Didier Auroux
University of Nice Sophia Antipolis
France
auroux@unice.fr

Nudging methods in geophysical data assimilation
Part 3 : Back and forth nudging

Mathematical and Statistical Foundations of Data Assimilation,
TIFR-CAM, Bangalore, India.
July 04-23, 2011

Talk overview

1. Back and forth nudging algorithm

2. Numerical experiments : shallow water, QG model, NEMO model

3. Extension to back and forth Kalman filters

4. Diffusive back and forth nudging

Talk overview

⇒ 1. Back and forth nudging algorithm

2. Numerical experiments : shallow water, QG model, NEMO model

3. Extension to back and forth Kalman filters

4. Diffusive back and forth nudging

Forward nudging

$$\begin{cases} \dfrac{dX}{dt} = F(X) + K(Y_{obs} - H(X)), \quad 0 < t < T, \\ X(0) = X_0, \end{cases}$$

where K is the nudging (or gain) matrix.

In the linear case (where F is a matrix), the forward nudging is called Luenberger or asymptotic observer.

- Meteorology : Hoke-Anthes (1976)
- Oceanography (QG model) : De Mey et al. (1987), Verron-Holland (1989)
- Atmosphere (meso-scale) : Stauffer-Seaman (1990)
- Optimal determination of the nudging coefficients :

  Zou-Navon-Le Dimet (1992), Stauffer-Bao (1993),

  Vidard-Le Dimet-Piacentini (2003)

Forward nudging : linear case

Luenberger observer, or asymptotic observer

(Luenberger, 1966)

$$\begin{cases} \dfrac{dX_{true}}{dt} = FX_{true}, \quad Y_{obs} = HX_{true}, \\ \dfrac{dX}{dt} = FX + K(Y_{obs} - HX). \end{cases}$$

$$\frac{d}{dt}(X - X_{true}) = (F - KH)(X - X_{true})$$

If $F - KH$ is a Hurwitz matrix, i.e. its spectrum is strictly included in the half-plane $\{\lambda \in \mathbb{C}; Re(\lambda) < 0\}$, then $X \to X_{true}$ when $t \to +\infty$.

Backward nudging

How to recover the initial state from the final solution ?

Backward model :

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

If we apply nudging to this backward model :

$$\begin{cases} \dfrac{d\tilde{X}}{dt} = F(\tilde{X}) - K(Y_{obs} - H\tilde{X}), \quad T > t > 0, \\ \tilde{X}(T) = \tilde{X}_T. \end{cases}$$

BFN : Back and Forth Nudging algorithm

Iterative algorithm (forward and backward resolutions) :

$$\tilde{X}_0(0) = X_b \text{ (first guess)}$$

$$\begin{cases} \dfrac{dX_k}{dt} = F(X_k) + K(Y_{obs} - H(X_k)) \\ X_k(0) = \tilde{X}_{k-1}(0) \end{cases}$$

$$\begin{cases} \dfrac{d\tilde{X}_k}{dt} = F(\tilde{X}_k) - K'(Y_{obs} - H(\tilde{X}_k)) \\ \tilde{X}_k(T) = X_k(T) \end{cases}$$

[Auroux - Blum, C. R. Acad. Sci. Math. 2005]

If X_k and $\tilde{X}_k$ converge towards the same limit X, and if $K = K'$, then X satisfies the state equation and fits to the observations.

Choice of the direct nudging matrix K

Implicit discretization of the direct model equation with nudging :

$$\frac{X^{n+1} - X^n}{\Delta t} = FX^{n+1} + K(Y_{obs} - HX^{n+1}).$$

Variational interpretation : direct nudging is a compromise between the minimization of the energy of the system and the quadratic distance to the observations :

$$\min_X \left[\frac{1}{2}\langle X - X^n, X - X^n \rangle - \frac{\Delta t}{2}\langle FX, X \rangle + \frac{\Delta t}{2}\langle R^{-1}(Y_{obs} - HX), Y_{obs} - HX \rangle \right],$$

by chosing

$$K = kH^T R^{-1}$$

where R is the covariance matrix of the errors of observation, and k is a scalar.

[Auroux-Blum, Nonlin. Proc. Geophys. 2008]

Choice of the backward nudging matrix K'

The feedback term has a double role :

- stabilization of the backward resolution of the model (irreversible system)
- feedback to the observations

If the system is observable, i.e. $rank[H, HF, \dots, HF^{N-1}] = N$, then there exists a matrix K' such that $-F - K'H$ is a Hurwitz matrix (pole assignment method).

Simpler solution : one can define $K' = k'H^T R^{-1}$, where k' is e.g. the smallest value making the backward numerical integration stable.

