Nudging and backward-forward approach for data assimilation

Prof. Didier Auroux

Abstract Nudging is a data assimilation method that uses dynamical relaxation to adjust a model towards observations. The standard nudging algorithm consists in adding to the model equations a feedback term, proportional to the difference between the observations and the corresponding model state. Also known as the Luenberger (or asymptotic) observer, it theoretically requires an infinite time window to converge. The Back and Forth Nudging (BFN) algorithm has been introduced in order to extend the efficiency of nudging to finite/small time windows. It consists in alternately solving the model forwards and backwards in time, with a nudging term in both cases, over the assimilation window. These approaches can be extended to more complex observers, for which non-observed variables can also be corrected with observed ones. We will give in this chapter an overview of nudging, observers, and backward-forward algorithms, with applications to oceanography and fluid dynamics.

1 Nudging schemes

Nudging is one of the very first data assimilation schemes that has been applied to geophysical systems. It was first used in meteorology [16], and then has been used with success in oceanography [29] and applied to a mesoscale model of the atmosphere [28]. Many results have also been carried out on the optimal determination of the nudging coefficients [31, 27, 30].

Several decades later, its ease of implementation and light computational cost, in comparison with 4D-Var or Kalman algorithms, make it still competitive. It indeed does not require any linearization step, nor adjoint model, nor manipulation of large covariance matrices . . .

Didier Auroux
Université Côte d'Azur (France), e-mail: didier.auroux@univ-cotedazur.fr

1.1 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 $\mathbf{x}(t) \in \mathbb{R}^n$ be the model state at time t, and let consider a generic model equation:

$$\frac{d\mathbf{x}}{dt} = \mathbf{m}(\mathbf{x}), \quad 0 < t < T, \tag{1}$$

with an initial condition $\mathbf{x}(0) = \mathbf{x}_0$. The assimilation window is assumed to be $t \in [0;T]$. Any (set of) ordinary differential equation(s) (ODE) can be written in such way. Note also that after spatial discretization (the choice of discretization is not discussed here), any partial differental equation (PDE) can also be written as in Eq. (1). We denote by n the dimension of the (discrete or discretized) state vector at any time.

We assume that we have observations or measurements $\mathbf{d}(t) \in \mathbb{R}^p$ of the state variable $\mathbf{x}(t) \in \mathbb{R}^n$. We assume that $p \leq n$, as we expect the measurements to live in a smaller dimension space than the state space. For theoretical purposes, we assume that the observations are available at any time t, but of course, we will later give implementation details in more realistic situations where the measurements are only available at several discrete times.

Finally, let $\mathbf{h}$ be the observation operator, that maps a model state vector $\mathbf{x} \in \mathbb{R}^n$ to an observation-like vector $\mathbf{h}(\mathbf{x}) \in \mathbb{R}^p$.

Then the nudging algorithm applied to Eq. (1) simply writes:

$$\frac{d\mathbf{x}}{dt} = \mathbf{m}(\mathbf{x}) + \mathbf{k}(\mathbf{d} - \mathbf{h}(\mathbf{x})), \quad 0 < t < T, \tag{2}$$

with the same initial condition $\mathbf{x}(0) = \mathbf{x}_0$. $\mathbf{k}$ is the nudging operator from $\mathbb{R}^p$ to $\mathbb{R}^n$.

It is quite easy to understand that if $\mathbf{k}$ is *large* enough, then the state vector transposed into the observation space (through the observation operator) $\mathbf{h}(\mathbf{x}(t))$ will tend towards the observation vector $\mathbf{d}(t)$. Indeed, in the linear case (where the model operator $\mathbf{m}$, the observation operator $\mathbf{h}$ and the nudging operator $\mathbf{k}$ are linear operators), the nudging method is nothing else than the Luenberger observer, also called asymptotic observer, where the operator $\mathbf{k}$ can sometimes be chosen so that the error goes to zero when time goes to infinity [22].

Note that the nudging algorithm is usually considered as a sequential data assimilation method. This can be understood if Eq. (2) is solved using a splitting method. For instance, with an Euler scheme, one can use the following algorithm:

$$\begin{cases} \mathbf{x}(t_n^-) = \mathbf{x}(t_{n-1}) + (t_n - t_{n-1})\, \mathbf{m}(\mathbf{x}(t_{n-1})), \\ \mathbf{x}(t_n) = \mathbf{x}(t_n^-) + (t_n - t_{n-1})\, \mathbf{k}(\mathbf{d}(t_n) - \mathbf{h}(\mathbf{x}(t_n^-))), \end{cases} \tag{3}$$

which can be seen as a Kalman filter's algorithm with its forecast and correction steps. Then, if at any time the nudging gain $\mathbf{k}$ is set in an optimal way, it is quite easy to see that it will exactly be the Kalman gain. Note that it is also possible to consider suboptimal nudging gains, that still correct all variables, and not only the observed ones [11].

1.2 Convergence in the linear case

We consider here a linear case: the model operator $\mathbf{m}$, the observation operator $\mathbf{h}$ and the nudging operator $\mathbf{k}$ are linear operators, hence matrices, and we denote them by $\mathbf{M} \in \mathcal{M}_n(\mathbb{R})$, $\mathbf{H} \in \mathcal{M}_{p,n}(\mathbb{R})$ and $\mathbf{K} \in \mathcal{M}_{n,p}(\mathbb{R})$ respectively.

Let assume that there are no errors here, which means that the observations perfectly fit the true state $\mathbf{x}_t$: $\mathbf{d} = \mathbf{H}\mathbf{x}_t$. We also assume the model to be perfect, so that the true state is a model solution:

$$\frac{d\mathbf{x}_t}{dt} = \mathbf{M}\mathbf{x}_t, \tag{4}$$

and the nudging system is then

$$\frac{d\mathbf{x}}{dt} = \mathbf{M}\mathbf{x} + \mathbf{K}(\mathbf{d} - \mathbf{H}\mathbf{x}). \tag{5}$$

Let $\mathbf{e} = \mathbf{x} - \mathbf{x}_t$ be the error on the state. By considering the difference between Eq. (5) and Eq. (4), $\mathbf{e}$ is propagated in time by the following dynamical system:

$$\frac{d\mathbf{e}}{dt} = (\mathbf{M} - \mathbf{K}\mathbf{H})\mathbf{e}. \tag{6}$$

If $\mathbf{M} - \mathbf{K}\mathbf{H}$ is a Hurwitz matrix, i.e. its spectrum is strictly included in the half-plane $\{\lambda \in \mathbb{C}; Re(\lambda) < 0\}$, then $\mathbf{e} \to 0$, i.e. $\mathbf{x} \to \mathbf{x}_t$ when $t \to +\infty$. This gives the asymptotic convergence of the Luenberger observer.

The pole assignment (or pole placement) method gives the existence of nudging gain matrices such that the observer system is stable, and hence the nudging solution converges towards the true state [4]:

Proposition 1 *If a system* $(\mathbf{M}, \mathbf{H})$ *is observable, then there exists a matrix* $\mathbf{K}$ *such that* $\mathbf{M} - \mathbf{K}\mathbf{H}$ *is stable, i.e. all eigenvalues have a strictly negative real part.*

Note that for linear systems, the observability condition is ensured if and only if the rank of the observability matrix

$$\left(\mathbf{H}; \mathbf{H}\mathbf{M}; \mathbf{H}\mathbf{M}^2; \ldots; \mathbf{H}\mathbf{M}^{n-1}\right) \tag{7}$$

is equal to n.

In order to illustrate the pole assignment method, we consider here a very simple example in dimension $n = 2$:

$$\mathbf{M} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \tag{8}$$

The eigenvalues of this matrix are 0 and 2, so that in one direction, the error on the state will remain constant in time, and in another direction, the error will increase exponentially in time. Let assume that $p = 1$ and only the first component of the state is observed: $\mathbf{H} = \begin{pmatrix} 1 & 0 \end{pmatrix}$. We are then looking for a matrix

$$\mathbf{K} = \begin{pmatrix} k_1 \\ k_2 \end{pmatrix}$$

such that $\mathbf{M} - \mathbf{HF}$ is stable. The two eigenvalues of $\mathbf{M} - \mathbf{HF}$ are

$$\lambda = \frac{-(k_1 - 2) \pm \sqrt{k_1^2 - 4k_2 + 4}}{2}.$$

Hence, for any desired λ, we can set

$$k_1 = -2\lambda + 2, \quad k_2 = \frac{k_1^2}{4} + 1, \tag{9}$$

so that both eigenvalues of $\mathbf{M} - \mathbf{HF}$ are λ. For instance, $k_1 = 4$ and $k_2 = 5$ lead to $\lambda = -1$, so that the error will now decrease exponentially in time, with a decay rate of 1.

