + P(0), \end{cases} \tag{17}$$

where $X_{background}$ is an approximation of the initial condition and usually results from a previous prediction.

The extended 4D-PSAS algorithm computation is then performed in the following way: we first need an initial guess m_0 (which can be taken equal to X_{obs} for example). Then, given the first guess, we use as in the 4D-Var algorithm an iterative algorithm:

$$m_n = m_{n-1} + \rho_n D_n.$$

The knowledge of m_{n-1} allows us to compute the corresponding solution P_{n-1} of (16) and then the solution X_{n-1} of (17). It is then easy to evaluate $\mathcal{J}_D(m_{n-1})$ and its gradient, and given a descent-type algorithm, to define a descent direction D_n and the corresponding step size ρ_n.

Once the minimization of the dual cost function $\mathcal{J}_D$ is achieved, we immediately obtain the corresponding trajectory $X(t)$ in the state space, thanks to (17).

3.4. Advantages and concerns

One of the main concerns of this extended algorithm is the loss of equivalence with the 4D-Var algorithm when the model is not linear. Therefore, it is difficult to compare theoretically the two algorithms because of the empirical extension of 4D-PSAS to nonlinear problems.

Hopefully, the extended 4D-PSAS algorithm has numerous advantages. First of all, it inherently takes into account the unknown model parameters. The adjoint model provides an estimation of the model parameters with no additional computational cost. The size of the control vector m is then exactly the size of the observation space, whereas in the 4D-Var algorithm, the size of the control vector (U, V) is at least a few times the size of the state space (and in the worst case, with a non reduced order 4D-Var, the size of the state space multiplied by the number of time steps, which can be about 10^3).

Moreover, the computational cost of one 4D-PSAS iteration is almost the same as one 4D-Var iteration, but the minimization of the dual cost function takes place on a space of smaller dimension. The minimization is hence generally faster and needs a smaller number of iterations.

4. An improved sequential DA algorithm: the SEEK filter

In this section, we will study data assimilation methods based on the statistical estimation theory, in which the Kalman filtering theory is the primary framework. But the application of this theory encounters enormous difficulties due to the huge dimension of the state vector of the considered system. A further major difficulty is caused by its non linear nature. To deal with this, one usually linearizes the ordinary Kalman filter (KF) leading to the so-called extended Kalman filter (EKF) (Ghil et al., 1982; Evensen, 1992; Gauthier et al., 1993; Verron et al., 1999). One way to get rid of the issue of dimension is to use singular low rank error covariance matrices. The resulting filter, called the singular evolutive extended Kalman (SEEK) filter, not only solves the practical problem of reducing the computational cost to an acceptable level, but in addition reduces the propagation of error from one step to the next (Cane et al., 1996; Pham et al., 1998).

4.1. The extended Kalman filter

Consider a physical system described by

$$X(t_i) = \mathcal{M}(t_{i-1}, t_i)X(t_{i-1}) + U_i \tag{18}$$

where $\mathcal{M}(t_{i-1}, t_i)$ is an operator describing the system transition from time t_{i-1} to t_i, usually obtained from the integration of a partial differential system, and U_i is the unknown term of the model (it can be a noise term, used to modelize the unknown parameters of the model). We suppose that at each time t_i, we have an observation vector $X_{obs}(t_i)$. Let us denote by ε_i the observation error, i.e. the difference between the observation vector and the corresponding state vector:

$$\varepsilon_i = X_{obs}(t_i) - C_i X(t_i), \tag{19}$$

where C_i is the observation operator at time t_i, mapping the state space into the space of observations. Q_i and R_i will be the covariance matrices of the model error (U_i) and the observation error (ε_i) respectively.

The extended Kalman filter operates sequentially: from an analysis state vector $X_a(t_{i-1})$ and its error covariance matrix $P^a(t_{i-1})$, it constructs the next analysis state vector $X_a(t_i)$ and $P^a(t_i)$ in two steps, a forecasting step and a correction step.