Example of convergence results

Viscous linear transport equation :

$$\begin{cases} \partial_t u - \nu \partial_{xx} u + a(x)\partial_x u = -K(u - u_{obs}), & u(x, t = 0) = u_0(x) \\ \partial_t \tilde{u} - \nu \partial_{xx} \tilde{u} + a(x)\partial_x \tilde{u} = K'(\tilde{u} - u_{obs}), & \tilde{u}(x, t = T) = u_T(x) \end{cases}$$

We set $w(t) = u(t) - u_{obs}(t)$ and $\tilde{w}(t) = \tilde{u}(t) - u_{obs}(t)$ the errors.

• If K and K' are constant, then $\forall t \in [0, T]$: $\widetilde{w}(t) = e^{(-K-K')(T-t)} w(t)$

(still true if the observation period does not cover $[0, T]$)

• If the domain is not fully observed, then the problem is ill-posed.

Error after k iterations : $w_k(0) = e^{-[(K+K')kT]} w_0(0)$

$\rightsquigarrow$ exponential decrease of the error, thanks to :

- $K + K'$: infinite feedback to the observations (not physical)
- T : asymptotic observer (Luenberger)
- k : infinite number of iterations (BFN)

[Auroux-Nodet, COCV 2011]

Observability condition

Consider the inviscid linear transport equation :

$$\partial_t u + a(x)\partial_x u = -K(u - u_{obs}),$$

with $K = K' = \mathbb{1}_{[0;0.5]}(x)$, $a(x) = 1$, $u_{true} = 0$, and $u_0(x) = \sin(2\pi x)$. Various times T are considered, from 0.05 to 1.

Half of the domain is observed, and as the speed is $a = 1$, the solution needs at least $T = 0.5$ in order to fully intersect the observation zone.

Example run : one back and forth iteration for linear transport.

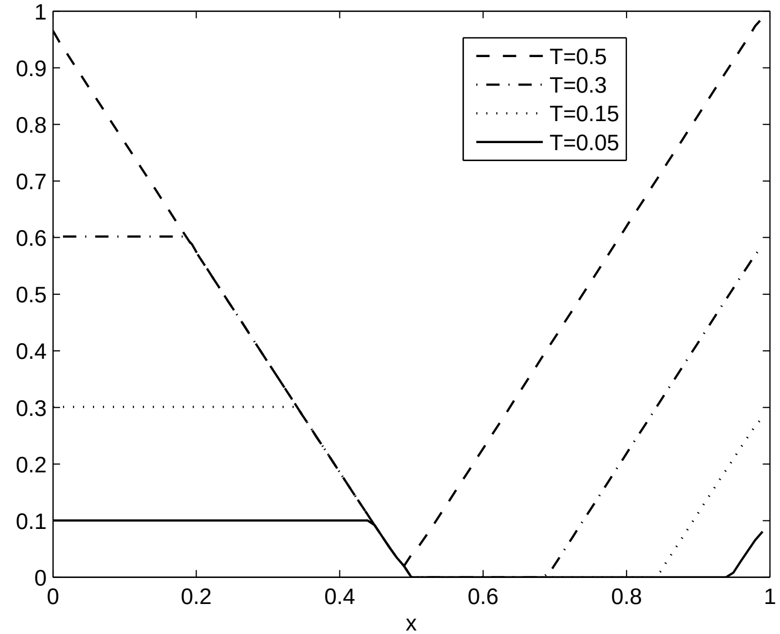
Example run : one back and forth iteration for Burgers equation.

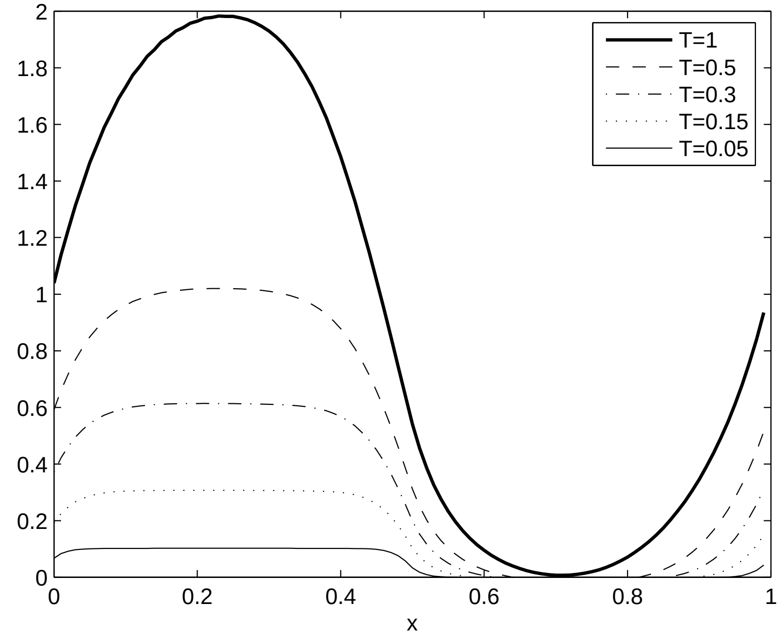
Université Nice Sophia Antipolis Laboratoire J. A. Dieudonné

Observability condition

Let $\chi(x)$ be the time during which the characteristic curve with foot x lies in the support of K. Then the system is observable if and only if $\min_x \chi(x) > 0$.