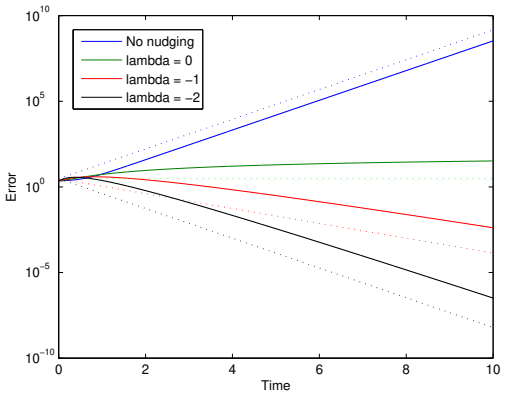

Fig. 1 Evolution of the error norm $\|\mathbf{e}\|$ versus time for the model given in Eq. (8), using various gain matrices $\mathbf{K}$ corresponding to various desired decay rates λ. The dotted lines correspond to a growth rate of 2 (blue), no variation (green), and decay rates of -1 (red) and -2 (black).

Fig. 1 shows the evolution of the error norm $\|\mathbf{e}\|$ versus time, in logarithmic scale, for this simple linear model. The dotted lines correspond to constant growth or decay rates. In blue, there is no nudging, we use $\mathbf{K} = \mathbf{0}$, so that the evolution of the error is driven by the largest eigenvalue of the matrix $\mathbf{M}$, which is 2. In green, red and black, we use gain matrices $\mathbf{K}$ defined by Eq. (9), so that the observed decay rates of the error are consistant with the desired rates (given by λ).

1.3 Extension to nonlinear cases: example on the Lorenz model

We now consider a nonlinear system, where the (exact) model equation for the true state is:

$$\frac{d\mathbf{x}_t}{dt} = \mathbf{M}\mathbf{x}_t + \mathbf{m}(\mathbf{x}_t), \tag{10}$$

where $\mathbf{M}$ stands for the linear part of the model, and $\mathbf{m}(.)$ is a nonlinear function, assumed to be differential and Lipschitz continuous:

$$\|\mathbf{m}(\mathbf{x}_1) - \mathbf{m}(\mathbf{x}_1)\| \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|, \quad \forall \mathbf{x}_1, \mathbf{x}_2, \tag{11}$$

for a given Lipschitz constant $L > 0$.

As in the previous section, we assume that the observation operator is linear, observations are perfect ($\mathbf{d} = \mathbf{H}\mathbf{x}_t$), and the linear model is observable. From the pole assignment method, we then know that there exists a matrix $\mathbf{K}$ such that $\mathbf{M}-\mathbf{K}\mathbf{H}$ is stable. The standard Luenberger observer is then:

$$\frac{d\mathbf{x}}{dt} = \mathbf{M}\mathbf{x} + \mathbf{m}(\mathbf{x}) + \mathbf{K}(\mathbf{d} - \mathbf{H}\mathbf{x}), \tag{12}$$

so that the error $\mathbf{e} = \mathbf{x} - \mathbf{x}_t$ is now solution of the following dynamical system:

$$\frac{d\mathbf{e}}{dt} = (\mathbf{M} - \mathbf{K}\mathbf{H})\mathbf{e} + \mathbf{m}(\mathbf{x}) - \mathbf{m}(\mathbf{x}_t). \tag{13}$$

Taking the inner product of Eq. (13) with $\mathbf{e}$ leads to

$$\frac{1}{2}\frac{d\|\mathbf{e}\|^2}{dt} = \mathbf{e} \cdot ((\mathbf{M} - \mathbf{K}\mathbf{H})\mathbf{e}) + \mathbf{e} \cdot (\mathbf{m}(\mathbf{x}) - \mathbf{m}(\mathbf{x}_t)),$$

that can be bounded by

$$\frac{1}{2}\frac{d\|\mathbf{e}\|^2}{dt} \leq (\lambda_{max} + L)\|\mathbf{e}\|^2, \tag{14}$$

thanks to Cauchy-Schwarz inequality, and Lipschitz continuity of $\mathbf{m}$. λ_{max} is the largest eigenvalue of $\mathbf{M} - \mathbf{K}\mathbf{H}$. As $(\mathbf{M}, \mathbf{H})$ is observable, from the pole assignment method, we can actually place the eigenvalues as we want, and in particular, we can choose $\mathbf{K}$ such that all eigenvalues of $\mathbf{M} - \mathbf{K}\mathbf{H}$ are (of real part) strictly smaller than $-L$, the opposite of the Lipschitz constant of $\mathbf{m}$. Thus, the largest eigenvalue $\lambda_{max} < -L$ and this ensures the asymptotic decrease of the norm of the error: $\mathbf{x} \to \mathbf{x}_t$ when time goes to infinity.

In order to illustrate this result, we consider here the Lorenz system [21]:

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y-x), \\ \frac{dy}{dt} &= \rho x - y - xz, \\ \frac{dz}{dt} &= xy - \beta z, \end{aligned} \tag{15}$$

with the standard values of parameters $\sigma = 10$, $\rho = 28$ and $\beta = \frac{8}{3}$, and where $\mathbf{x} = (x; y; z)$ is the state vector in $\mathbb{R}^3$.

The model (15) can be decomposed into a linear part $\mathbf{M}$ and a nonlinear part $\mathbf{m}$ as in Eq. (10):

$$\mathbf{M} = \begin{pmatrix} -\sigma & \sigma & 0 \\ \rho & -1 & 0 \\ 0 & 0 & -\beta \end{pmatrix}, \qquad \mathbf{m}(\mathbf{x}) = \begin{pmatrix} 0 \\ -xz \\ xy \end{pmatrix}. \tag{16}$$

Assuming that all considered trajectories are bounded, then the function $\mathbf{m}$ is Lipschitz continuous.

If for instance we choose $\mathbf{H} = (1 \;\; 0 \;\; 1)$, which means that there is only one observation of the system, and it corresponds to the sum of the first and third components of the state, we can easily see that the observability matrix (7) is invertible, so that the system $(\mathbf{M}, \mathbf{H})$ is observable.

Note that for other choices of $\mathbf{H}$, e.g. $\mathbf{H} = (1 \;\; 0 \;\; 0)$ or $\mathbf{H} = (0 \;\; 1 \;\; 0)$, the system is not observable, so that it is not possible to place the poles anywhere, but we can still prove that it is possible to find matrices $\mathbf{K}$ such that $\mathbf{M} - \mathbf{KH}$ is stable. As an example, with $\mathbf{H} = (1 \;\; 0 \;\; 0)$, one can choose $\mathbf{K} = (0 \;\; \rho \;\; 0)^T$ and the three eigenvalues of $\mathbf{M} - \mathbf{KH}$ are then -1, $-\beta$ and $-\sigma$.

We can also easily define nonlinear observers, where $\mathbf{K}$ is now a nonlinear operator. As an example, we assume here that only the first component x is observed, which means that $\mathbf{H} = (1 \;\; 0 \;\; 0)$. In such case, we can consider the following observer system:

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y-x), \\ \frac{dy}{dt} &= \rho x_t - y - x_t z, \\ \frac{dz}{dt} &= x_t y - \beta z, \end{aligned} \tag{17}$$

which is exactly the original Lorenz model where we replaced x by the observation $x_t = \mathbf{H}\mathbf{x}_t$ in the y and z equations. The idea is quite easy to understand: the observation gives us the first component of the true state x_t, then we use it (instead of the observer's first component x) in the other equations of the observer, so that y and z will be controlled by x_t. Eq. (17) can be rewritten in a more explicit way:

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y-x),\\ \frac{dy}{dt} &= \rho x - y - xz + \rho(x_t - x) - z(x_t - x),\\ \frac{dz}{dt} &= xy - \beta z + y(x_t - x),\end{aligned} \tag{18}$$

which is then the previous (linear) observer $\mathbf{K} = (0 \quad \rho \quad 0)^T$ augmented by an additional nonlinear term $\mathbf{k}(\mathbf{x}) = (0 \quad -z \quad y)^T$.

The error $\mathbf{e}$ is then solution of

$$\frac{d\mathbf{e}}{dt} = (\mathbf{M} - \mathbf{KH})\mathbf{e} + \mathbf{S}(t)\mathbf{e}, \tag{19}$$

where

$$\mathbf{M} - \mathbf{KH} = \begin{pmatrix} -\sigma & \sigma & 0 \\ 0 & -1 & -0 \\ 0 & 0 & -\beta \end{pmatrix}, \qquad \mathbf{S}(t) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -x_t \\ 0 & x_t & 0 \end{pmatrix}.$$

The eigenvalues of $\mathbf{M} - \mathbf{KH} + \mathbf{S}(t)$ are the roots of the polynomial

$$(\lambda + \sigma)\left(\lambda^2 + \lambda(1+\beta) + (\beta + x_t^2)\right),$$

and as $\beta > 0$ and $\sigma > 0$, then for any value of x_t, all three eigenvalues are (of real part) strictly negative, so that the error asymptotically decreases in time.