The first step is used to forecast the state at time t_i:

$$X^f(t_i) = M(t_{i-1}, t_i) X^a(t_{i-1}), \tag{20}$$

where $M(t_{i-1}, t_i)$ is the linearized model around $X^a(t_{i-1})$. The forecast error covariance matrix is then approximately

$$P^f(t_i) = M(t_{i-1}, t_i) P^a(t_i) M(t_{i-1}, t_i)^T + Q_i. \tag{21}$$

The second step is an analysis step, the newly available observation $X_{obs}(t_i)$ is used to correct the forecast state vector $X^f(t_i)$ in order to define a new analysis vector:

$$X^a(t_i) = X^f(t_i) + K_i(X_{obs}(t_i) - C_i X^f(t_i)), \tag{22}$$

where K_i is a gain matrix, called the Kalman matrix. The optimal gain is given by

$$K_i = P^f(t_i) C_i^T \left(C_i P^f(t_i) C_i^T + R_i \right)^{-1}. \tag{23}$$

The corresponding analysis error covariance matrix is given by

$$P^a(t_i) = P^f(t_i) - P^f(t_i) C_i^T \left(C_i P^f(t_i) C_i^T + R_i \right)^{-1} C_i P^f(t_i). \tag{24}$$

One main issue of the EKF is that the covariance matrices R_i, Q_i and P_0^a have to be known. Some statistical information can be obtained for observation error from the knowledge of the instrumental error variances in situations such as altimetric observations from satellites over the ocean, for which the error estimates have become fairly solidly

established. But it is not clear how the correlations of these errors can be obtained. The covariances matrices Q_i and P_0^a are much more difficult to obtain, because very little is known concerning the true initial state of the system. These matrices are of very large dimension, and usually have a quite large number of independent elements. Is it really useful to estimate such a huge number of parameters? The theory for such equations (Eqs. (21) and (24)) state that for linear autonomous systems, even if P_0^a is poorly specified, one may hopefully still have a good approximation to P_i^a in the long term. The Kalman filter is optimal only if the covariance matrices R_i and Q_i are correctly specified. Thus, in practice, the Kalman filter is suboptimal.

4.2. The SEEK (Singular Evolutive Extended Kalman) filter

It seems that a relatively optimal Kalman filter is quite ambitious. Therefore, one can only impose that the filtering error should remain bounded. The propragation of the filter error is given by

$$\begin{aligned} X^a(t_i) - X^t(t_i) &= (I - K_i C_i)\, M(t_{i-1}, t_i) \left(X^a(t_{i-1} - X^t(t_{i-1}) \right) \\ &- K_i \varepsilon_i - (I - K_i C_i) U_i. \end{aligned} \tag{25}$$

This clearly shows that the stability of the filter depends essentially on the matrices $(I - K_i C_i) M(t_{i-1}, t_i)$. Therefore, it is necessary that all eigenvalues of these matrices have modulus smaller than 1.

For computational reasons, it is also crucial to use low rank error covariance matrices P_a^i. Hence, the initialization of the SEEK filter is performed with matrices of the form LUL^T: one may first choose an initial analysis state $X^a(t_0)$, and a low rank error covariance matrix

$$P_0^a = L_0 U_0 L_0^T$$

where L_0 is a column vector and U_0 is a positive definite matrix with dimension equal to the rank of P_0^a, this being low in practical applications.

The forecasting step is then given by

$$X^f(t_i) = M(t_{i-1}, t_i) X^a(t_{i-1}) \tag{26}$$

and

$$L_i = M(t_{i-1}, t_i) L_{i-1}. \tag{27}$$

The correction step is the following: compute U_i by the following way

$$U_i^{-1} = \left(U_{i-1} + (L_i^T L_i)^{-1} L_i^T Q_i L_i (L_i^T L_i)^{-1} \right)^{-1} + L_i^T C_i^T R_i^{-1} C_i L_i \tag{28}$$

and then compute the new analysis vector

$$X^a(t_i) = X^f(t_i) + (L_i U_i L_i^T) C_i^T R_i^{-1} \left(X_{obs}(t_i) - C_i X^f(t_i) \right). \quad (29)$$

Finally, there is an additional step, the renormalization. One can change L_i to NL_i and U_i to $(N^T)^{-1} U_i N^{-1}$ without changing the algorithm. This should be done periodically to avoid the column of L_i from becoming large and nearly parallel each to the others, and U_i becoming ill conditioned. One usually takes N to be the Cholesky factor of U_i^{-1}, so as to change U_i to the identity matrix.

From these equations, one sees that corrections are made parallel to the space spanned by the columns of L_i. Moreover, it is possible to prove that this filter is stable.