Partial observations in space : half of the domain is observed.





Decrease rate of the error after one iteration of BFN as a function of the space variable x, for various final times T.

Linear case (left) : theoretical observability condition $= T > 0.5$

Nonlinear case (right) : numerical observability condition $= T > 1$

Talk overview

1. Back and forth nudging algorithm

⇒ 2. Numerical experiments : shallow water, QG model, NEMO model

3. Extension to back and forth Kalman filters

4. Diffusive back and forth nudging

Shallow water model

$$\begin{aligned}
\partial_t u - (f+\zeta)v + \partial_x B &= \frac{\tau_x}{\rho_0 h} - ru + \nu\Delta u \\
\partial_t v + (f+\zeta)u + \partial_y B &= \frac{\tau_y}{\rho_0 h} - rv + \nu\Delta v \\
\partial_t h + \partial_x(hu) + \partial_y(hv) &= 0
\end{aligned}$$

- $\zeta = \partial_x v - \partial_y u$ is the relative vorticity ;
- $B = g^* h + \frac{1}{2}(u^2+v^2)$ is the Bernoulli potential ;
- $g^* = 0.02\ m.s^{-2}$ is the reduced gravity ;
- $f = f_0 + \beta y$ is the Coriolis parameter (in the β-plane approximation), with $f_0 = 7.10^{-5}\ s^{-1}$ and $\beta = 2.10^{-11}\ m^{-1}.s^{-1}$;
- $\tau = (\tau_x, \tau_y)$ is the forcing term of the model (e.g. the wind stress), with a maximum amplitude of $\tau_0 = 0.05\ s^{-2}$;
- $\rho_0 = 10^3\ kg.m^{-3}$ is the water density ;
- $r = 9.10^{-8}\ s^{-1}$ is the friction coefficient.
- $\nu = 5\ m^2.s^{-1}$ is the viscosity (or dissipation) coefficient.

Shallow water model

2D shallow water model, state = height h and horizontal velocity (u, v)

Numerical parameters : (run example)

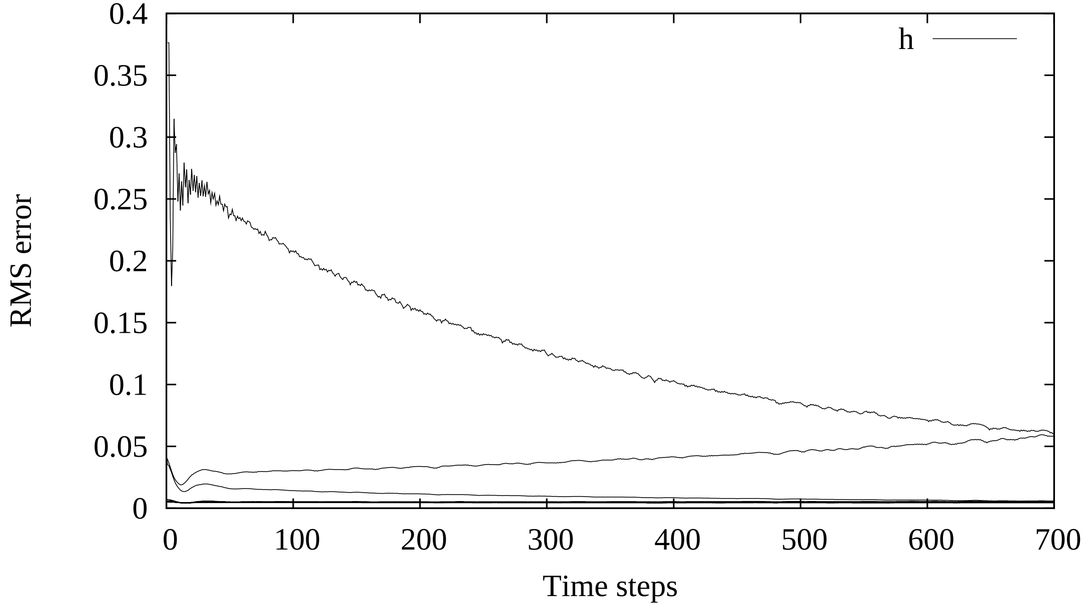
Domain : $L = 2000$ km $\times$ 2000 km ; Rigid boundary and no-slip BC ; Time step = 1800 s ; Assimilation period : 15 days ; Forecast period : 15 + 45 days

