ANNALES
DE LA FACULTÉ
DES SCIENCES
DE
TOULOUSE
Mathématiques

DIDIER AUROUX

Data assimilation for geophysical fluids

Tome XXVI, n° 4 (2017), p. 767-793.

<http://afst.cedram.org/item?id=AFST_2017_6_26_4_767_0>

Annales de la faculté des sciences de Toulouse Volume XXVI, n° 4, 2017
pp. 767-793

Data assimilation for geophysical fluids

Didier Auroux [1]

Abstract. — Data assimilation is the domain at the interface between observations and models, which makes it possible to identify the global structure of a geophysical system from a set of discrete space-time data. After recalling state-of-the-art data assimilation methods, the variational 4D-VAR algorithm and the dual variational 4D-PSAS algorithm, and sequential Kalman filters, we will present the Back and Forth Nudging (BFN) algorithm, and the Diffusive Back and Forth Nudging (DBFN) algorithm, which is a natural extension of the BFN to some particular diffusive models.

Résumé. — L'assimilation de données est l'ensemble des techniques qui permettent de combiner un modèle et des observations. Le but est ici d'identifier l'état d'un système géophysique à partir de données discrètes en temps et en espace. Après un rappel de l'état de l'art en assimilation de données (méthode variationnelle 4D-VAR et approche duale 4D-PSAS, filtres séquentiels de type Kalman), nous présentons l'algorithme du nudging direct et rétrograde, ainsi que son extension naturelle (le nudging direct et rétrograde diffusif) à certains modèles géophysiques contenant un terme de diffusion.

1. Introduction

It is 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 nonlinear because of their fluid component. Such nonlinearities 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

[1] Université Côte d'Azur, Inria, CNRS, LJAD, France — auroux@unice.fr

improvement of the models used to simulate the geophysical processes. For example, a major problem comes from the fact that sub-scaled 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 with sub-grid processes by some additional terms in the equations [38, 55].

Over the past twenty years, observations of ocean and atmosphere circulation have become much more readily available [14], as a result of new satellite techniques and international field programs (MERCATOR, CLIPPER, GODAE, ARGO, ...). 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 [63]: variational methods based on the optimal control theory [45] and statistical methods based on the theory of optimal statistical estimation (see, for example, [13, 14, 41] 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 [42, 44, 64] and more recently for oceanic data [50, 53, 54, 59, 60, 65]. The statistical (or sequential) methods (optimal interpolation, Kalman filter, SEEK filter, ...) were introduced in oceanography roughly fifteen years ago [31, 33]. The Kalman filter was extended to nonlinear cases [30, 40] but it has been mostly applied in oceanography to quasi-linear situations, in particular tropical oceans [17, 27, 28, 35, 67].

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. We present then a generalization to nonlinear models of the four dimensional variational dual method, the 4D-PSAS algorithm. The idea of 4D-PSAS (Physical Space Analysis System) is to perform the minimization in the space of the observations, rather than in the model space as in the primal 4D-VAR scheme. Despite the formal equivalence between 4D-VAR and 4D-PSAS in a linear situation (both for model equations and observation operators), the dual method has several important advantages: in oceanographic cases, the observation space is smaller than the model space, which should improve the minimization process; for no additional cost, it provides an estimation of the model error; and finally, it does not have any singularities when the covariance error matrices tend to zero.

Sequential methods are mostly based on the Kalman filtering theory, which consists in a forecast step and an analysis (or correction) step. In Section 3, we present the extended Kalman filter (EKF), for nonlinear models. A main drawback of the (extended) Kalman filter is the computational cost of propagating in time the error covariance matrices, and we present then the SEEK (Singular Evolutive Extended Kalman) filter, in which the dimension

of the problem is reduced. Finally, we present the ensemble Kalman filter (EnKF), for which an ensemble of states (members) is used to compute actual covariance matrices at a lower computational cost.

We recall in Section 4 the standard nudging algorithm, and then the Back and Forth Nudging (BFN) algorithm, which is the 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. We also present the Diffusive Back and Forth Nudging (DBFN) algorithm, which is a natural extension of the BFN to some particular diffusive models. This section ends with theoretical considerations on both BFN and DBFN algorithms.

Finally, some concluding remarks and perspectives are given in Section 5.

2. Variational methods

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. The state-of-the-art variational method is the 4D-VAR, 4D meaning that it can assimilate observations in space (3D) and time (1D). The 4D-VAR usually assumes that the model is a strong constraint: at each iteration, the computed trajectory is a solution of the model equations. However, it is possible to take into account a model error, but the size of the control vector usually becomes too large for operational computations. We will see that the computational cost can be too expensive, and we will present a reduced order 4D-VAR, in which the dimension of the problem is drastically reduced. Then, we will see the dual method, called 4D-PSAS, in which the minimization is performed in the dual space.

2.1. Model and observations

Every DA method needs both a model describing the evolution of the fluid, basically a system of nonlinear partial differential equations (PDE),

and a set of discrete observations. Firstly, we assume that the model can be written, after discretization in space of the set of PDE:

$$\begin{cases} \dfrac{\mathrm{d}X}{\mathrm{d}t} = F(X,U), \quad 0 < t < T, \\ X(0) = V, \end{cases} \tag{2.1}$$

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 nonlinear differential operator, describing the dynamics of the system. U generally represents a model error (unknown terms in the model, due to incomplete modelling of some specific physical phenomena), but it can also model some internal variables of the model (parameters or boundary conditions) and it 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 nonlinear in some cases.