4.3. Initialization of the SEEK filter

The initialization of the filter is one of the largest issues of this algorithm. To initialize the SEEK filter (but also any other Kalman filter), one needs an initial analysis state vector $X^a(t_0)$ and its error covariance matrix P_0^a. The most frequent way to choose them is the EOFs (Empirical Orthogonal Functions) technique. The initial state may be set arbitrarily if one has taken care to wait until the model has been settled into a stable regime. It is quite easy to generate long sequences of state vectors from the model equation (18). Then, it is possible to take as $X^a(t_0)$ the average of the simulated state vectors, and as P_0^a the low rank approximation of the sample covariance matrix P_0 of these vectors. The EOFs technique provides such an estimation. Let V_i be the eigenvectors of P_0, ordered according to their eigenvalues λ_i (where λ_1 is the largest). One has then to choose the rank r of the covariance matrix approximation, and then set

$$L_0 = [V_1, \ldots, V_r]$$

and

$$U_0 = diag(\lambda_1, \ldots, \lambda_r).$$

The ratio

$$\frac{\sum_{j>r} \lambda_j}{Tr(P_0)}$$

represents the relative error and can be used to assess the accuracy of the approximation for choosing the appropriate value of r (Pham et al., 1998).

One also needs to specify the matrices R_i and Q_i in order to apply the SEEK filter. These matrices are generally unknown, and R_i can

be taken as σ^2 times a constant matrix, often an identity matrix (for computational reasons). Then, using such matrices R_i in the SEEK equations, one can easily see that only $\frac{U_i}{\sigma^2}$ has to be known, and hence, using equation (28), it is enough to specify $\frac{U_0}{\sigma^2}$. Consequently, if U_0 is carefully choosen, it is no more necessary to know σ^2. Usually, σ^2 is very small with respect to U_0, and then, it is safe to take U_0 very large for stability reasons.

5. A new simple DA algorithm: the Back and Forth Nudging

The main issues of data assimilation for geophysical systems are the huge dimension of the control vectors (and hence of the covariance matrices) and the non linearities (most of the time, one has to linearize the model and/or some operators). The computation of the adjoint model is for example a difficult step in the variational algorithms. To get rid of these difficulties, we have very recently introduced a new algorithm, based on the nudging technique.

5.1. The nudging algorithm

The standard nudging algorithm consists in adding to the state equations a feedback term, which is proportional to the difference between the observation and its equivalent quantity computed by the resolution of the state equations. The model appears then as a weak constraint, and the nudging term forces the state variables to fit as well as possible to the observations.

Let us remind the model

$$\begin{cases} \dfrac{dX}{dt} = F(X,U), \quad 0 < t < T, \\ X(0) = V. \end{cases} \tag{30}$$

We still suppose that we have an observation $X_{obs}(t)$ of the state variable $X(t)$. The nudging algorithm simply gives

$$\begin{cases} \dfrac{dX}{dt} = F(X,U) + K(X_{obs} - CX), \quad 0 < t < T, \\ X(0) = V, \end{cases} \tag{31}$$

where C is still the observation operator, and K is the nudging matrix. It is quite easy to see that if K is large enough, then the state

vector transposed into the observation space (through the observation operator) $CX(t)$ will tend towards the observation vector $X_{obs}(t)$.

This algorithm was first used in meteorology (Hoke et al., 1976), and then has been used with success in oceanography (Verron et al., 1989) and applied to a mesoscale model of the atmosphere (Stauffer et al., 1990). Many results have also been carried out on the optimal determination of the nudging coefficients K (Zou et al., 1992; Stauffer et al., 1993; Vidard et al., 2003).

The nudging algorithm is usually considered as a sequential data assimilation method. If one solves equation (31) with a numerical scheme, then it is equivalent with the following algorithm:

$$\begin{cases} X_n^f = X_{n-1} + dt \times F(X_{n-1}, U), \\ X_n = X_n^f + K_n(X_{obs}(t_n) - C_n X_n^f), \end{cases} \tag{32}$$

which is exactly the Kalman filter's algorithm. Then, if at any time the nudging matrix K is set in an optimal way, it is quite easy to see that K will be exactly the Kalman gain matrix.

5.2. Backward nudging

The backward nudging algorithm consists in solving the state equations of the model backwards in time, starting from the observation of the state of the system at the final instant. A nudging term, with the opposite sign compared to the standard nudging algorithm, is added to the state equations, and the final obtained state is in fact the initial state of the system (Auroux, 2003; Auroux et al., 2005).