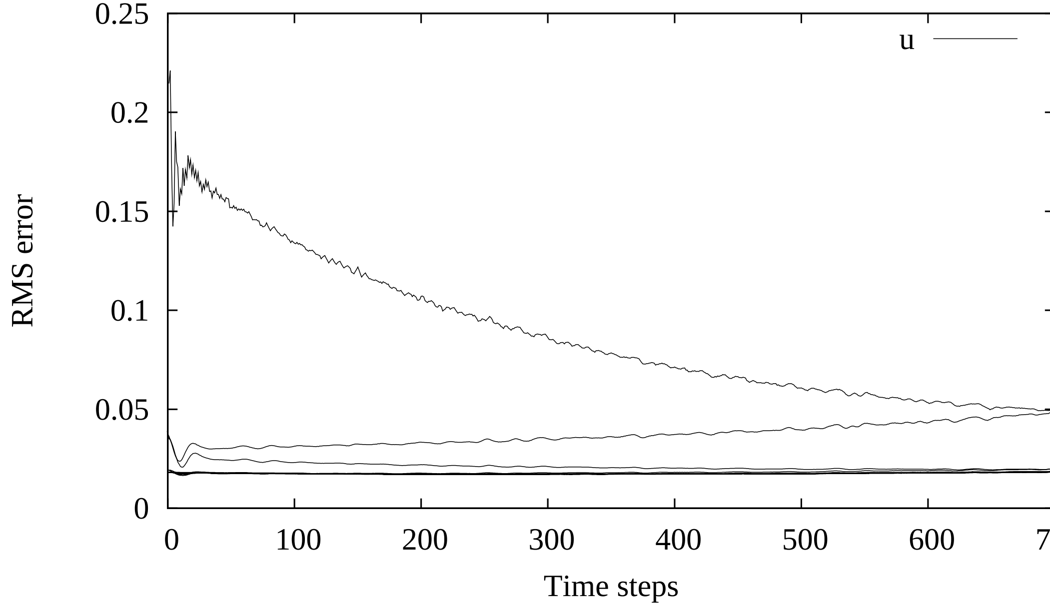
Observations : of h only ($\sim$ satellite obs), every 5 gridpoints in each space direction, every 24 hours.

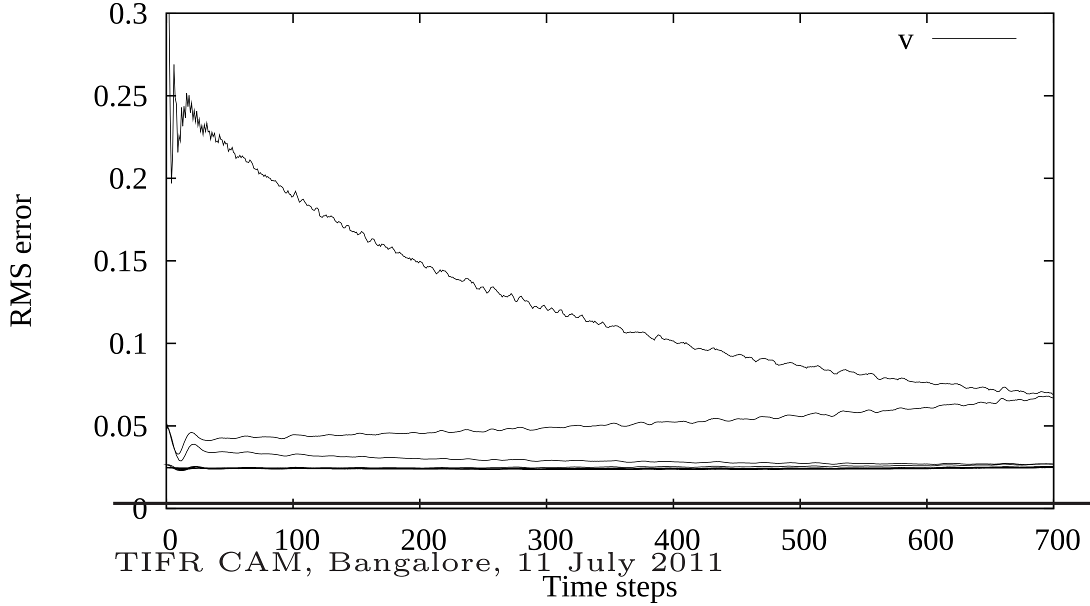
Background : true state one month before the beginning of the assimilation period + white gaussian noise ($\sim$ 10%)

Comparison BFN - 4DVAR : sea height h ; velocity :u and v.