2.2. Cost function of the 4D-VAR

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

$$\mathcal{J}(U,V) = \frac{1}{2}\int_0^T \langle R^{-1}(CX - X_{obs}), CX - X_{obs}\rangle \,\mathrm{d}t + \frac{1}{2}\langle P_0^{-1}V, V\rangle + \frac{1}{2}\int_0^T \langle Q^{-1}U, U\rangle \,\mathrm{d}t \tag{2.2}$$

where X is the solution of (2.1). 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 system state, and the two others act like a regularization term in the sense of Tikhonov. 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 [42].

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:

$$\begin{cases} \text{Find } (U^*, V^*) \text{ such that} \\ \mathcal{J}(U^*, V^*) = \inf\limits_{(U,V)} \mathcal{J}(U, V). \end{cases} \tag{2.3}$$

2.3. Adjoint state and optimality system

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 [20, 42]. 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 discretized partial differential equations, known as the tangent linear model:

$$\begin{cases} \dfrac{\mathrm{d}\hat{X}}{\mathrm{d}t} = \dfrac{\partial F}{\partial X}\hat{X} + \dfrac{\partial F}{\partial U}u, \\ \hat{X}(0) = v, \end{cases} \tag{2.4}$$

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

If we assume that the operator C is linear (otherwise, we have to linearize it), the derivative of $\mathcal{J}$ with respect to (U, V) in the direction (u, v) is then

$$\langle \hat{\mathcal{J}}(U,V), (u,v) \rangle = \int_0^T \langle R^{-1}(CX - X_{obs}), C\hat{X} \rangle \, \mathrm{d}t + \langle P_0^{-1}V, v \rangle + \int_0^T \langle Q^{-1}U, u \rangle \, \mathrm{d}t.$$

We can introduce the so-called adjoint state P (which lives in the same space as X), solution of the adjoint model [42]:

$$\begin{cases} -\dfrac{\mathrm{d}P}{\mathrm{d}t} = \left(\dfrac{\partial F}{\partial X}\right)^T P - C^T R^{-1}(CX - X_{obs}), \\ P(T) = 0. \end{cases} \tag{2.5}$$

We have then:

$$\langle \hat{\mathcal{J}}(U,V), (u,v)\rangle = \int_0^T \langle \frac{\mathrm{d}P}{\mathrm{d}t} + \left(\frac{\partial F}{\partial X}\right)^T P, \hat{X}\rangle \,\mathrm{d}t + \langle P_0^{-1}V, v\rangle + \int_0^T \langle Q^{-1}U, u\rangle \,\mathrm{d}t$$

and an integration by part shows that, using (2.4):

$$\langle \hat{\mathcal{J}}(U,V), (u,v)\rangle = \int_0^T \langle -P, \frac{\partial F}{\partial U}u\rangle \,\mathrm{d}t - \langle P(0), v\rangle + \langle P_0^{-1}V, v\rangle + \int_0^T \langle Q^{-1}U, u\rangle \,\mathrm{d}t.$$

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

$$\nabla \mathcal{J}(U,V) = \begin{pmatrix} -\left(\dfrac{\partial F}{\partial U}\right)^T P + Q^{-1}U \\ -P(0) + P_0^{-1}V \end{pmatrix}. \tag{2.6}$$

Therefore, the gradient is obtained by a backward integration of the adjoint model (2.5), which has the same computational cost as one evaluation of $\mathcal{J}$.

The minimization problem (2.3) is then equivalent to the following optimality system:

$$\begin{cases} \dfrac{\mathrm{d}X}{\mathrm{d}t} = F(X, U^*), \\ X(0) = V^*, \\ -\dfrac{\mathrm{d}P}{\mathrm{d}t} = \left(\dfrac{\partial F}{\partial X}\right)^T P - C^T R^{-1}(CX - X_{obs}), \\ P(T) = 0, \\ \left(\dfrac{\partial F}{\partial U}\right)^T P = Q^{-1}U^*, \\ P(0) = P_0^{-1}V^*. \end{cases} \tag{2.7}$$

2.4. 4D-VAR algorithm computation

The determination of (U^*, V^*), solution of (2.3) and (2.7), 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 [34]:

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

where D_n is a 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 (2.1), and consequently to evaluate the cost function $\mathcal{J}(U_{n-1}, V_{n-1})$. Then we solve the adjoint model (2.5) and compute the adjoint solution P_{n-1}, and using (2.6), 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 chosen 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 nonlinear, we can get a high computational cost because it will require several evaluations of $\mathcal{J}$, and hence several integrations of the model (2.1) [16, 34, 46, 66].

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) [36, 51, 58].

Another issue is the relative ill-posedness of the problem when the model is nonlinear. The cost function $\mathcal{J}$ is hence non 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 (2.2), which correspond to two regularization terms with respect to the two control variables. This has to be done carefully because it can generate 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.

2.5. Reduced-order 4D-VAR

If in (2.1) 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:

$$X^* = X_{background} + \sum_{i=1}^{r} \lambda_i L_i, \tag{2.8}$$

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 model 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 [22].

The same idea is used for the model parameters:

$$U = \bar{U} + \sum_{i=1}^{s} \alpha_i u_i, \tag{2.9}$$

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 [22, 69].

2.6. Duality and 4D-PSAS algorithm

The primal method has many disadvantages. First, the minimization process is often stopped before convergence to the minimum, because of the size of the state vector. Both from a theoretical and numerical point of view, as the dimension of the control vector can reach several millions, the minimization algorithm would need at least thousands of iterations to converge. It is then necessary (for computational cost reasons) to stop the algorithm after a fixed (usually a few tens) and small (particularly within an operational point of view) number of iterations. Moreover, it is also impossible to take into account a model error: in the previous section, we have supposed that the model and the equations were perfect. This is obviously not the case (for example, not all parameters are well known). The only solution to incorporate the model error into the minimization process is to add corrective terms to the model, consider them as part of the control vector, and add a third term to the cost function. This is not computationally realistic because the size of the control vector would be multiplied by the number of time steps. Therefore, it is not possible to take into account in a straightforward way the model error in the primal variational approach.