We now assume that we have a final condition in (30) instead of an initial condition. This leads to the following backward equation

$$\begin{cases} \dfrac{d\tilde{X}}{dt} = F(\tilde{X}, U), \quad T > t > 0, \\ \tilde{X}(T) = \tilde{V}. \end{cases} \tag{33}$$

If we apply nudging to this backward model with the opposite sign of the feedback term (in order to have a well posed problem), we obtain

$$\begin{cases} \dfrac{d\tilde{X}}{dt} = F(\tilde{X}, U) - K(X_{obs} - C\tilde{X}), \quad T > t > 0, \\ \tilde{X}(T) = \tilde{V}. \end{cases} \tag{34}$$

Once again, it is easy to see that if K is large enough, the state vector $X(t)$ will tend (through the observation operator) towards the observation vector $X_{obs}(t)$.

5.3. The BFN algorithm

The back and forth nudging (BFN) algorithm consists in solving first the forward (standard) nudging equation, and then the direct system backwards in time with a feedback term. After resolution of this backward equation, one obtains an estimate of the initial state of the system. We repeat these forward and backward resolutions with the feedback terms until convergence of the algorithm (Auroux et al., 2005).

The BFN algorithm is then the following:

$$\begin{cases} \dfrac{dX_k}{dt} = F(X_k, U) + K(X_{obs} - CX_k), \\ X_k(0) = \tilde{X}_{k-1}(0), \end{cases}$$
$$\begin{cases} \dfrac{d\tilde{X}_k}{dt} = F(\tilde{X}_k, U) - K(X_{obs} - C\tilde{X}_k), \\ \tilde{X}_k(T) = X_k(T), \end{cases} \tag{35}$$

with $\tilde{X}_{-1}(0) = V$. Then, $X_(0) = V$, and a resolution of the direct model gives $X_0(T)$ and hence $\tilde{X}_0(T)$. A resolution of the backward model provides $\tilde{X}_0(0)$, which is equal to $X_1(0)$, and so on.

This algorithm can be compared to the 4D-Var algorithm, which also consists in a sequence of forward and backward resolutions. In the BFN algorithm, even for nonlinear problems, it is useless to linearize the system and the backward system is not the adjoint equation but the direct system, with an extra feedback term that stabilizes the resolution of this ill-posed backward resolution.

It can also be compared to the 4D-PSAS algorithm, because it provides automatically a corrective term in the model equations, which can play the role of the unknown parameters of the model. It is then nearly useless to consider U as a part of the control vector, and one simply sets U as a constant variable (either 0 or some estimation of the model parameters) and looks only for V.

5.4. Theoretical results

Some theoretical results can be proved for a linear model. If F is a linear operator, i.e. $F(X, U)$ can be written FX (as previously said, there is no more need to take into account the model parameters U), it is straight forward to prove the following result (Auroux et al., 2005).

If $n \to +\infty$, we have convergence of $X_n(0)$ and if KC is a definite positive matrix,

$$\lim_{n \to +\infty} X_n(0) = X_\infty(0) \tag{36}$$

$$= \left(I - e^{-2KCT}\right)^{-1} \int_0^T \left(e^{-(KC+A)s} + e^{-2KCT} e^{(KC-A)s}\right) K X_{obs}(s) ds.$$

Moreover, it $T > 0$, for any $t \in [0, T]$,

$$\lim_{n \to +\infty} X_n(t) = X_\infty(t) \tag{37}$$
$$= e^{-(KC-A)t} \int_0^T e^{(KC-A)s} K X_{obs}(s) ds + e^{-(KC-A)t} X_\infty(0).$$

Under the same hypothesis, we have a similar result for backward trajectories, e.g. there exists a function $\tilde{X}_\infty(t)$ so that $\lim_{n \to +\infty} \tilde{X}_n(t) = \tilde{X}_\infty(t)$, for any $t \in [0, T]$. This proves the convergence of the BFN algorithm.

5.5. Numerical results

We consider in this subsection a very simple nonlinear geophysical model, often used as a simplified model for turbulence, shock wave formation, and mass transport. The evolution model is the viscous Burgers' equation over a cyclic one-dimensional domain:

$$\frac{\partial X}{\partial t} + \frac{1}{2} \frac{\partial X^2}{\partial s} - \nu \frac{\partial^2 X}{\partial s^2} = 0, \tag{38}$$

where X is the state variable, s represents the distance in meters around the 45^oN constant-latitude circle and t is the time. The period of the domain is roughly $28.3 \times 10^6 m$. The diffusion coefficient ν is set to $10^5\ m^2.s^{-1}$ (Fisher et al., 1995). The assimilation period is roughly one month and the time step is one hour.