We now numerically illustrate this convergence by choosing the following starting points for the initialization of the Lorenz system:

$$\mathbf{x}_t(0) = \begin{pmatrix} 5 \\ -5 \\ -4 \end{pmatrix}, \quad \mathbf{x}(0) = -\mathbf{x}_t(0) = \begin{pmatrix} -5 \\ 5 \\ 4 \end{pmatrix}.$$

Fig. 2 shows the evolution of the observer and true states versus time: the three components x, y and z are respectively shown on top left, top right, and bottom left figures. The true state $\mathbf{x}_t$ is represented in blue, the observer state without nudging is in green, and the observer state with nudging is in red. When there is no nudging, the state $\mathbf{x}$ remains far away from the true state $\mathbf{x}_t$. But when we use the previously defined feedback term, the observer state quickly converges towards the true state.

The bottom right figure shows the evolution of $\|\mathbf{e}\|$, the norm of the error, versus time in logarithmic scale. The slope approximately corresponds to a decay rate of 1. Note that the oscillations are due to the nonlinearities, the eigenvalues depending on t.

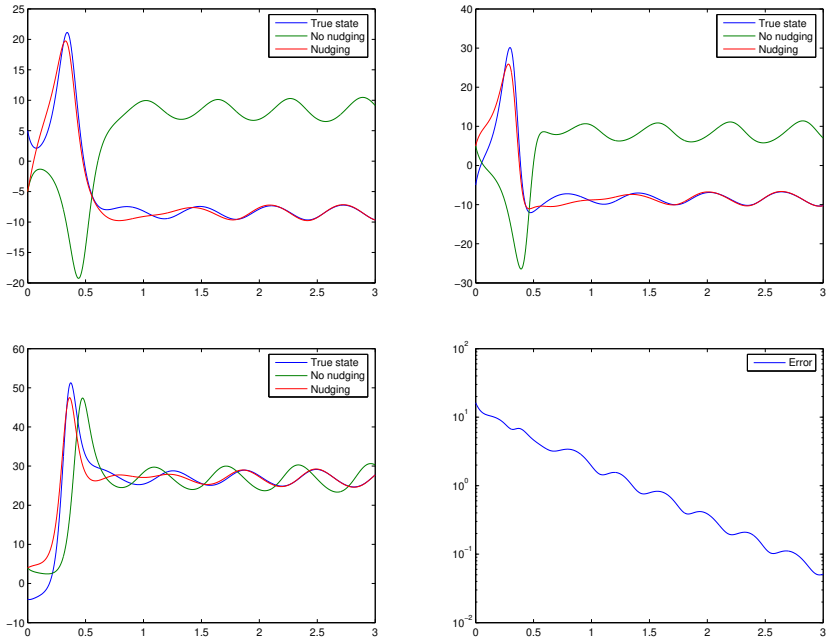

Fig. 2 Evolution of the state versus time: x (top left), y (top right), and z (bottom left), for the true state $\mathbf{x}_t$ and the observer state $\mathbf{x}$ with or without the nudging term. Evolution of the norm of the error $\mathbf{e}$ versus time, in logarithmic scale (bottom right).

2 Observers and controllability of non-observed variables

As seen in the previous subsection, on Lorenz model, usually all variables of the model are not observed, only some (few) of them. We study in this section the design of observers that will control non-observed variables, so that all variables, observed or not, will be corrected, leading to a global convergence of the observer state towards the true state.

2.1 Observer design on a 2D shallow water model

We consider here a 2D shallow water model, written in a non conservative form:

$$\frac{\partial h}{\partial t} = -\nabla \cdot (h\mathbf{v}), \tag{20}$$

$$\frac{\partial \mathbf{v}}{\partial t} = -(\mathbf{v} \cdot \nabla)\mathbf{v} - g\,\nabla h, \tag{21}$$

on a square domain, with rigid boundaries, and no-slip lateral boundary conditions. The state is $(h, \mathbf{v})$, h being the water height, and $\mathbf{v} \in \mathbb{R}^2$ the velocity vector. Finally, g is the gravity. This simple model is derived from Navier-Stokes equations, based on the conservation of mass and of momentum, assuming the horizontal scale is much greater than the vertical one [24].

In a geophysical framework, assuming that only the water height h is observed, the goal is to retrieve both variables (height and velocity). Let $\mathbf{d} = \mathbf{H}(h_t, \mathbf{v}_t) = h_t$ be the available observations in this framework. As the height is a scalar function, the observation also, so that we will use the notation $d = h_t$ in the following.

Any (nonlinear) observer for this model writes:

$$\frac{\partial h}{\partial t} = -\nabla \cdot (h\mathbf{v}) + f_h(d, \mathbf{v}, h), \tag{22}$$

$$\frac{\partial \mathbf{v}}{\partial t} = -(\mathbf{v} \cdot \nabla)\mathbf{v} - g\,\nabla h + \mathbf{f}_\mathbf{v}(d, \mathbf{v}, h), \tag{23}$$

where f_h and $\mathbf{f}_\mathbf{v}$ are respectively scalar and 2D-vectorial functions representing the feedback terms in the observer equations.

The idea is then to impose a couple of conditions on the feedback terms. First, f should be equal to 0 when the estimated (observer) height h is equal to the observed height d. Then, the feedback terms should preserve the model properties, which include here symmetry (invariance to translations and rotations). Knowing that the observations can be noisy, we also want the observer to have smoothing properties, typically using convolution with smoothing kernels. Finally, the goal of the observer is to ensure asymptotic convergence of the error, at least for the linearized system.

By taking into account these conditions, and by choosing the smallest order of derivatives (in order not to have large errors due to observation noise), one of the simplest feedback is then:

$$f_h(d, \mathbf{v}, h) = \varphi_h * (h - d), \qquad \mathbf{f}_\mathbf{v}(d, \mathbf{v}, h) = \varphi_\mathbf{v} * \nabla(h - d), \tag{24}$$

where φ_h and $\varphi_\mathbf{v}$ are simple invariant kernels:

$$\varphi(x, y) = \beta \exp(-\alpha(x^2 + y^2)), \tag{25}$$

where β is a scaling coefficient, and α measures the size of the kernel. When $\alpha \to +\infty$, φ converges towards a Dirac measure at 0, so that $f_h(d, \mathbf{v}, h) = \beta_h(h - d)$, as in the standard nudging algorithm.

By linearizing the system around a reference trajectory (leading to the tangent linear model), the error e_h on the water height is then solution of the following damped wave equation:

$$\frac{\partial^2 e_h}{\partial t^2} = g\bar{h}\nabla e_h + \bar{h}\varphi_\mathbf{v} * \nabla e_h - \varphi_h * \frac{\partial e_h}{\partial t}, \tag{26}$$

where $\bar{h}$ is the water height of the reference trajectory.

Using Fourier decompositions of the states, it is then possible to prove that

$$\lim_{t \to +\infty} \left(\|\nabla e_h\|^2 + \left| \frac{\partial e_h}{\partial t} \right|^2 \right) = 0, \tag{27}$$

using the standard L^2 norms [11]. This proves the strong and asymptotic convergence of the error e_h towards 0, and also of the error on the velocity. In other words, the observer solution $(h, \mathbf{v})$ tends to the true state $(h_t, \mathbf{v}_t)$ when time goes to infinity.

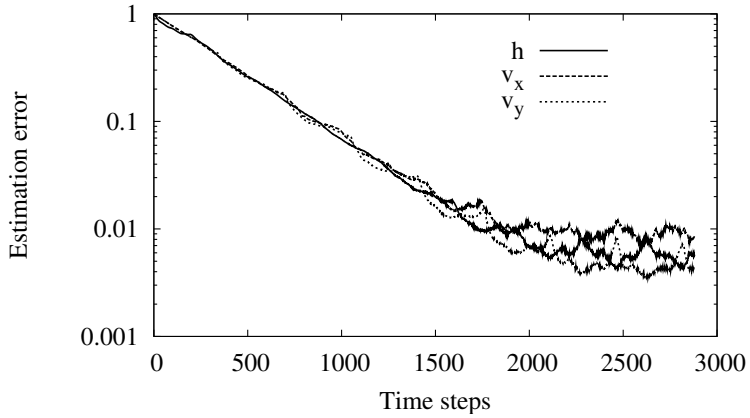

Fig. 3 Evolution of the relative error in logarithmic scale, versus the number of time steps, in the case of noisy observations, for the 3 components of the state: height h, longitudinal velocity $\mathbf{v}_x$ and transversal velocity $\mathbf{v}_y$.