A new approach to data assimilation problems has been proposed in the early 90's [1, 12, 19] in order to overcome these limitations. Rather than minimizing a cost function on the state space, the dual method works in the observation space (which is most of the time much smaller than the state space). This algorithm is called the 4D-PSAS: Physical Space Analysis System.

This method has been numerically studied in linear (or linearized) situations (see e.g. [47, 48] for its implementation in the oceanic primitive equation model MICOM), and then extended to nonlinear cases [3].

Instead of solving first the direct equations and then the adjoint equations as in the primal variational approach, the dual method consists in solving first the adjoint equations in order to use the information contained in the observation vector, and then the direct equations in order to reconstruct a trajectory. The main issue with the dual method is the nonlinearity of the model, which makes it necessary to update the reference trajectory used to linearize the equations and the observation operators for each iteration in the adjoint state computation.

Contrarily to the previous 4D-VAR formulation, we now consider the model as a weak constraint. It is then possible to introduce a Lagrange multiplier for this constraint [1, 12, 13, 19].

2.7. Weak model constraint formulation

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

$$\mathcal{M}(U,V) = X \tag{2.10}$$

where X is the solution of (2.1). We would like that CX fits with the observations X_{obs}. Let us introduce a new variable Y, living in the same space as the observation vector X_{obs}, and the idea is to impose (as a weak constraint) that $C\mathcal{M}(U,V) = Y$. In some sense, we are looking for Y, a better observation vector than X_{obs}, in the sense that it is perfectly matching a model solution $\mathcal{M}(U,V)$.

Let m be the Lagrange multiplier for this constraint, m lives in the observation space, and we can define the following Lagrangian:

$$\mathcal{L}((U,V,Y),m) = \mathcal{J}(U,V,Y) + \int_0^T \langle m, C\mathcal{M}(U,V) - Y\rangle \,\mathrm{d}t, \tag{2.11}$$

where the observation vector Y is now a variable of the primal cost function $\mathcal{J}$.

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

$$\min_{(U,V,Y)} \mathcal{J} = \min_{(U,V,Y)} \max_m \mathcal{L} = \max_m \min_{(U,V,Y)} \mathcal{L}$$

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

$$\mathcal{J}_D(m) = -\min_{(U,V)} \mathcal{L}\left((U,V,Y),m\right). \tag{2.12}$$

We have then the following result:

$$\min_{(U,V,Y)} \mathcal{J}(U,V,Y) = \max_m(-\mathcal{J}_D(m)) = -\min_m \mathcal{J}_D(m). \tag{2.13}$$

Mathematically, the minimization of $\mathcal{J}$, and hence the resolution of problems (2.3) and (2.7), 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,Y), it is quite easy to obtain an explicit definition of $\mathcal{J}_D$:

$$\mathcal{J}_D(m) = \frac{1}{2}\langle(\mathcal{D}+R)m, m\rangle - \langle d, m\rangle \tag{2.14}$$

where d is the innovation vector: $d = X_{obs} - CX_{background}$.

By considering that the model is linear, the operator $\mathcal{M}$ acts on (U, V) as

$$\mathcal{M}(U, V) = MV + NU, \tag{2.15}$$

where M and N are linear operators defined on the appropriate spaces.

The matrix $\mathcal{D}$ is called the representers' matrix [12, 13], and has the following definition:

$$\mathcal{D} = CMP_0M^TC^T + CNQN^TC^T \tag{2.16}$$

where M^T and N^T represent the adjoint model (adjoint operators of the direct 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.

2.8. Nonlinear extended 4D-PSAS algorithm computation

When the model (and/or the observation operator) is nonlinear, it is possible to extend the previous duality results in an empirical way [6]. Let m be a vector of the observation space, we first have to solve an adjoint (backward) model:

$$\begin{cases} -\dfrac{\mathrm{d}P}{\mathrm{d}t} = \left(\dfrac{\partial F}{\partial X}\right)^T P - C^T R^{-1}(m - X_{obs}), \\ P(T) = 0, \end{cases} \tag{2.17}$$

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

$$\begin{cases} \dfrac{\mathrm{d}X}{\mathrm{d}t} = F(X, P), \\ X(0) = X_{background} + P(0), \end{cases} \tag{2.18}$$

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 (2.17) and then the solution X_{n-1} of (2.18). 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 (2.18).

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.

3. Sequential methods: Kalman filtering

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 nonlinear nature. To deal with this, one usually linearizes the ordinary Kalman filter (KF) leading to the so-called extended Kalman filter (EKF) [23, 29, 32, 67]. We will also present the ensemble Kalman filter (EnKF), which allows one to get rid of too expensive computations of covariance matrices.

3.1. The extended Kalman filter

Consider a physical system described by

$$X(t_i) = \mathcal{M}(t_{i-1}, t_i)X(t_{i-1}) + U_i \tag{3.1}$$

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 an unknown term of the model (it can be a noise term, used to model the unknown parameters of the model [18]). 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:

$$\varepsilon_i = X_{obs}(t_i) - C_i X(t_i), \tag{3.2}$$

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:

$$X^f(t_i) = M(t_{i-1}, t_i)X^a(t_{i-1}), \tag{3.3}$$

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

$$P^f(t_i) = M(t_{i-1}, t_i)P^a(t_{i-1})M(t_{i-1}, t_i)^T + Q_i. \tag{3.4}$$

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:

$$X^a(t_i) = X^f(t_i) + K_i(X_{obs}(t_i) - C_i X^f(t_i)), \tag{3.5}$$

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

$$K_i = P^f(t_i)C_i^T \left(C_i P^f(t_i) C_i^T + R_i\right)^{-1}. \tag{3.6}$$

The corresponding analysis error covariance matrix is given by

$$P^a(t_i) = P^f(t_i) - P^f(t_i)C_i^T \left(C_i P^f(t_i) C_i^T + R_i\right)^{-1} C_i P^f(t_i). \tag{3.7}$$

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 covariance 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 ((3.4) and (3.7)) states 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.