The experimental approach consists in performing twin experiments with simulated data. First, a reference experiment is run and the corresponding data are extracted. This reference trajectory will be further called the exact solution. Experimental data are supposed to be obtained on every fifth gridpoint of the model, with a time sampling of 10 hours (every 10 time steps). This provides a spatial density nearly similar to the longitudinal distribution of the mid-latitude radiosonde network. Simulated data are then noised with a blank Gaussian distribution, with a 5% root mean square (RMS) error. The first guess of the assimilation experiments is chosen as a constant field ($X = 0$ everywhere). The results of the identification process are then compared to the reference experiment.

We first focus our interest on the numerical convergence of the BFN algorithm we have proposed because, currently, the mathematical convergence is only valid for a linear model.

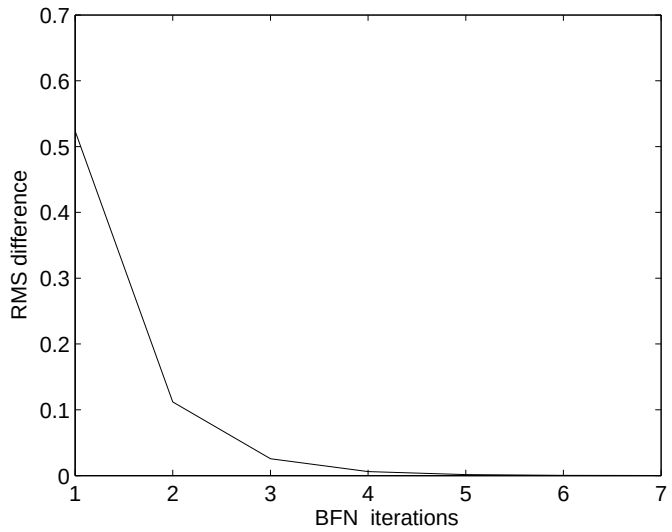


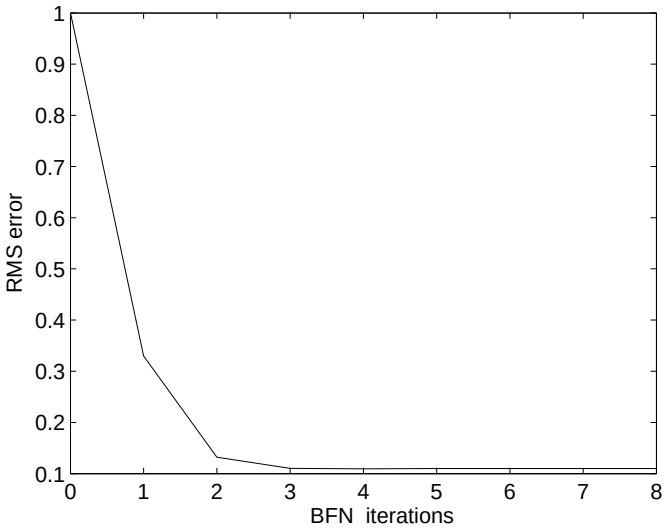
Figure 1. RMS relative difference between two consecutive iterates of the BFN algorithm versus the number of iterations.

Figure 1 shows the RMS relative difference between two iterates of the BFN algorithm $\frac{\|X_{k+1} - X_k\|}{\|X_k\|}$ (see (35) for the notations) versus the number of iterations. We can clearly see that in less than 5 iterations, the difference between two iterates is smaller than 1%. The numerical convergence of the algorithm is then obvious, and very quickly achieved.

We have then compared the BFN iterates with the exact solution (or reference trajectory) with the aim of quantifying the identification of the true initial state.

Figure 2-a shows the RMS relative difference between the BFN iterates at time $t = 0$ and the exact initial condition $\frac{\|X_k(0) - V_{true}\|}{\|V_{true}\|}$ versus the number of iterations. We still observe the convergence in less than 5 iterations. The identification error is nearly 12% at the end of the process. This seems huge, but compared to the other DA techniques, the BFN algorithm is not supposed to identify precisely the initial condition but the reference trajectory as a whole. Figure 2-b shows indeed the RMS difference between the BFN iterates and the exact final condition (i.e. the reference trajectory at the end of the assimilation period) versus the number of iterations. We can see that the difference between the true final solution and the identified final solution is about 5%.