Fig. 3 shows the evolution of the relative error, in logarithmic scale, versus time (expressed as number of time steps here). The observations being noisy (approximately 20% of relative error), the error does not decrease to 0. But the main point is that all variables have the same behaviour, and the error decreases on both height and velocity at the same speed, towards the same level of relative error, while only the height is observed.

We refer the reader to [11] for more details and numerical results on this observer.

2.2 Observer design on a compressible Navier-Stokes model

As another example, we consider here a compressible Navier-Stokes equation, written in conservation form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{28}$$

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} \right) = -\nabla p(\rho) + \mu \Delta \mathbf{v} + (\lambda + \mu) \nabla (\nabla \cdot \mathbf{v}), \tag{29}$$

where the pressure is $p(\rho) = \rho^\gamma$, $\mu > 0$ and $\lambda + \frac{2\mu}{3} \geq 0$. We consider here a n-dimensional square domain with periodic boundary conditions.

We assume that only the velocity is observed, and the goal is to recover both variables, including the (non-observed) density. Note that the symmetric problem of recovering the velocity from the density is completely equivalent from the mathematical point of view, and that all the following results are also valid in this case.

As in the previous subsection, let $\mathbf{d} = \mathbf{H}\mathbf{x}_t = \mathbf{v}_t$ be the observation, the observer system can then be written:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = f_\rho(\rho, \mathbf{v}, \mathbf{d}), \tag{30}$$

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} \right) = -\nabla p(\rho) + \mu \Delta \mathbf{v} + (\lambda + \mu)\nabla(\nabla \cdot \mathbf{v}) + f_\mathbf{v}(\rho, \mathbf{v}, \mathbf{d}). \tag{31}$$

Using preservation of the model physical properties, and convolution with smoothing kernels in order to handle observation noise, the feedback terms are then defined as:

$$f_\rho(\rho, \mathbf{v}, \mathbf{d}) = \varphi_\rho * D_\rho(\mathbf{d} - \mathbf{v}), \qquad f_\mathbf{v}(\rho, \mathbf{v}, \mathbf{d}) = \varphi_\mathbf{v} * D_\mathbf{v}(\mathbf{d} - \mathbf{v}), \tag{32}$$

where D_ρ and $D_\mathbf{v}$ are differential operators. Then, again, we use the simplest operators D, which are here:

$$D_\rho(\mathbf{d} - \mathbf{v}) = \rho_0 \nabla \cdot (\mathbf{d} - \mathbf{v}), \qquad D_\mathbf{v}(\mathbf{d} - \mathbf{v}) = \rho_0(\mathbf{d} - \mathbf{v}), \tag{33}$$

where ρ_0 is the mean reference density. Using (33) and (32) in the tangent linear observer model (derived from (30)-(31)), the observer error equations are then:

$$\frac{\partial e_\rho}{\partial t} + \rho_0 \nabla \cdot \mathbf{e}_\mathbf{v} = -\rho_0 \varphi_\rho * \nabla \cdot \mathbf{e}_\mathbf{v}, \tag{34}$$

$$\rho_0 \frac{\partial \mathbf{e}_\mathbf{v}}{\partial t} = \mu \Delta \mathbf{e}_\mathbf{v} + (\lambda + \mu)\nabla(\nabla \cdot \mathbf{e}_\mathbf{v}) - \gamma \rho_0^{\gamma - 1} \nabla e_\rho - \rho_0 \varphi_\mathbf{v} * \mathbf{e}_\mathbf{v}, \tag{35}$$

with $e_\rho = \rho - \rho_t$ and $\mathbf{e}_\mathbf{v} = \mathbf{v} - \mathbf{v}_t$ the differences between the observer state and the true state.

We can see from Eq. (34) that the feedback term from the ρ equation will then control the divergence of the velocity, and from Eq. (35) that the feedback term from the $\mathbf{v}$ equation will control the velocity equation.

Using a Fourier decomposition of both variables, it is then possible to study the eigenvalues of the system, and to choose for each mode constant Fourier coefficients of the kernels φ_ρ and $\varphi_\mathbf{v}$ in order to control all eigenvalues.

The main result on the tangent linear model is that for any decay rate (specified by the user), we can give an explicit formulation of the kernels φ_ρ and $\varphi_\mathbf{v}$ such that the decay rate of the errors e_ρ and $\mathbf{e}_\mathbf{v}$ towards 0 is at least the specified decay rate [3].

We refer the reader to [3] for the detailed calculations, for the symmetric case where only the density is observed, and for the computation of the decay rate in the case of simple nudging (for a comparison with the full observer).

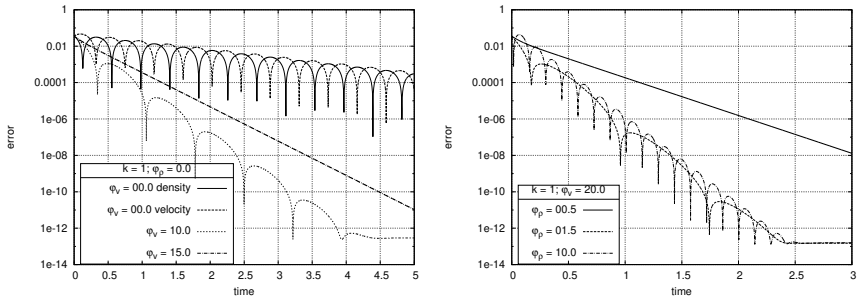

Fig. 4 Norm of the error in logarithmic scale, versus time, for the first Fourier mode, for various values of the convolution kernels: $\varphi_\rho = 0$ on the left figure, and various values of $\varphi_{\mathbf{v}}$ are tested; $\varphi_{\mathbf{v}} = 20$ on the right figure, and various values of φ_ρ are tested.

Fig. 4 shows the evolution of the error (between the observer and the true states) in logarithmic scale, versus time, for various values of the convolution kernels. This experiment only shows what happens for the first Fourier mode. We refer the reader to [3] for more details on the calculations and more numerical results in this framework.

On the left figure, $\varphi_\rho = 0$, so that only the velocity equation is corrected. When there is no feedback on the velocity equation ($\varphi_{\mathbf{v}} = 0$), the error slightly decreases thanks to diffusion. When $\varphi_{\mathbf{v}}$ is increased, the decay rate also increases. This confirms the fact that $\varphi_{\mathbf{v}}$ theoretically controls the decay rate of the error. Note that oscillations are due to the imaginary part of the eigenvalues.

On the right figure, $\varphi_{\mathbf{v}}$ is now fixed to 20, and various values of φ_ρ are tested. When φ_ρ is too small (or equal to 0), the decay rate is decreased, showing the importance to correct both equations. But increasing φ_ρ stops improving the decay rate after some value. Instead, φ_ρ can be used to tune the imaginary part of the eigenvalues, and to remove (or attenuate) the oscillations.

3 Backward and forward observers

Observers usually converge in infinite time (asymptotically), so that in a standard data assimilation framework, where the time window is fixed (and possibly small), there is usually not enough time to benefit from the exponential decrease of the error. It is then necessary to implement a new strategy, allowing one to perform multiple passes on the same assimilation window.

3.1 Backward nudging

The backward nudging algorithm consists in solving the state equations of the model backwards in time, starting from a final state. 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 an initial state of the system [8].

We assume that we have a final condition in the original model given by Eq. (1), instead of an initial condition. This leads to the following backward equation:

$$\frac{d\tilde{\mathbf{x}}}{dt} = \mathbf{m}(\tilde{\mathbf{x}}), \quad T > t > 0. \tag{36}$$

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

$$\frac{d\tilde{\mathbf{x}}}{dt} = \mathbf{m}(\tilde{\mathbf{x}}) - \tilde{\mathbf{k}}(\mathbf{d} - \mathbf{h}(\tilde{\mathbf{x}})), \quad T > t > 0, \tag{37}$$

where $\tilde{\mathbf{k}}$ is the backward feedback. Solving Eq. (37) backwards in time, from $t = T$ to $t = 0$, allows to recover the state at the initial time.

3.2 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 (which is the backward nudging). 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 [8].