3.2. The SEEK (Singular Evolutive Extended Kalman) filter

It seems that a relatively optimal Kalman filter is quite ambitious. 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 [17, 56].

We still need to impose that the filtering error should remain bounded. The propragation of the filter error is given by

$$X^a(t_i) - X^t(t_i) = (I - K_i C_i)\, M(t_{i-1}, t_i) \left(X^a(t_{i-1} - X^t(t_{i-1})\right) - K_i \varepsilon_i - (I - K_i C_i) U_i. \quad (3.8)$$

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.

But for computational reasons, it is also crucial to use low rank error covariance matrices P_i^a. 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 low rank matrix (with only several column vectors) 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

$$X^f(t_i) = M(t_{i-1}, t_i) X^a(t_{i-1}) \quad (3.9)$$

and

$$L_i = M(t_{i-1}, t_i) L_{i-1}. \tag{3.10}$$

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

$$U_i^{-1} = \left(U_{i-1} + (L_i^T L_i)^{-1} L_i^T Q_i L_i (L_i^T L_i)^{-1}\right)^{-1} + L_i^T C_i^T R_i^{-1} C_i L_i \tag{3.11}$$

and then compute the new analysis vector

$$X^a(t_i) = X^f(t_i) + (L_i U_i L_i^T) C_i^T R_i^{-1} \left(X_{obs}(t_i) - C_i X^f(t_i)\right). \tag{3.12}$$

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, 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.

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 (3.1). 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

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

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 [56].

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 (3.11), 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.

3.3. Ensemble Kalman filter

A main issue of the (extended) Kalman filter is the computational cost of propagating the covariance matrices in time. The dimension of P^f and P^a matrices is usually too large in geophysical problems, and there are several ways to avoid this point. One interesting approach is the ensemble Kalman filter (EnKF). The EnKF can be seen as a Monte Carlo approximation of the KF, avoiding evolving the full covariance matrix of the probability density function of the state vector [14, 24, 25, 26, 39].

As error statistics of background errors are not very well known, the idea is to generate a set of perturbed background states, with small perturbations around the background state with the same probability distribution. For $1 \leqslant j \leqslant M$, M being the size of the ensemble, we define the background ensemble members:

$$X_j(t_0) = X^b(t_0) + \varepsilon_j, \tag{3.13}$$

where X^b is the background state, and ε_j is the statistical perturbation, with a probability distribution consistent with the background error covariance matrix.

Then, we obtain an ensemble of forecast states with

$$X_j^f(t_i) = \mathcal{M}(t_{i-1}, t_i) X_j^a(t_{i-1}), \tag{3.14}$$

where we use the nonlinear model $\mathcal{M}(t_{i-1}, t_i)$. The correlation between these states gives some information about the forecast error statistics. The forecast error covariance matrix is then the actual covariance of the ensemble of states.

Then, the analysis state step is similar to the standard Kalman filter, with the computation of the Kalman gain matrix, and the correction is applied to each member:

$$X_j^a(t_i) = X_j^f(t_i) + K_i(X_{obs}(t_i) - C_i X_j^f(t_i)), \tag{3.15}$$

where K_i is the Kalman gain matrix computed with the actual covariance matrices of the ensemble. Then, the analysis error covariance matrix is computed from the ensemble of analysis states.

The EnKF is then *simply* a Kalman filter, applied to a discrete set (ensemble) of states (members). The covariance matrices are computed from this set, without using the linear or adjoint model. There are two main advantages: first, the computational cost is much lower, as there are no costly computations for the covariance matrices; and second, the covariance matrices represent the actual covariance statistics of the members in the ensemble.

4. Nudging schemes

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 nonlinearities (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.

4.1. 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

$$\begin{cases} \dfrac{\mathrm{d}X}{\mathrm{d}t} = F(X,U), \quad 0 < t < T, \\ X(0) = V. \end{cases} \tag{4.1}$$

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

$$\begin{cases} \dfrac{\mathrm{d}X}{\mathrm{d}t} = F(X,U) + K(X_{obs} - CX), \quad 0 < t < T, \\ X(0) = V, \end{cases} \tag{4.2}$$

where C is still the observation operator, and K is the nudging matrix. It is quite easy to understand 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 $X_{obs}(t)$. In the linear case

(where F and C are linear operators), the forward nudging method is nothing else than the Luenberger observer, also called asymptotic observer, where the operator K can be chosen so that the error goes to zero when time goes to infinity [49].

This algorithm was first used in meteorology [37], and then has been used with success in oceanography [68] and applied to a mesoscale model of the atmosphere [62]. Many results have also been carried out on the optimal determination of the nudging coefficients K [61, 70, 71].

The nudging algorithm is usually considered as a sequential data assimilation method. If one solves equation (4.2) with a numerical scheme, then it is equivalent with the following algorithm:

$$\begin{cases} X_n^f = X_{n-1} + \mathrm{d}t \times F(X_{n-1}, U_{n-1}), \\ X_n = X_n^f + K_n(X_{obs}(t_n) - C_n X_n^f), \end{cases} \tag{4.3}$$

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. It is also possible to consider suboptimal K matrices, that still correct all variables, and not only the observed ones [10].

4.2. 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 [2, 7].