(a)



(b)

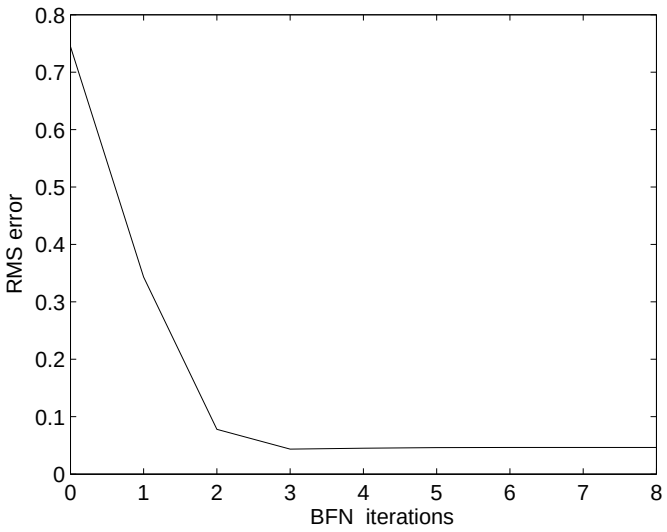


Figure 2. RMS relative difference between the BFN iterates and the exact solution versus the number of iterations, at time $t = 0$ (a) and at time $t = T$ (b).

We finally focus our interest on the forecast of the system evolution after the assimilation period. This is the most frequent application of data assimilation. We have consider on one hand the final solution (at time $t = T$) provided by the BFN algorithm and on the other hand an interpolation in the state space of the last available observation (still at time $t = T$). We use these two states as two initial conditions for the

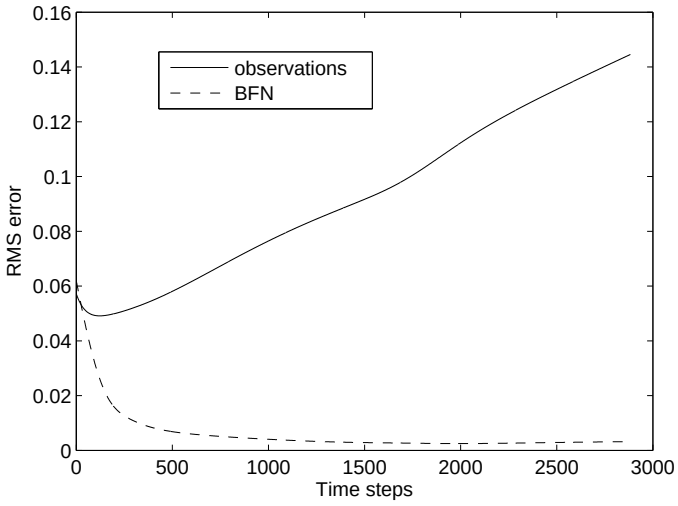


Figure 3. RMS relative difference between the reference trajectory and the prediction trajectories provided by the interpolated observations (plain line) and by the BFN algorithm (dashed line), versus the time steps.

exact model (38) and we compute the two corresponding trajectories on a 3 months time period after the end of the assimilation period. It corresponds to a 3 months prediction.

Figure 3 shows the RMS relative difference between the BFN trajectory (computed with the final BFN state as an initial condition) and the reference trajectory, and between the observation trajectory (computed with the interpolation of the last available observation as an initial condition) and the reference trajectory, versus the time steps. We can see that the observation trajectory has an error of about 6% at the end of the assimilation period (or at the beginning of the prediction period). Then, the stable modes of the model make the error decrease, and then, after a few days (200 time steps, nearly 6 days), the unstable modes make it increase all the time. If we look at the BFN trajectory, we can see that at the beginning of the prediction period, the error is nearly the same as for the observation trajectory, nearly 6%, but after one week, the error is smaller than 1%, even after 4 months, the error is still much smaller than 1% whereas the error of the observation trajectory has nearly reached 15%. This clearly proves the usefulness of the BFN algorithm, which makes it possible to reconstruct a trajectory over a 4 months period with an assimilation period of only one month,

with less than 1% RMS error using noised observations with a 5% RMS error.

6. Conclusion

Geophysical flows are governed by complex equations and present certain typical characteristics of turbulent flows. Besides, in practice, the observation data are of various natures and should be combined together into a single data assimilation system. Presently, data assimilation is a very active domain of research with extensions toward several directions.

We have presented in this paper a small overview of data assimilation, in introducing the most common variational and sequential algorithms: 4D-VAR, 4D-PSAS, SEEK filter. We also introduced a new DA method, the BFN algorithm. This algorithm is hence very promising to obtain an estimation of the initial state of a system and especially some good predictions of its evolution, with a very easy implementation because it requires neither the linearization of the equations, nor any minimization process. It also does not require the use of large covariance matrices.