The BFN algorithm is the following:

$$\begin{cases} \dfrac{d\mathbf{x}_k}{dt} = \mathbf{m}(\mathbf{x}_k) + \mathbf{k}(\mathbf{d} - \mathbf{h}(\mathbf{x}_k)), & 0 < t < T \\ \mathbf{x}_k(0) = \tilde{\mathbf{x}}_{k-1}(0), \end{cases}$$
$$\begin{cases} \dfrac{d\tilde{\mathbf{x}}_k}{dt} = \mathbf{m}(\tilde{\mathbf{x}}_k) - \tilde{\mathbf{k}}(\mathbf{d} - \mathbf{h}(\tilde{\mathbf{x}}_k)), & T > t > 0 \\ \tilde{\mathbf{x}}_k(T) = \mathbf{x}_k(T), \end{cases} \tag{38}$$

with the notation $\tilde{\mathbf{x}}_{-1}(0) = \mathbf{x}_0$. Then, $\mathbf{x}_0(0) = \mathbf{x}_0$ the initial condition, and a resolution of the direct model gives $\mathbf{x}_0(T)$ and hence $\tilde{\mathbf{x}}_0(T)$. A resolution of the backward model provides then $\tilde{\mathbf{x}}_0(0)$, which defines $\mathbf{x}_1(0)$, and so on.

As we have seen in previous sections, the standard nudging converges asymptotically in time, so that a quite long time window can be necessary to get a strong

decrease of the error. In many situations, this cannot be done as the time window is fixed and it can be relatively small. The idea of the Back and Forth Nudging is then to use this (small) time window to perform back and forth resolutions of the model, with nudging, so that the observations are assimilated multiple times over the window. It is a way to artificially extend the time window to a much larger one.

This algorithm can be compared to the 4D-Var algorithm [17], which also consists in a sequence of forward and backward resolutions. In the BFN algorithm, even for nonlinear problems, there is no need to linearize the system and the backward system is not the adjoint equation but simply the direct system, with an extra feedback term that stabilizes the resolution of the usually ill-posed backward resolution.

The BFN algorithm has been successfully tested for the system of Lorenz equations, Burgers equation and a quasi-geostrophic ocean model in [9], for a shallow-water model in [5] and compared with a variational approach for all these models. It has also been used to assimilate the wind data in a mesoscale model [13], for the reconstruction of quantum states in [20], for the assimilation of wide-swath altimetric data in a quasi-geostrophic ocean model [19, 18], ...

3.3 Theoretical considerations

We first look at a variational interpretation of the nudging: assuming here (only) that the direct model and observation operators are linear, and assuming an implicit discretization of the model with nudging (see Eq. (2)), at a given time step, the nudging equation writes:

$$\frac{\mathbf{x}(t_{j+1}) - \mathbf{x}(t_j)}{\delta t} = \mathbf{M}\mathbf{x}(t_{j+1}) + \mathbf{K}\left(\mathbf{d}(t_{j+1}) - \mathbf{H}\mathbf{x}(t_{j+1})\right), \tag{39}$$

with $\delta t = t_{j+1} - t_j$ the time step. The solution at the new time step $\mathbf{x}(t_{j+1})$ is then the minimum of the following cost function:

$$J(\mathbf{y}) = \frac{1}{2}\|\mathbf{y} - \mathbf{x}(t_j)\|^2 - \frac{1}{2}\delta t\,(\mathbf{M}\mathbf{y})\cdot\mathbf{y} + \frac{1}{2}\delta t\,(\mathbf{C_{dd}}^{-1}(\mathbf{d} - \mathbf{H}\mathbf{y}))\cdot(\mathbf{d} - \mathbf{H}\mathbf{y}) \tag{40}$$

if we choose $\mathbf{K} = \mathbf{H}^T\mathbf{C_{dd}}^{-1}$, $\mathbf{C_{dd}}$ being the covariance matrix of the observation error.

This choice seems indeed reasonable in the sense that $\mathbf{K}$ should be proportional to $\mathbf{C_{dd}}^{-1}$: the larger the observation errors, the smaller the feedback term. Moreover, $\mathbf{K}$ should spread the feedback from the observation space $\mathbb{R}^p$ back to the state space $\mathbb{R}^n$, so that using the transpose of the observation operator is natural.

The two first terms of Eq. (40) correspond to the energy of the system. For instance, if you think at the heat equation, $\mathbf{M}$ is the Laplacian operator, so that $-(\mathbf{M}\mathbf{y})\cdot\mathbf{y} = -(\Delta\mathbf{y})\cdot\mathbf{y} = \|\nabla\mathbf{y}\|^2$. The third term is the observation part of the 4D-Var cost function, that measures the quadratic distance between the model solution and the observations. We can then deduce that direct nudging can be seen as a compromise

between minimizing the energy of the system and minimizing the quadratic distance to the observations. This variational principle gives a way to set the nudging gain matrix:

$$\mathbf{K} = k\mathbf{H}^T{\mathbf{C_{dd}}}^{-1}, \tag{41}$$

with k a positive scalar used to weight the feedback term relatively to the model terms.

For the backward nudging matrix $\tilde{\mathbf{K}}$, we can of course use the linear theory of observability and pole assignment method, as the goal of the backward nudging is to perform a feedback to the observations. But many physical models can be irreversible in time (mainly because of diffusion processes), so that the backward feedback term also has the role of stabilizing the backward resolution of the model. We can then set $\tilde{\mathbf{K}} = \tilde{k}\mathbf{H}^T{\mathbf{C_{dd}}}^{-1}$, as in the forward model, but $\tilde{k}$ has to be large enough to make the backward numerical integration stable.

The convergence of the BFN algorithm has been proved by Auroux and Blum in [8] for linear systems of ordinary differential equations and full observations, by Ramdani et al [25] for reversible linear partial differential equations (wave and Schrödinger equations), by Donovan et al [15] for the reconstruction of quantum states... In [12], 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 [12].

Assuming that we set $\mathbf{K} = \tilde{\mathbf{K}}$, and assuming that the BFN converges, then the forward and backward solutions are equal at the limit, i.e $\tilde{\mathbf{x}}_\infty = \mathbf{x}_\infty$. If we take the sum of the two equations in (38) when the iteration number k goes to infinity, then the limit trajectory $\mathbf{x}_\infty$ satisfies the model equation (1) (including possible model viscosity). Moreover, the difference between the two equations in (38) shows that the limit trajectory is solution of the following equation:

$$\mathbf{K}(\mathbf{d} - \mathbf{H}\mathbf{x}_\infty) = 0. \tag{42}$$

Eq. (42) shows that the limit trajectory perfectly fits the observations (through the observation operator, and the gain matrix). This simple calculation shows that the BFN algorithm over a finite time window $[0;T]$ is very similar to a simple direct nudging algorithm over an infinite time window, allowing then an asymptotic decay of the error when the number of backward-forward iterations goes to infinity (instead of time going to infinity).

3.4 Examples of theoretical and numerical results on transport and fluid models

We give here a couple of examples of the Back and Forth Nudging algorithm applied to geophysical models. We start with a viscous linear transport equation, in 1D, that can be seen as a very simple atmospheric model around the latitude 45°N constant-latitude circle.

The direct model reads:

$$\frac{\partial v}{\partial t} + a(x)\frac{\partial v}{\partial x} = \mu \frac{\partial^2 v}{\partial x^2}, \quad 0 < t < T, \tag{43}$$

with an initial condition $v(t = 0, x) = v_0(x)$, and where $v(t, x)$ is the velocity at time t and space coordinate x, μ is the diffusion coefficient, and $a(x)$ is the (non constant) transport coefficient.

Assuming that the velocity is observed, one iteration of Back and Forth Nudging applied to Eq. (43) consists then in solving the following equations:

$$\frac{\partial v}{\partial t} + a(x)\frac{\partial v}{\partial x} = \mu \frac{\partial^2 v}{\partial x^2} - k(v - d), \quad 0 < t < T, \quad v(t = 0, x) = v_0(x), \tag{44}$$

$$\frac{\partial \tilde{v}}{\partial t} + a(x)\frac{\partial \tilde{v}}{\partial x} = \mu \frac{\partial^2 \tilde{v}}{\partial x^2} + \tilde{k}(\tilde{v} - d), \quad T > t > 0, \quad \tilde{v}(t = T, x) = v(T, x). \tag{45}$$

Note that the feedback terms are added only at observation times and locations. We can then theoretically prove that:

- if the space domain is not fully observed, the back and forth system is ill-posed, in the sense that the solution to the backward equation (45) usually does not exist. But numerically, the algorithm works, as it can be seen on Fig. 5. It is mainly due to the fact that numerically, the spectrum of the diffusion operator is bounded, contrarily to the theoretical unbounded operator;
- if the space domain is fully observed, and if k and $\tilde{k}$ are constant, then the error after one iteration of BFN is decreased by a factor $e^{-(k+\tilde{k})T_{obs}}$, where T_{obs} is the length of the observation period over $[0; T]$ ($T_{obs} = T$ if the velocity is fully observed in time, $T/2$ if the velocity is observed only half of the time over the assimilation window, ...).