We now assume that we have a final condition in (4.1) instead of an initial condition. This leads to the following backward equation

$$\begin{cases} \dfrac{\mathrm{d}\tilde{X}}{\mathrm{d}t} = F(\tilde{X}, U), \quad T > t > 0, \\ \tilde{X}(T) = \tilde{V}. \end{cases} \tag{4.4}$$

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

$$\begin{cases} \dfrac{\mathrm{d}\tilde{X}}{\mathrm{d}t} = F(\tilde{X}, U) - K(X_{obs} - C\tilde{X}), \quad T > t > 0, \\ \tilde{X}(T) = \tilde{V}. \end{cases} \tag{4.5}$$

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 $X_{obs}(t)$.

4.3. 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 [7].

The BFN algorithm is then the following:

$$\begin{cases} \dfrac{\mathrm{d}X_k}{\mathrm{d}t} = F(X_k, U) + K(X_{obs} - CX_k), \\ X_k(0) = \tilde{X}_{k-1}(0), \end{cases} \qquad \begin{cases} \dfrac{\mathrm{d}\tilde{X}_k}{\mathrm{d}t} = F(\tilde{X}_k, U) - K(X_{obs} - C\tilde{X}_k), \\ \tilde{X}_k(T) = X_k(T), \end{cases} \tag{4.6}$$

with $\tilde{X}_{-1}(0) = V$. Then, $X_0(0) = V$, 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.

The BFN algorithm has been tested successfully for the system of Lorenz equations, Burgers equation and a quasi-geostrophic ocean model in [8], for a shallow-water model in [4] and compared with a variational approach for all these models. It has been used to assimilate the wind data in a mesoscale model [15] and for the reconstruction of quantum states in [43].

4.4. DBFN: Diffusive Back and Forth Nudging algorithm

In the framework of oceanographic and meteorological problems, there is usually no diffusion in the model equations. However, the numerical equations that are solved contain some diffusion terms in order to both stabilize the numerical integration (or the numerical scheme is set to be slightly diffusive) and model some subscale turbulence processes. We can then separate the diffusion term from the rest of the model terms, and assume that the partial differential equations read:

$$\frac{\mathrm{d}X}{\mathrm{d}t} = F(X) + \nu\Delta X, \quad 0 < t < T, \tag{4.7}$$

where F has no diffusive terms, ν is the diffusion coefficient, and we assume that the diffusion is a standard second-order Laplacian (note that it could be a fourth or sixth order derivative in some oceanographic models, but for clarity, we assume here that it is a Laplacian operator).

We introduce the D-BFN algorithm in this framework, for $k \geqslant 1$:

$$\begin{cases} \dfrac{\mathrm{d}X_k}{\mathrm{d}t} = F(X_k) + \nu\Delta X_k + K(X_{obs} - C(X_k)), \\ X_k(0) = \tilde{X}_{k-1}(0), \qquad 0 < t < T, \\ \dfrac{\mathrm{d}\tilde{X}_k}{\mathrm{d}t} = F(\tilde{X}_k) - \nu\Delta \tilde{X}_k - K'(X_{obs} - C(\tilde{X}_k)), \\ \tilde{X}_k(T) = X_k(T), \qquad T > t > 0. \end{cases} \tag{4.8}$$

It is straightforward to see that the backward equation can be rewritten, using $t' = T - t$:

$$\frac{\mathrm{d}\tilde{X}_k}{\mathrm{d}t'} = -F(\tilde{X}_k) + \nu\Delta\tilde{X}_k + K'(X_{obs} - C(\tilde{X}_k)), \quad \tilde{X}_k(t' = 0) = X_k(T), \tag{4.9}$$

where $\tilde{X}$ is evaluated at time t'. Then the backward equation is well-posed, with an initial condition and the same diffusion operator as in the forward equation. The diffusion term both takes into account the subscale processes and stabilizes the numerical backward integrations, and the feedback term still controls the trajectory with the observations.

The main interest of this new algorithm is that for many geophysical applications, the non diffusive part of the model is reversible, and the backward model is then stable. Moreover, the forward and backward equations are now consistent in the sense that they will be both diffusive in the same way (as if the numerical schemes were the same in forward and backward integrations), and only the non-diffusive part of the physical model is solved backwards. Note that in this case, it is reasonable to set $K' = K$.

The DBFN algorithm has been tested successfully for a linear transport equation in [9] and for non-linear Burgers equation in [5].

4.5. Theoretical considerations

The convergence of the BFN algorithm has been proved by Auroux and Blum in [7] for linear systems of ordinary differential equations and full observations, by Ramdani et al. [57] for reversible linear partial differential equations (wave and Schrödinger equations), by Donovan et al. [21] for the reconstruction of quantum states. In [11], the authors consider the BFN algorithm on transport equations. They show that for non viscous equations (both linear transport and Burgers), the convergence of the algorithm holds under observability conditions. Convergence can also be proven for viscous linear transport equations under some strong hypothesis, but not for viscous Burgers' equation. Moreover, the authors show that the convergence rate is always exponential in time [11]. In [9], the authors prove the theoretical convergence of DBFN algorithm for linear transport equations.

Data Assimilation is the ensemble of techniques combining the mathematical information provided by the equations of the model and the physical information given by the observations in order to retrieve the state of a flow. In order to show that both BFN and DBFN algorithms achieve this double objective, let us give a formal explanation of the way these algorithms proceed.

If $K' = K$ and the forward and backward limit trajectory are equal, i.e. $\tilde{X}_\infty = X_\infty$, then taking the sum of the two equations in (4.6) shows that the limit trajectory X_∞ satisfies the model equation (4.1) (including possible model viscosity). Moreover, the difference between the two equations in (4.6) shows that the limit trajectory is solution of the following equation:

$$K(X_{obs} - C(X_\infty)) = 0. \tag{4.10}$$

Equation (4.10) shows that the limit trajectory perfectly fits the observations (through the observation operator, and the gain matrix).