Convergence - perfect obs.







Relative difference between the BFN iterates (5 first iterations) and the true solution versus the time steps, for h, u and v.

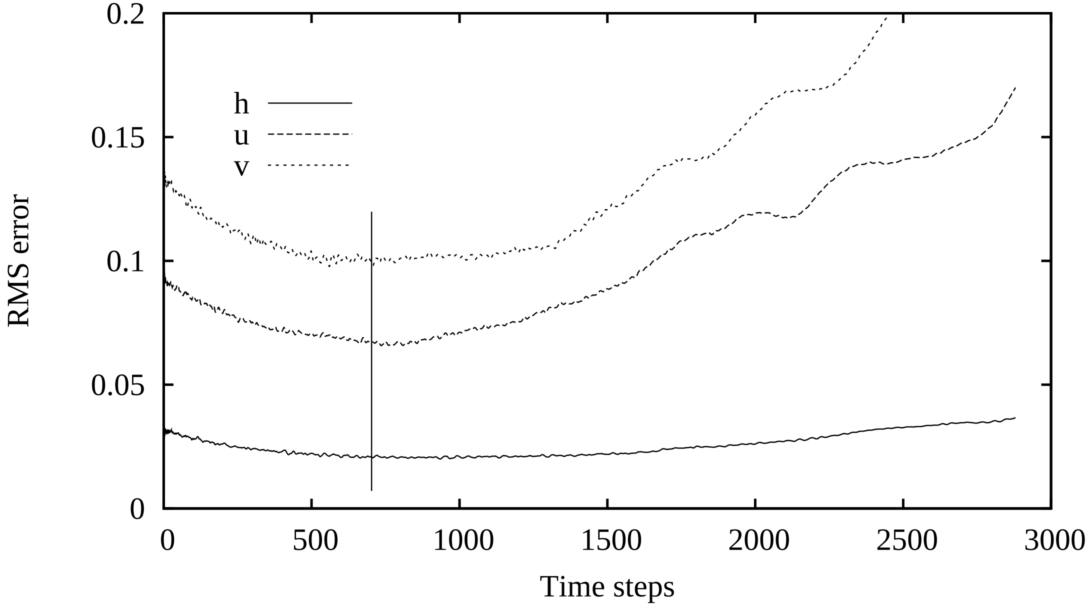
[Auroux, Int. J. Num. Meth. Fluids 2009]

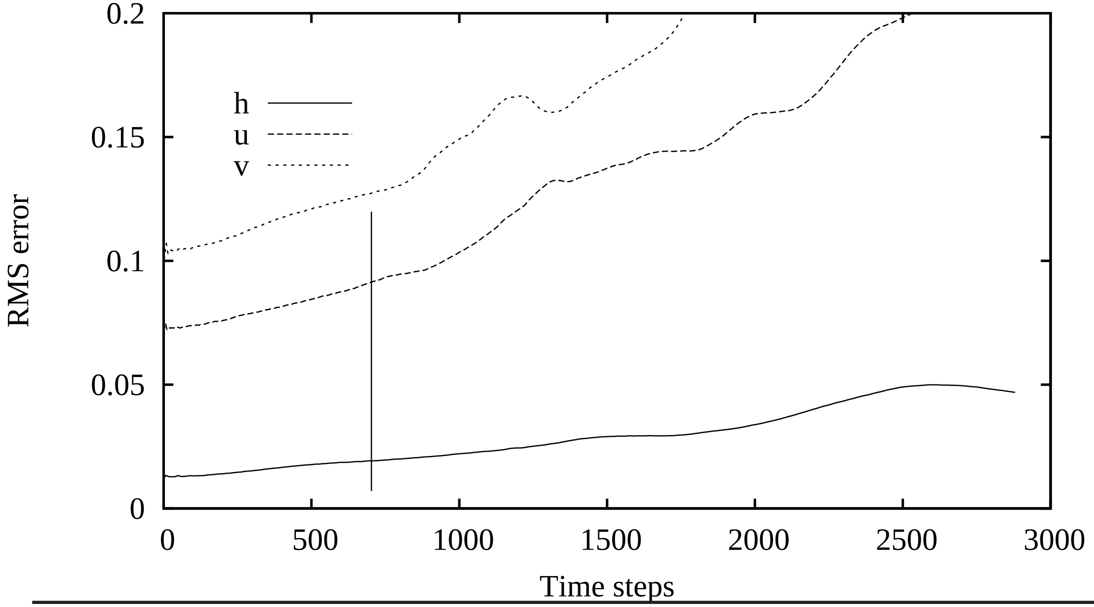
Comparaison - perfect obs.