After m iterations of BFN, we then get:

$$e_m(0) = e^{-\left[(k+\tilde{k})mT_{obs}\right]} e_0(0), \tag{46}$$

where $e_i(0)$ is the norm of the error on the velocity at time $t = 0$ after i iterations of BFN. We can then observe an exponential decrease of the error thanks to $(k + \tilde{k})$ (but increasing too much the feedback to the observations may not be physical), T_{obs} (but the assimilation window is usually fixed, and the observation period over the assimilation window also), and m (the number of BFN iterations). Using multiple backward and forward resolutions of the equation over the same assimilation window

is then a way to simulate the asymptotic time convergence of the standard Luenberger observer, on a finite time window.

We now consider partial observations in space: only half of the domain is observed. We also consider partial observations in time, so that observations are not available at any time. This allows to tune T_{obs}, the total length of the observation period. Equivalently, we can assume that we observe all the time over the assimilation window, and tune the length of the time window. Note that when observations are not available, the feedback terms are not added to Eqs. (44) and (45), so that the model is used without nudging term.

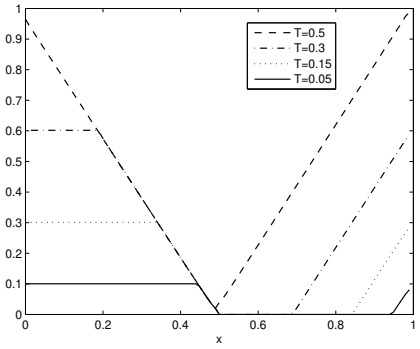

Fig. 5 Decrease rate of the error after one BFN iteration as a function of the space variable, for various lengths of observation.

Fig. 5 shows the decrease rate of the error after one BFN iteration, as a function of the space variable x, and for various T_{obs} (which is equivalent to observe all the time, over a variable time window), in the case of partial observations in space (half of the domain is observed). As it can be seen, for $T < 0.5$, the decay rate is equal to 0 in some part of the domain, so that the error does not decrease: the solution is not corrected. We can prove that $T \geq 0.5$ is also the theoretical observability condition in this case, ensuring that what happens in the non-observed domain will have enough time to travel to the observed domain. We refer the reader to [12] for more theoretical and numerical results on the BFN applied to a linear transport equation, and also to a nonlinear Burgers equation.

The BFN algorithm has also been successfully applied to a shallow water model, with observations of the SSH only, discrete in time and space (in order to mimic satellite observations), with noise, and it has been compared with the 4D-Var [6].

Fig. 6 shows the SSH at the initial time identified by the BFN after 5 iterations (top left), and by the 4D-Var after 5 iterations (top right) and 50 iterations (bottom left). These SSH maps can be compared with the true SSH, represented on the bottom right plot. After only 5 iterations, the BFN almost converged and provides a better solution than the 4D-Var (in the same number of iterations). Of course, using more 4D-Var iterations leads to a better (and smoother) solution. Note that the apparent grid oscillations are due to the sparcity in space of the noisy observations. We refer the reader to [6] for more details.

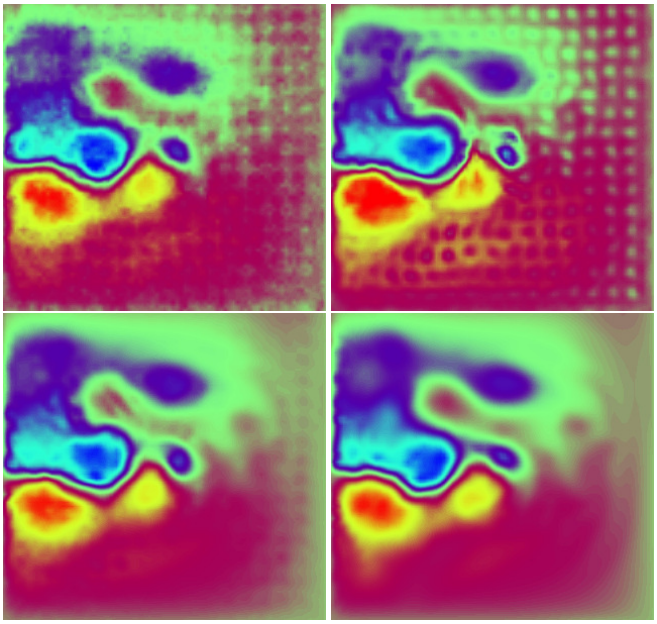

Fig. 6 SSH at the initial time for the shallow water model, after: 5 iterations of BFN (top left), 5 iterations of 4D-Var (top right), 50 iterations of 4D-Var (bottom left); true SSH state (bottom right).

Finally, as another example, the BFN has also been used on a realistic quasi-geostrophic ocean model, with SWOT-like observations (SWOT - Surface Water Ocean Topography - is a satellite mission that has been launched end of 2022) providing SSH with a swath wide of 120km and repeat period of 21 days.

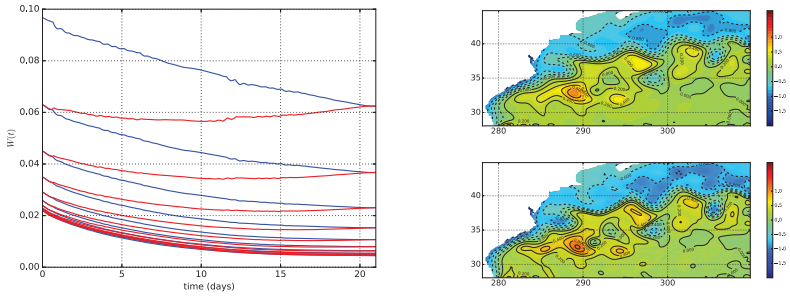

Fig. 7 Decrease of the error versus time and BFN iterations (forward nudging resolutions in blue, backward nudging resolutions in red) (left); SSH at the end of the assimilation window (right): after 10 BFN iterations (top), and true SSH (bottom).

Fig. 7 shows the results of the BFN applied to the quasi-geostrophic ocean model using realistic SWOT-like data (including expected noise). On the left figure, one can see the decrease of the error versus time, during the back and forth resolutions. The forward nudging resolutions are in blue, the backward nudging resolutions are in red. On the right side, one can see the SSH at the end of the assimilation window, after 10 BFN iterations (top), to be compared with the true SSH (bottom). We refer the reader to [2] for more details.

Note also that the backward-forward framework can be extended to any type of observers, and not only nudging or its extensions where non-observed variables are corrected by the observed ones (as shown in previous subsections). One can for instance think at BF-Kalman algorithms, using full or reduced-order Kalman filters.

3.5 Diffusive Back and Forth Nudging algorithm

Another example of backward-forward observer is a more recent extension of the BFN algorithm. In the framework of oceanographic and meteorological problems, there is usually no diffusion in the physical process. 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 differential equation reads:

$$\frac{d\mathbf{x}}{dt} = \mathbf{m}(\mathbf{x}) + \mu\Delta\mathbf{x}, \quad 0 < t < T, \tag{47}$$

where we assume that $\mathbf{m}$ has no diffusive terms, μ is the diffusion coefficient, and we assume that the diffusion is here a standard second-order Laplacian (note that it could be a fourth or sixth order derivative as in some oceanographic QG models).

The idea of the D-BFN (Diffusive BFN) algorithm is to solve forward and backward the model $\mathbf{m}$, and to use alternating signs for both the feedback term (as in the standard BFN) and the diffusion term. The D-BFN reads:

$$\frac{d\mathbf{x}_k}{dt} = \mathbf{m}(\mathbf{x}_k) + \mu\Delta\mathbf{x}_k + \mathbf{k}(\mathbf{d} - \mathbf{h}(\mathbf{x}_k)), \quad 0 < t < T, \quad \mathbf{x}_k(0) = \tilde{\mathbf{x}}_{k-1}(0), \tag{48}$$

$$\frac{d\tilde{\mathbf{x}}_k}{dt} = \mathbf{m}(\tilde{\mathbf{x}}_k) - \mu\Delta\tilde{\mathbf{x}}_k - \tilde{\mathbf{k}}(\mathbf{d} - \mathbf{h}(\tilde{\mathbf{x}}_k)), \quad T > t > 0, \quad \tilde{\mathbf{x}}_k(T) = \mathbf{x}_k(T). \tag{49}$$

It is straightforward to see that the backward equation (49) can be rewritten, using $t' = T - t$, as:

$$\frac{d\tilde{\mathbf{x}}_k}{dt'} = -\mathbf{m}(\tilde{\mathbf{x}}_k) + \mu\Delta\tilde{\mathbf{x}}_k + \tilde{\mathbf{k}}(\mathbf{d} - \mathbf{h}(\tilde{\mathbf{x}}_k)), \quad 0 < t < T, \quad \tilde{\mathbf{x}}_k(t' = 0) = \mathbf{x}_k(T), \tag{50}$$

where $\tilde{\mathbf{x}}$ is evaluated at time t'. Then the backward equation is well-posed, solved forwards in time with an initial condition, and with the same diffusion operator

as in the forward equation (48). Then the diffusion term both takes into account the subscale processes and stabilizes the numerical backward integration, and the feedback term still controls the trajectory with the observations. Note that in this case, it is usually possible to set $\tilde{\mathbf{k}} = \mathbf{k}$ as only the non-diffusive terms are now reversed in the model equation.