In a similar way, for the DBFN algorithm, taking the sum of the two equations in (4.8) shows that the limit trajectory X_∞ satifies the model equations without diffusion:

$$\frac{\mathrm{d}X_\infty}{\mathrm{d}t} = F(X_\infty) \tag{4.11}$$

while taking the difference between the two same equations shows that X_∞ satisfies the Poisson equation:

$$\Delta X_\infty = -\frac{K}{\nu}(X_{obs} - C(X_\infty)) \tag{4.12}$$

which represents a smoothing process on the observations for which the degree of smoothness is given by the ratio $\frac{\nu}{K}$ [9]. (4.12) corresponds, in the case where C is a matrix and $K = kC^T R^{-1}$, to the Euler equation of the minimization of the following cost function

$$J(X) = k\langle R^{-1}(X_{obs} - CX), (X_{obs} - CX)\rangle + \nu \int_\Omega \|\nabla X\|^2 \qquad (4.13)$$

where the first term represents the quadratic difference to the observations and the second one is a first order Tikhonov regularisation term over the domain of resolution Ω. The vector X_∞, solution of (4.12), is the point where the minimum of this cost function is reached. This is a nice increment to the BFN algorithm, in which the limit trajectory fits the observations, while in the DBFN algorithm, the limit trajectory is the result of a smoothing process on the observations (which are often very noisy).

5. Conclusion

We have presented in this paper several data assimilation methods: variational methods (4D-VAR and 4D-PSAS), sequential methods (extended Kalman filter, ensemble Kalman filter), with their reduced-order version, and back and forth nudging methods (BFN and DBFN).

Variational methods are very powerfull, as they allow us to identify not only the initial condition of the system, but also model parameters, model error, ... However they require the adjoint model, and from a computational point of view, deriving the adjoint model and computing the adjoint state can be a tough task. People should think at the adjoint derivation while writing the direct code (that solves the direct model), so that some automatic differentiation tools could be used for deriving the adjoint model.

On the other hand, sequential methods are quite easy to implement, as there is no adjoint model/state. However, the size of the covariance matrices is usually too large to allow their full storage, and tricky linear algebra might be necessary to compute their update and matrix-vector computations. Some recent works have shown that Kalman filters could be used for parameter estimation as well: some ad hoc equations for the time evolution of the parameters are added to the system, and a Kalman filter applied to the coupled system allows one to identify both the system state and the model parameters (see e.g. [52]).

Finally, back and forth nudging algorithms (BFN and DBFN) are very easy to implement, as they do not impose any linearization of the model,

they do not require neither the construction of an adjoint model, nor an optimization procedure as in the 4D-VAR method. Moreover they are very fast, as they converge in less iterations than variational methods. As the nudging matrices are much more simple than Kalman gain matrices, the storage, computation and memory costs are much smaller for BFN and DBFN than for Kalman filters and variational methods. The progress of the DBFN, compared to the BFN method, is a better smoothing of the noise on the observations, which results from a more diffusive process (forward and backward). The distance from the reconstructed solution to the true one is also smaller with the DBFN algorithm than the one obtained with the BFN method.

Depending on the application, the size of the problem, the amount of observations, ..., people should carefully choose the class of data assimilation algorithms that is more suitable to their own problem.

Bibliography

[1] L. Amodei, "Solution approchée pour un problème d'assimilation de données météorologiques avec prise en compte de l'erreur modèle", *C. R. Acad. Sci. Paris, Ser. II* **321** (1995), p. 1087-1094.

[2] D. Auroux, "Étude de différentes méthodes d'assimilation de données pour l'environnement", PhD Thesis, University of Nice - Sophia Antipolis (France), 2003.

[3] ———, "Generalization of the dual variational data assimilation algorithm to a nonlinear layered quasi-geostrophic ocean model", *Inverse Probl.* **23** (2007), no. 6, p. 2485-2503.

[4] ———, "The Back and Forth Nudging algorithm applied to a shallow water model, comparison and hybridization with the 4D-VAR", *Int. J. Numer. Methods Fluids* **61** (2009), no. 8, p. 911-929.

[5] D. Auroux, P. Bansart & J. Blum, "An evolution of the Back and Forth Nudging for geophysical data assimilation: application to Burgers equation and comparisons", *Inverse Probl. Sci. Eng.* **21** (2013), no. 3, p. 399-419.

[6] D. Auroux & J. Blum, "Data assimilation methods for an oceanographic problem", in *Multidisciplinary methods for analysis optimization and control of complex systems*, Mathematics in Industry, vol. 6, Springer, 2004.

[7] ———, "Back and forth nudging algorithm for data assimilation problems", *C. R. Acad. Sci. Paris, Ser. I* **340** (2005), no. 12, p. 873-878.

[8] ———, "A nudging-based data assimilation method for oceanographic problems: the Back and Forth Nudging (BFN) algorithm", *Nonlin. Proc. Geophys.* **15** (2008), p. 305-319.

[9] D. Auroux, J. Blum & M. Nodet, "Diffusive Back and Forth Nudging algorithm for data assimilation", *C. R. Acad. Sci. Paris, Ser. I* **349** (2011), no. 15-16, p. 849-854.

[10] D. Auroux & S. Bonnabel, "Symmetry-based observers for some water-tank problems", *IEEE Trans. Autom. Contr.* **56** (2011), no. 5, p. 1046-1058.

[11] D. Auroux & M. Nodet, "The Back and Forth Nudging algorithm for data assimilation problems: theoretical results on transport equations", *ESAIM, Control Optim. Calc. Var.* **18** (2012), no. 2, p. 318-342.