Relative error	h	u	v
Background state	37.6%	21.5%	30.3%
BFN (5 iterations, converged)	0.44%	1.78%	2.41%
4D-VAR (5 iterations)	0.64%	3.14%	4.47%
4D-VAR (18 iterations, converged)	0.61%	2.43%	3.46%

Relative error of the background state and various identified initial conditions for the three variables.

Noisy observations

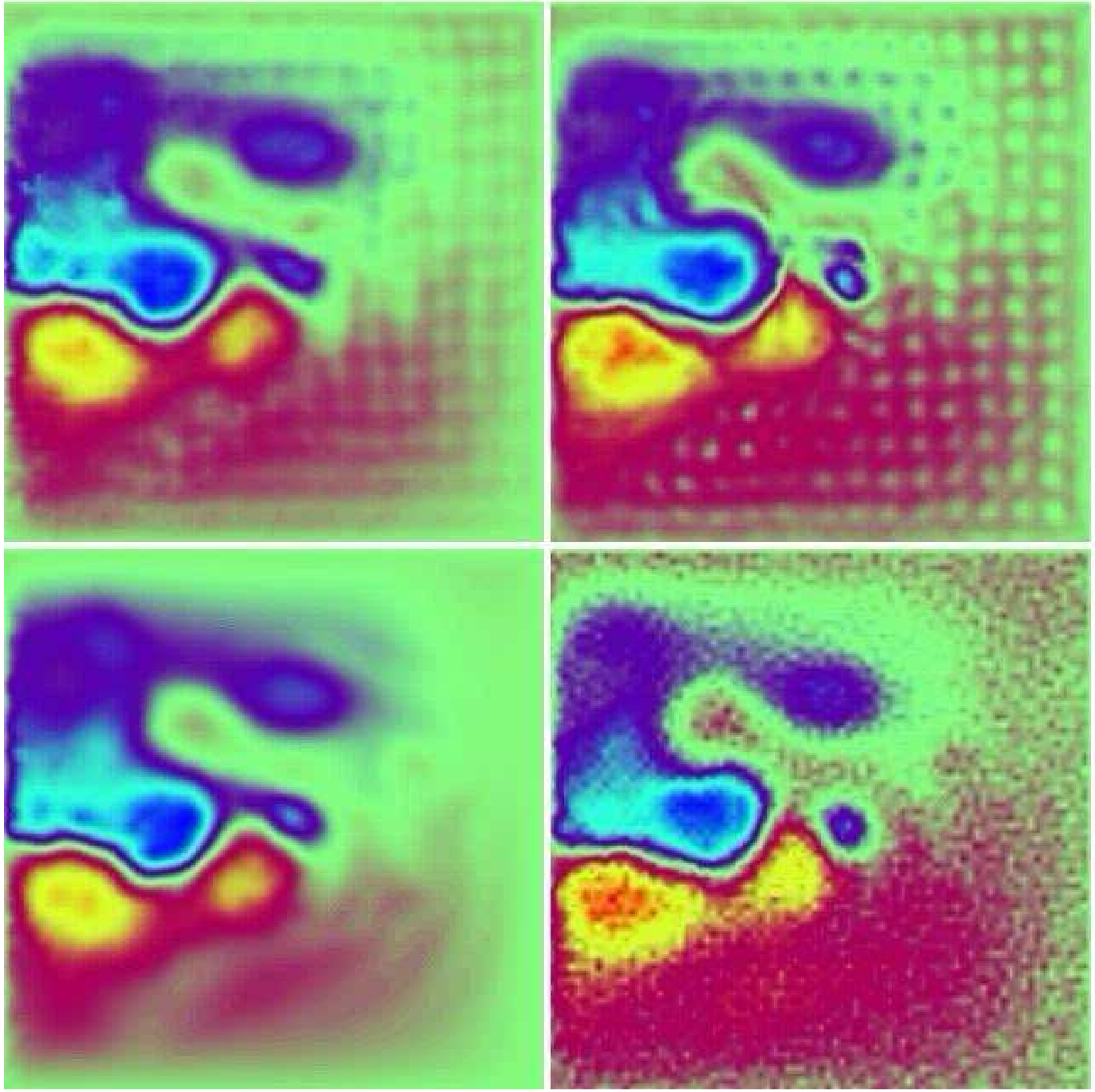




Relative difference between the true solution and the forecast trajectory corresponding to the BFN (top) and 4D-VAR (bottom) identified initial conditions at convergence, versus time, for the three variables and in the case of noisy observations.