Data assimilation has become an essential tool for modelling and prediction of the evolution of geophysical fluids. In many other domains for which data and models are the main sources of information, these methods could be developed in the near future.

References

Amodei, L. Solution approchée pour un problème d'assimilation de données météorologiques avec prise en compte de l'erreur modèle. *C.R. Acad. Sci. Paris*, 321(II):1087–1094, 1995.

Auroux, D. Étude de différentes méthodes d'assimilation de données pour l'environnement. *Ph.D. Thesis*, University of Nice Sophia-Antipolis, France, 2003.

Auroux, D. and J. Blum. Data assimilation methods for an oceanographic problem. In: *Multidisciplinary methods for analysis, optimization and control of complex systems*, Mathematics in Industry Series, Springer, 2004.

Auroux, D. and J. Blum. Back and forth nudging algorithm for data assimilation problems. *C. R. Acad. Sci. Paris, Ser. I*, 340:873–878, 2005.

Bennett, A. F. Inverse methods in physical oceanography. Cambridge University Press, Cambridge, 1992.

Bennett, A. F. Inverse Modeling of the Ocean and Atmosphere. Cambridge University Press, Cambridge, 2002.

Broyden, C. G. A new double-rank minimization algorithm. *Notices American Math. Soc.*, 16:670, 1969.

Cane, M. A., A. Kaplan, R. N. Miller, B. Tang, E. C. Hackert, and A. J. Busalacchi. Mapping tropical Pacific sea level: data assimilation via a reduced state Kalman filter. *J. Geophys. Res.*, 101(C10):22599–22617, 1996.

Courtier, P. and O. Talagrand. Variational assimilation of meteorological observations with the adjoint equations Part 2. Numerical results. *Quart. J. Roy. Meteor. Soc.*, 113:1329–1347, 1987.

Courtier, P. Dual formulation of four-dimensional variational assimilation. *Q. J. R. Meteorol. Soc.*, 123:2449–2461, 1997.

Durbiano, S. Vecteurs caractéristiques de modèles océaniques pour la réduction d'ordre en assimilation de données *Ph.D. Thesis*, University of Grenoble I, France, 2001.

Evensen, G. Using the extended Kalman filter with a multilayer quasi-geostrophic ocean model. *J. Geophys. Res.*, 97:17905–17924, 1992.

Fisher, M. and P. Courtier. Estimating the covariance matrix of analysis and forecast error in variational data assimilation. ECMWF Technical Memorandum, 220, 1995.

Fukumori, I., J. Benveniste, C. Wunsch, and D. B. Haidvogel. Assimilation of sea surface topography into an ocean circulation model using a steady state smoother. *J. Phys. Oceanogr.*, 23:1831–1855, 1993.

Fukumori, I. Assimilation of Topex sea level measurements with a reduced-gravity, shallow water model of the tropical Pacific ocean. *J. Geophys. Res.*, 100(C12):25027–25039, 1995.

Gauthier, P., P. Courtier, and P. Moll. Assimilation of simulated wind lidar data with a Kalman filter. *Mon. Wea. Rev.*, 121:1803–1820, 1993.

Gelb, A. Applied Optimal Estimation. MA: MIT Press, Cambridge, 1974.

Ghil, M., S. E. Cohn, and A. Dalcher. Sequential estimation, data assimilation and initialization. In *Williamson Ed., The interaction between objective analysis and initialization*, Publ. Meteor. 127, McGill University, Montréal, 1982.

Ghil, M. Meteorological data assimilation for oceanographers. Part I: description and theoretical framework. *Dyn. Atmos. Oceans*, 13:171–218, 1989.

Ghil, M. and P. Manalotte-Rizzoli. Data assimilation in meteorology and oceanography. *Adv. Geophys.*, 23:141–265, 1991.

Gilbert, J.-Ch. and C. Lemaréchal. Some numerical experiments with variable storage quasi-Newton algorithms. *Math. Prog.*, 45:407–435, 1989.

Gourdeau, L., S. Arnault, Y. Ménard, and J. Merle. Geosat sea-level assimilation in a tropical Atlantic model using Kalman filter. *Ocean. Acta*, 15:567–574, 1992.

Holland, W. R. The role of mesoscale eddies in the general circulation of the ocean. *J. Phys. Ocean.*, 8(3):363–392, 1978.

Hoke, J. and R. A. Anthes. The initialization of numerical models by a dynamic initialization technique. *Month. Weaver Rev.*, 104:1551–1556, 1976.

Jazwinski, A. H. Stochastic Processes and Filtering Theory. Academic, New York, 1970.