In a similar way as in Section 3.3, we assume here the convergence of the D-BFN algorithm in a linear situation. Then, if we take the sum of Eqs. (48) and (49) at the limit when the number of iterations goes to infinity, we see that the limit trajectory $\mathbf{x}_\infty$ is now a solution of the model equation without diffusion. And the difference between Eqs. (48) and (49) shows that the limit trajectory satisfies the Poisson equation:

$$-\Delta \mathbf{x}_\infty = \frac{1}{\mu}\mathbf{K}(\mathbf{d} - \mathbf{H}\mathbf{x}_\infty) \tag{51}$$

which represents a well-known smoothing process on the observations, for which the degree of smoothness is given by μ divided by the order of magnitude of $\mathbf{K}$ [10]. Eq. (51) indeed corresponds to the Euler equation of the minimization of the following cost function

$$J(\mathbf{x}) = k\left(\mathbf{C_{dd}}^{-1}(\mathbf{d} - \mathbf{H}\mathbf{x})\right) \cdot (\mathbf{d} - \mathbf{H}\mathbf{x}) + \mu\|\nabla \mathbf{x}\|^2 \tag{52}$$

if we set $\mathbf{K} = k\mathbf{H}^T\mathbf{C_{dd}}^{-1}$ as in Section 3.3. The first term represents the quadratic distance to the observations (as in the 4D-Var cost function) and the second one is a first order Tikhonov regularisation term over the domain of resolution. This is a nice increment to the BFN algorithm (in which the limit trajectory fits the observations, see Eq. (42)), as in the DBFN algorithm, the limit trajectory is the result of a smoothing process on the observations (which are often very noisy).

The D-BFN algorithm has been successfully tested on a linear transport equation in [10] and on a non-linear Burgers equation in [7]. It has also been used for the full primitive ocean model NEMO [26].

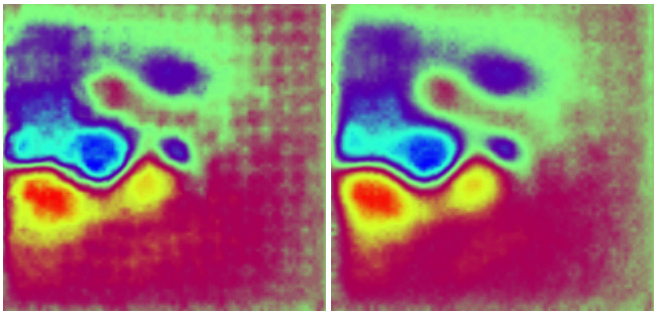

Fig. 8 SSH at the initial time for the shallow water model, after 5 iterations of BFN (left) and 5 iterations of D-BFN (right).

We just give in Fig. 8 a numerical example on a shallow-water model, in order to see the improvement between the BFN and D-BFN algorithms. This figure complements Fig. 6, with the result of the D-BFN (see Section 3.4 for more details about the experiment). The solution is smoother, thanks to the diffusion process, and much closer to the true state (see Fig. 6) even with sparse and noisy observations.

4 New developments: observers for parameter estimation

Among the new developments of observers (in the forward only, or backward-forward framework), we can cite the identification of model parameters, jointly with the model state.

4.1 Ad-hoc parameter equation

Variational approaches are often used for model calibration and/or parameter estimation, as it can also quite easily identify the parameters (and not only the state), thanks to the same adjoint model. But observers have been recently developed in order to also identify the parameters. Using the idea developed in [23, 1], we can use sequential data assimilation methods (like Kalman filters, simple nudging, ...) for the identification of model parameters by simply adding an ad-hoc parameter equation. We consider the following coupled model:

$$\frac{d\mathbf{x}}{dt} = \mathbf{m}(\mathbf{x}, \mathbf{c}), \quad 0 < t < T, \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{53}$$

$$\frac{d\mathbf{c}}{dt} = 0, \quad 0 < t < T, \quad \mathbf{c}(0) = \mathbf{c}_0, \tag{54}$$

where $\mathbf{c}$ represents the model parameter(s). The first equation (53) is the standard model equation, and we add an ad-hoc equation (54) for the parameter, $\mathbf{c}_0$ being an a priori estimation of the parameter.

It is then straightforward, on the paper, to apply the nudging, or BFN, or other observer, to this coupled system, assuming the state $\mathbf{x}$ is observed, with the aim of recovering both the state and the parameter. In practice, the observer design will clearly depend on the model equation, as the keypoint is to find how to control or correct the parameter equation with the innovation vector on the state. The general and theoretical idea is to define a Lyapunov function, enabling us to obtain the expression of the nudging term to be added to the parameter equation in order to ensure the exponential decrease of the error [15].

4.2 Identification of the transport velocity

As an example, we consider here an inviscid linear transport equation. The model is the same as Eq. (43), with $\mu = 0$ (no viscosity). The parameter to be estimated is the transport coefficient $\mathbf{c} = a$, and the state is the velocity $\mathbf{x} = v$. Applying nudging to the coupled state-parameter model gives the P-BFN (Parameter BFN) algorithm:

$$\frac{\partial v}{\partial t} + a(x)\frac{\partial v}{\partial x} = k_v(d - v), \quad 0 < t < T, \quad v(t = 0, x) = v_0(x), \tag{55}$$

$$\frac{\partial a}{\partial t} = k_a \mathcal{F}(d - v), \quad 0 < t < T, \quad a(t = 0, x) = a_0(x), \tag{56}$$

where $\mathcal{F}$ is a feedback function involving spatial differential operators, such that there exists a Lyapunov function which exponentially decreases in time, leading to the convergence of both the state and parameter observers. The idea is similar to the work presented in Section 2.1: eliminate the parameter equation and work only on the velocity equation, and ensure that all terms are controlled.

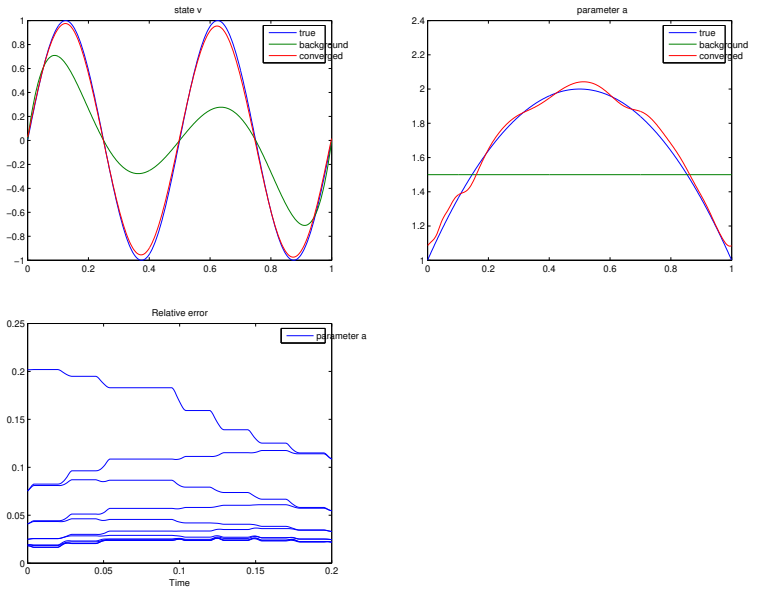

Fig. 9 Initial guess (green), true solution (blue), and P-BFN identified solution (red) for the state (top left) and parameter (top right); Decrease of the error versus time and the P-BFN iterations (bottom).

Fig. 9 shows the initial guess, the true solution and the solution identified by the P-BFN algorithm after 5 iterations, for both the parameter (left) and state (right). The bottom figure shows the decrease of the Lyapunov function during the back and forth

iterations. Note that in this simulation, the observations are only available at some subperiods in time over the time window $[0; 0.2]$ (and also only on some part of the space domain). This explains all the plateaux in the bottom figure: no observations are available at that times, so that no feedback terms are added to the equations, and thus the error does not decrease. The idea is similar to Kalman filters: where there are no available observations, only the model is used, while when observations are available, some correction is done. We can see that the parameter is very well recovered after just a few iterations.

4.3 QG model with SWOT-like data

As another example, the P-BFN algorithm has been tested on a quasi-geostrophic ocean model in the framework of SWOT data (see Section 3.4 and [2] for more details). We consider here that the phase speed c, or the barotropic deformation wavenumber $\kappa = \frac{f^2}{c^2}$ (f is the Coriolis parameter), is unknown (see e.g. [14]). We then add an ad hoc equation to the QG model, claiming that the time derivative of c is equal to 0. And we apply the P-BFN algorithm to the coupled system in order to identify both the state and the parameter.