Université Nice Sophia Antipolis
Laboratoire J. A. Dieudonné

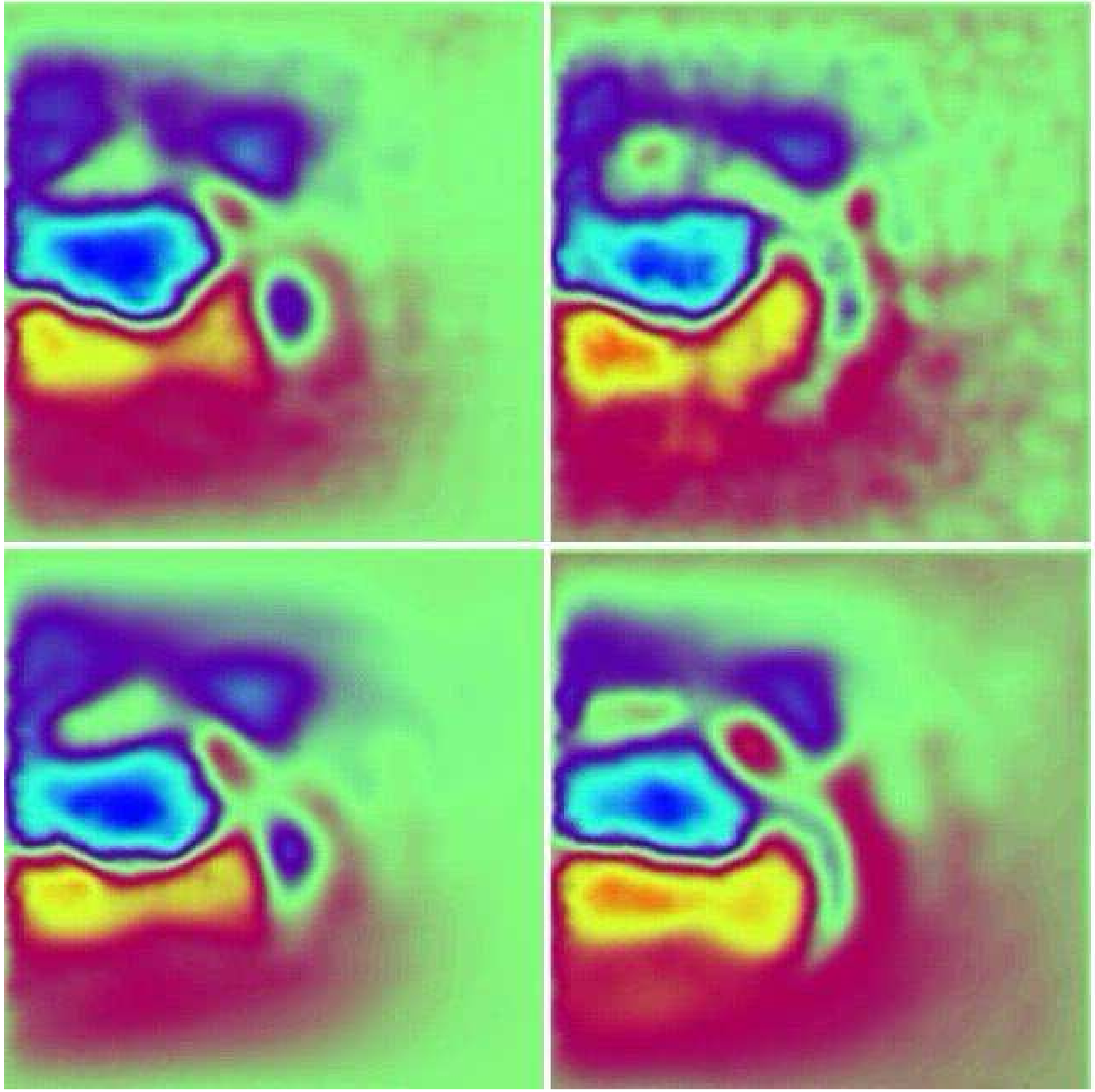
Comparison - noisy obs.



Top : identified initial condition after 5 iterations of BFN and 4D-VAR.