Kalnay, E. Atmospheric modeling, data assimilation and predictability. Camdrige Univ. Press, 2003.

Le Dimet, F.-X. and O. Talagrand. Variational algorithms for analysis and assimilation of meteorogical observations: theoretical aspects. *Tellus*, 38A:97–110, 1986.

Lewis, J. M. and J. C. Derber. The use of adjoint equations to solve a variational adjustment problem with convective constraints. *Tellus*, 37A:309–322, 1985.

Lions, J. L. Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles. Dunod, 1968.

Liu, D. C. and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Prog.*, 45:503–528, 1989.

Luong, B., J. Blum, and J. Verron. A variational method for the resolution of a data assimilation problem in oceanography. *Inverse Problems*, 14:979–997, 1998.

Mohammadi, B. and O. Pirroneau. Applied shape optimization for fluids. Clarendon Press, Oxford, 2001.

Moore, A.M. Data assimilation in a quasigeostrophic open-ocean model of the Gulf-Stream region using the adjoint model. *J. Phys. Oceanogr.*, 21:398–427, 1991.

Nechaev, V. and M. I. Yaremchuk. Application of the adjoint technique to processing of a standard section data set: world ocean circulation experiment section S4 along 67S in the Pacific ocean. *J. Geophys. Res.*, 100(C1):865–879, 1994.

Pedlosky, J. Geophysical fluid dynamics. Springer-Verlag, New-York, 1979.

Pham, D. T., J. Verron, and M. C. Roubaud. A singular Evolutive Extended Kalman filter for data assimilation in oceanography. *Inverse Problems*, 14:979–997, 1998.

Rostaing-Schmidt, N. and E. Hassold. Basic function representation of programs for automatic differentiation in the Odyssée system. In *F-X Le Dimet editor, High performance computing in the geosciences*, pp. 207–222, Kluwer Academic Publishers, 1994.

Schröter, J., U. Seiler, and M. Wenzel. Variational assimilation of Geosat data into an eddy-resolving model of the Gulf Stream area. *J. Phys. Oceanogr.*, 23:925–953, 1993.

Sheinbaum, J. and D. L. T. Anderson. Variational assimilation of XBT data. Part I. *J. Phys. Oceanogr.*, 20:672–688, 1990.

Stauffer, D. R. and N. L. Seaman. Use of four dimensional data assimilation in a limited area mesoscale model - Part 1: Experiments with synoptic-scale data. *Month. Weather Rev.*, 118:1250–1277, 1990.

Stauffer, D. R. and J. W. Bao. Optimal determination of nudging coefficients using the adjoint equations. *Tellus A*, 45:358–369, 1993.

Talagrand, O. and P. Courtier. Variational assimilation of meteorological observations with the adjoint vorticity equation. Part I: Theory. *Q. J. R. Meteorol. Soc.*, 113:1311–1328, 1987.

Talagrand, O. Assimilation of observations, an introduction. *Journal of the Met. Soc. of Japan*, 75(1B):191–209, 1997.

Thacker, W. C. and R. B. Long. Fitting dynamics to data. *J. Geophys. Res.*, 93:1227–1240, 1988.

Veersé, F., D. Auroux, and M. Fisher. Limited-memory BFGS diagonal preconditioners for a data assimilation problem in meteorology. *Optimization and Engineering*, 1(3):323–339, 2000.

Verron, J. and W. R. Holland. Impact de données d'altimétrie satellitaire sur les simulations numériques des circulations générales océaniques aux latitudes moyennes. *Ann. Geophys.*, 7(1):31–46, 1989.

Verron, J., L. Gourdeau, D. T. Pham, R. Murtugudde, and A. J. Busalacchi. An extended Kalman filter to assimilate satellite altimeter data into a non-linear numerical model of the tropical Pacific Ocean: method and validation. *J. Geophys. Res.*, 104:5441–5458, 1999.

Vidard, P. A. Vers une prise en compte des erreurs modèle en assimilation de données 4D-variationnelle - Application à un modèle réaliste d'océan. *Ph.D. Thesis*, University of Grenoble I, France, 2001.

Vidard, A., F.-X. Le Dimet, and A. Piacentini. Determination of optimal nudging coefficients. *Tellus A*, 55:1–15, 2003.

Zou, X., I. M. Navon, and F.-X. Le Dimet. An optimal nudging data assimilation scheme using parameter estimation. *Quart. J. Roy. Meteorol. Soc.*, 118:1163–1186, 1002.