[12] A. F. Bennett, *Inverse methods in physical oceanography*, Cambridge Monographs on Mechanics and Applied Mathematics, Cambridge University Press, 1992, xvi+346 pages.
[13] ———, *Inverse Modeling of the Ocean and Atmosphere*, Cambridge University Press, 2002, xxii+234 pages.
[14] J. Blum, F.-X. Le Dimet & I. M. Navon, "Data assimilation for geophysical fluids", in *Computational methods for the atmosphere and the oceans*, Handbook of Numerical Analysis, vol. 14, Elsevier, 2009, p. 385-441.
[15] A. Boilley & J.-f. Mahfouf, "Assimilation of low-level wind in a high resolution mesoscale model using the back and forth nudging algorithm", *Tellus A* **64** (2012), p. 18697.
[16] C. G. Broyden, "A new double-rank minimization algorithm", *Notices American Math. Soc.* **16** (1969), p. 670.
[17] M. A. Cane, A. Kaplan, R. N. Miller, B. Tang, E. C. Hackert & A. J. Busalacchi, "Mapping tropical Pacific sea level: data assimilation via a reduced state Kalman filter", *J. Geophys. Res.* **101(C10)** (1996), p. 22599-22617.
[18] A. Carrassi & S. Vannitsem, "Deterministic treatment of model error in geophysical data assimilation", in *Mathematical Paradigms of Climate Science*, Springer INdAM Series, vol. 15, Springer, 2016, p. 175-213.
[19] P. Courtier, "Dual formulation of four-dimensional variational assimilation", *Quart. J. R. Meteor. Soc.* **123** (1997), p. 2449-2461.
[20] P. Courtier & O. Talagrand, "Variational assimilation of meteorological observations with the adjoint equations Part 2. Numerical results", *Quart. J. Roy. Meteor. Soc.* **113** (1987), p. 1329-1347.
[21] A. Donovan, M. Mirrahimi & P. Rouchon, "Back and Forth Nudging for quantum state reconstruction", in *4th Int. Symp. Communications Control Signal Proc.*, 2010, p. 1-5.
[22] S. Durbiano, "Vecteurs caractéristiques de modèles océaniques pour la réduction d'ordre en assimilation de données", PhD Thesis, University of Grenoble I (France), 2001.
[23] G. Evensen, "Using the extended Kalman filter with a multilayer quasi-geostrophic ocean model", *J. Geophys. Res.* **97** (1992), p. 17905-17924.
[24] ———, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics", *J. Geophys. Res.* **99** (1994), no. C5, p. 10143-10162.
[25] ———, "The Ensemble Kalman Filter: theoretical formulation and practical implementation", *Ocean Dynamics* **53** (2003), p. 343-367.
[26] ———, *Data assimilation: the Ensemble Kalman Filter*, Springer, 2009.
[27] I. Fukumori, "Assimilation of Topex sea level measurements with a reduced-gravity, shallow water model of the tropical Pacific Ocean", *J. Geophys. Res.* **100(C12)** (1995), p. 25027-25039.
[28] I. Fukumori, B. Jérôme, C. Wunsch & D. B. Haidvogel, "Assimilation of sea surface topography into an ocean circulation model using a steady state smoother", *J. Phys. Oceanogr.* **23** (1993), p. 1831-1855.
[29] P. Gauthier, P. Courtier & P. Moll, "Assimilation of simulated wind lidar data with a Kalman filter", *Mon. Wea. Rev.* **121** (1993), p. 1803-1820.
[30] A. Gelb, *Applied Optimal Estimation*, MIT Press, 1974.
[31] M. Ghil, "Meteorological data assimilation for oceanographers. Part I: Description and theoretical framework", *Dyn. Atmos. Oceans* **13** (1989), no. 3-4, p. 171-218.