Fig. 10 shows on the top the result of BFN iterations when the parameter is fixed to the exact value (left) and to a wrong value (right). The forward (blue) and backward (red) iterations are done on the time period $[0; 20]$ days, and then the forecast period is $[20; 85]$ days. Using the exact parameter, as shown in Section 3.4, the SSH is very well identified and the forecast remains excellent. Using a wrong parameter, it is still possible to reasonably identify the state, but once the assimilation window is finished, the forecast uses a wrong model parameter, so that the SSH error increases quite quickly.

Using now the P-BFN algorithm to jointly identify the state and parameter (see bottom left figure), the SSH is quickly corrected, and the parameter is also well identified, so that the SSH is small at the end of the assimilation window (almost as in the top left figure), and it remains small as well during the forecast. Finally, the bottom right figure shows the decrease of the Lyapunov function describing the parameter error during the P-BFN iterations. As with the standard BFN, the P-BFN only requires a few iterations to converge, and identify here the parameter and the state.

Several developments are now in progress with (forward-backward) observers in this framework, mainly with the idea of identifying the full state of a system from a very partial knowledge, sometimes using indirect variables like passive tracers in a flow.

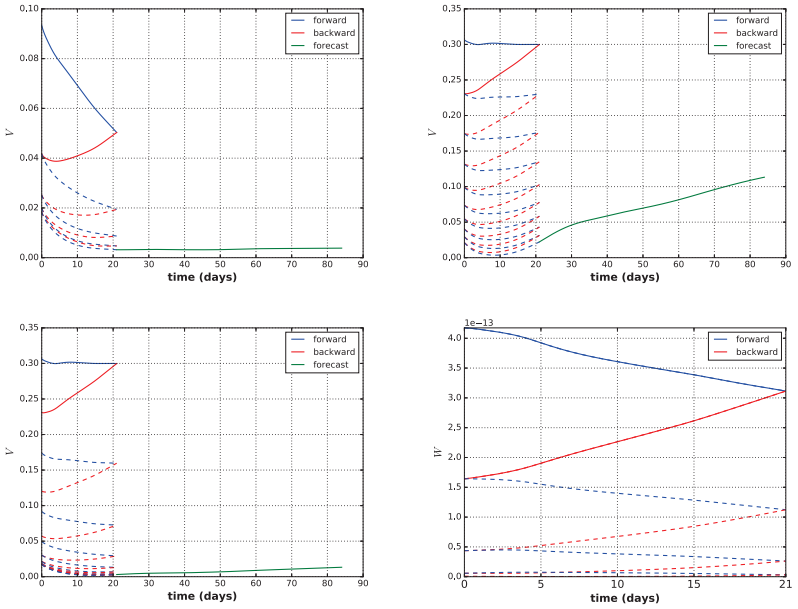

Fig. 10 SSH error versus time using BFN with the true parameter (top left), BFN with a fixed wrong parameter (top right), and P-BFN for jointly estimating the state and parameter (bottom left); Lyapunov function describing the parameter error versus time during the P-BFN iterations (bottom right).

References

1. C. Afri, V. Andrieu, L. Bako, and P. Dufour. State and parameter estimation: A nonlinear luenberger observer approach. *IEEE Trans. Automat. Control*, 62:973–980, 2017.
2. S. Amraoui, D. Auroux, J. Blum, and E. Cosme. Back-and-Forth Nudging for the quasi-geostrophic ocean dynamics with altimetry: theoretical convergence study and numerical experiments with the future SWOT observations. *Discrete & Continuous Dynamical Systems - S*, 2022. http://dx.doi.org/10.3934/dcdss.2022058.
3. A. Apte, D. Auroux, and M. Ramaswamy. Observers for compressible Navier-Stokes equation. *SIAM J. Control Optim.*, 56(2):1081–1104, 2018.
4. M. Arnold and B. N. Datta. Single-input eigenvalue assignment algorithms: A close look. *SIAM J. Matrix Anal. Appl.*, 19(2):444–467, 1998.
5. D. Auroux. 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(8):911–929, 2009.
6. D. Auroux. 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(8):911–929, 2009.
7. D. Auroux, P. Bansart, and J. Blum. An evolution of the Back and Forth Nudging for geophysical data assimilation: application to burgers equation and comparisons. *Inv. Prob. Sci. Eng.*, 21(3):399–419, 2013.
8. D. Auroux and J. Blum. Back and forth nudging algorithm for data assimilation problems. *C. R. Acad. Sci. Paris, Ser. I*, 340:873–878, 2005.

9. D. Auroux and J. Blum. A nudging-based data assimilation method for oceanographic problems: the Back and Forth Nudging (BFN) algorithm. *Nonlin. Proc. Geophys.*, 15:305–319, 2008.
10. D. Auroux, J. Blum, and M. Nodet. Diffusive Back and Forth Nudging algorithm for data assimilation. *C. R. Acad. Sci. Paris, Ser. I*, 349(15-16):849–854, 2011.
11. D. Auroux and S. Bonnabel. Symmetry-based observers for some water-tank problems. *IEEE Trans. Automat. Contr.*, 56(5):1046–1058, 2011.
12. D. Auroux and M. Nodet. The Back and Forth Nudging algorithm for data assimilation problems: theoretical results on transport equations. *ESAIM Control Optim. Calc. Var.*, 18(2):318–342, 2012.
13. A. Boilley and J.-F. Mahfouf. Assimilation of low-level wind in a high resolution mesoscale model using the back and forth nudging algorithm. *Tellus A*, 64:18697, 2012.
14. D. B. Chelton, R. A. Deszoeke, M. G. Schlax, K. El Naggar, and N. Siwertz. Geographical variability of the first baroclinic Rossby radius of deformation. *J. Phys. Ocean.*, 28:433–460, 1998.
15. A. Donovan, M. Mirrahimi, and P. Rouchon. Back and Forth Nudging for quantum state reconstruction. In *4th Int. Symp. Communications Control Signal Proc.*, pages 1–5, 2010.
16. J. Hoke and R. A. Anthes. The initialization of numerical models by a dynamic initialization technique. *Month. Weaver Rev.*, 104:1551–1556, 1976.
17. F.-X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus*, 38A:97–110, 1986.
18. F. Le Guillou, N. Lahaye, C. Ubelmann, S. Metref, E. Cosme, and A. Ponte. Joint estimation of balanced motions and internal tides from future wide-swath altimetry. *J. Adv. Model. Earth Syst.*, 13(12), 2021. https://doi.org/10.1029/2021MS002613.
19. F. Le Guillou, S. Metref, E. Cosme, C. Ubelmann, M. Ballarotta, J. Le Sommer, and J. Verron. Mapping altimetry in the forthcoming SWOT era by Back-and-Forth Nudging a one-layer quasigeostrophic model. *J. Atmos. Ocean. Tech.*, 38(4):697–710, 2021.
20. Z. Leghtas, M. Mirrahimi, and P. Rouchon. Observer-based quantum state estimation by continuous weak measurement. In *American Control Conference (ACC)*, pages 4334–4339, 2011.
21. E. N. Lorenz. Deterministic non periodic flow. *J. Atmos. Sci.*, 20:130–141, 1963.
22. D. Luenberger. Observers for multivariable systems. *IEEE Trans. Autom. Contr.*, 11:190–197, 1966.
23. P. Moireau and D. Chapelle. Reduced-order Unscented Kalman Filtering with application to parameter identification in large-dimensional systems. *ESAIM: Control, Optimisation and Calculus of Variations*, 17(2):380–405, 2011.
24. J. Pedlosky. *Geophysical fluid dynamics*. Springer-Verlag, New-York, 1979.
25. K. Ramdani, M. Tucsnak, and G. Weiss. Recovering the initial state of an infinite-dimensional system using observers. *Automatica*, 46(10):1616–1625, 2010.
26. G. A. Ruggiero, Y. Ourmières, E. Cosme, J. Blum, D. Auroux, and J. Verron. Data assimilation experiments using the diffusive back and forth nudging for the nemo ocean model. *Nonlin. Proc. Geophys.*, 22:233–248, 2015.
27. D. R. Stauffer and J. W. Bao. Optimal determination of nudging coefficients using the adjoint equations. *Tellus A*, 45:358–369, 1993.
28. D. R. Stauffer 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.
29. J. Verron 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.
30. P. A. Vidard, F.-X. L. Dimet, and A. Piacentini. Determination of optimal nudging coefficients. *Tellus A*, 55:1–15, 2003.
31. X. Zou, 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, 1992.

Index