[32] M. Ghil, S. Cohn & A. Dalcher, "Sequential estimation, data assimilation and initialization", in *The interaction between objective analysis and initialization*, Publ. Meteor., vol. 127, McGill University, 1982.
[33] M. Ghil & P. Manalotte-Rizzoli, "Data assimilation in meteorology and oceanography", *Adv. Geophys.* **33** (1991), p. 141-265.
[34] J. C. Gilbert & C. Lemaréchal, "Some numerical experiments with variable storage quasi-Newton algorithms", *Math. Program.* **45** (1989), p. 407-435.
[35] L. Gourdeau, S. Arnault, Y. Ménard & J. Merle, "GEOSAT sea-level assimilation in a tropical Atlantic model using Kalman filter", *Ocean. Acta* **15** (1992), p. 567-574.
[36] A. Griewank, "Automatic Differentiation", in *Princeton Companion to Applied Mathematics*, Princeton University Press, 2014.
[37] J. E. Hoke & R. A. Anthes, "The initialization of numerical models by a dynamic initialization technique", *Mon. Wea. Rev.* **104** (1976), p. 1551-1556.
[38] W. R. Holland, "The role of mesoscale eddies in the general circulation of the ocean", *J. Phys. Oceanogr.* **8** (1978), no. 3, p. 363-392.
[39] P. L. Houtekamer & H. L. Mitchell, "Data assimilation using an ensemble Kalman filter technique", *Mon. Wea. Rev.* **126** (1998), p. 796-811.
[40] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Mathematics in Science and Engineering, vol. 64, Academic Press, 1970, xiv+376 pages.
[41] E. Kalnay, *Atmospheric modeling, data assimilation and predictability*, Cambridge University Press, 2003, 341 pages.
[42] F.-X. Le Dimet & O. Talagrand, "Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects", *Tellus A* **38** (1986), no. 2, p. 97-110.
[43] Z. Leghtas, M. Mirrahimi & P. Rouchon, "Observer-based quantum state estimation by continuous weak measurement", in *American Control Conference (ACC)*, 2011, p. 4334-4339.
[44] J. M. Lewis & J. C. Derber, "The use of adjoint equations to solve a variational adjustment problem with convective constraints", *Tellus A* **37** (1985), p. 309-322.
[45] J.-L. Lions, *Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles*, Dunod, 1968, xii+426 pages.
[46] D. C. Liu & J. Nocedal, "On the limited memory BFGS method for large scale optimization", *Math. Program.* **45** (1989), no. 3, p. 503-528.
[47] S. Louvel, "Étude d'un algorithme d'assimilation variationnelle de données à contrainte faible. Mise en œuvre sur le modèle océanique aux équations primitives MICOM", PhD Thesis, Université Paul Sabatier, Toulouse (France), 1999.
[48] ———, "Implementation of a dual variational algorithm for assimilation of synthetic altimeter data in the oceanic primitive equation model MICOM", *J. Geophys. Res.* **106** (2001), p. 9199-9212.
[49] D. Luenberger, "Observers for multivariable systems", *IEEE Trans. Autom. Contr.* **11** (1966), p. 190-197.
[50] B. Luong, J. Blum & J. Verron, "A variational method for the resolution of a data assimilation problem in oceanography", *Inverse Probl.* **14** (1998), p. 979-997.
[51] B. Mohammadi & O. Pironneau, *Applied shape optimization for fluids*, Numerical Mathematics and Scientific Computation, Clarendon Press, 2001, xvi+251 pages.
[52] P. Moireau & D. Chapelle, "Reduced-order Unscented Kalman Filtering with application to parameter identification in large-dimensional systems", *ESAIM, Control Optim. Calc. Var.* **17** (2011), no. 2, p. 380-405, erratum in *ibid* **17** (2011), no. 2, p. 406-409.

[53] A. M. Moore, "Data assimilation in a quasi-geostrophic open-ocean model of the Gulf-Stream region using the adjoint model", *J. Phys. Oceanogr.* **21** (1991), p. 398-427.

[54] D. A. Nechaev & M. I. Yaremchuk, "Application of the adjoint technique to processing of a standard section data set: world ocean circulation experiment section S4 along 67°S in the Pacific Ocean", *J. Geophys. Res.* **100(C1)** (1994), p. 865-879.

[55] J. Pedlosky, *Geophysical fluid dynamics*, Springer, 1979, xii+624 pages.

[56] D. T. Pham, J. Verron & M.-C. Roubaud, "A Singular Evolutive Extended Kalman filter for data assimilation in oceanography", *Inverse Probl.* **14** (1998), p. 979-997.

[57] K. Ramdani, M. Tucsnak & G. Weiss, "Recovering the initial state of an infinite-dimensional system using observers", *Automatica* **46** (2010), no. 10, p. 1616-1625.

[58] N. Rostaing-Schmidt & E. Hassold, "Basic function representation of programs for automatic differentiation in the Odyssée system", in *High performance computing in the geosciences*, Kluwer Academic Publishers, 1994, p. 207-222.

[59] J. Schröter, U. Seiler & M. Wenzel, "Variational assimilation of GEOSAT data into an eddy-resolving model of the Gulf Stream area", *J. Phys. Oceanogr.* **23** (1993), p. 925-953.

[60] J. Sheinbaum & D. L. T. Anderson, "Variational assimilation of XBT data. Part I", *J. Phys. Oceanogr.* **20** (1990), p. 672-688.

[61] D. R. Stauffer & J.-W. Bao, "Optimal determination of nudging coefficients using the adjoint equations", *Tellus A* **45** (1993), p. 358-369.

[62] D. R. Stauffer & N. L. Seaman, "Use of four dimensional data assimilation in a limited area mesoscale model - Part 1: Experiments with synoptic-scale data", *Mon. Wea. Rev.* **118** (1990), p. 1250-1277.

[63] O. Talagrand, "Assimilation of observations, an introduction", *Journal of the Met. Soc. of Japan* **75** (1997), no. 1B, p. 191-209.

[64] O. Talagrand & P. Courtier, "Variational assimilation of meteorological observations with the adjoint vorticity equation. Part I: Theory", *Quart. J. R. Meteor. Soc.* **113** (1987), p. 1311-1328.

[65] W. C. Thacker & R. B. Long, "Fitting dynamics to data", *J. Geophys. Res.* **93** (1988), p. 1227-1240.

[66] F. Veersé, D. Auroux & M. Fisher, "Limited-memory BFGS diagonal preconditioners for a data assimilation problem in meteorology", *Optimization and Engineering* **1** (2000), no. 3, p. 323-339.

[67] J. Verron, L. Gourdeau, D. T. Pham, R. Murtugudde & 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** (1999), p. 5441-5458.

[68] J. Verron & W. R. Holland, "Impact de données d'altimétrie satellitaire sur les simulations numériques des circulations générales océaniques aux latitudes moyennes", *Ann. Geophys.* **7** (1989), no. 1, p. 31-46.

[69] A. Vidard, "Vers une prise en compte des erreurs modèle en assimilation de données 4D-variationnelle - Application à un modèle réaliste d'océan", PhD Thesis, University of Grenoble I (France), 2001.

[70] A. Vidard, F.-X. Le Dimet & A. Piacentini, "Determination of optimal nudging coefficients", *Tellus A* **55** (2003), p. 1-15.

[71] X. Zou, I. M. Navon & F.-X. Le Dimet, "An optimal nudging data assimilation scheme using parameter estimation", *Quart. J. Roy. Meteor. Soc.* **118** (1992), p. 1163-1186.