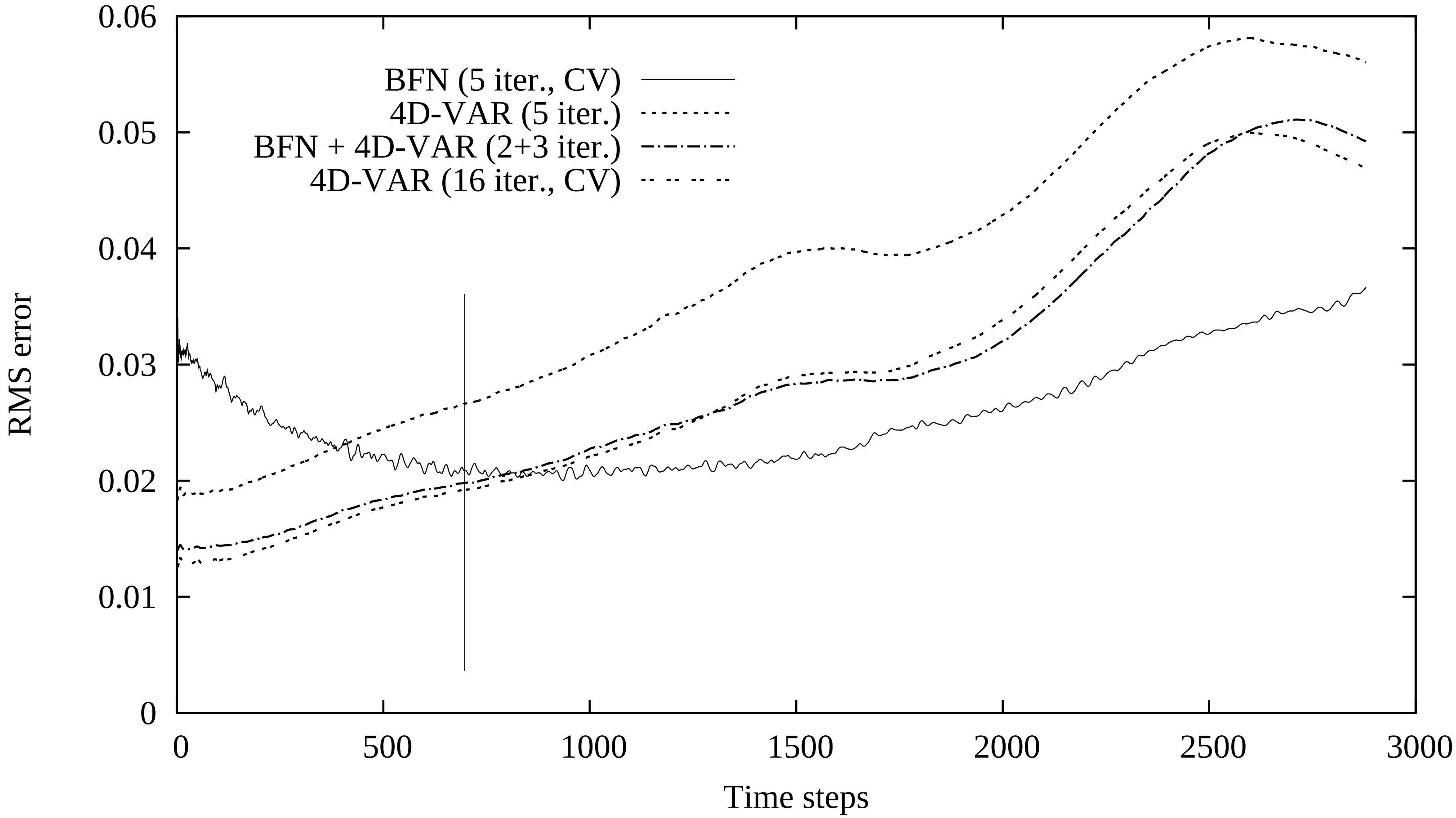
Bottom : true initial condition and background state.

Comparison - noisy obs.



Corresponding states at the end of the forecast period (45 days) : BFN, 4D-VAR, true, background.

BFN-preprocessed 4D-VAR



Relative difference between the true solution and the forecast trajectory corresponding to the BFN, 4D-VAR and BFN-preprocessed 4D-VAR identified initial conditions, after 5 iterations, versus time, for the height variable in the case of noisy observations.

Quasi-geostrophic ocean model

The oceanic model used in this study is based on the quasi-geostrophic approximation obtained by writing the conservation of the potential vorticity. The vertical structure of the ocean is divided into N layers. Each one has a constant density ρ_k with a depth H_k $(k = 1, \ldots, N)$. We get a coupled system of N equations :

$$\frac{D_k(\theta_k(\Psi) + f)}{Dt} + \delta_{k,N}\, C_1 \Delta \Psi_N - C_3 \Delta^3 \Psi_k \;=\; F_k \qquad \text{in } \Omega \times]0, T[, \qquad \forall k = 1, \ldots, N;$$

where :

- $\Omega \subset I\!R^2$ is the oceanic basin, and $[0, T]$ the time interval for the study ;
- Ψ_k is the stream function in the layer k ;
- $C_1 \Delta \Psi_N$ is the dissipation on the bottom of the ocean ;
- $C_3 \Delta^3 \Psi_k$ is the parametrization of internal and subgrid dissipation ;

Example

- $\theta_k(\Psi)$ is the potential vorticity in the layer k, given by :

$$\begin{pmatrix} \theta_1(\Psi) \\ \vdots \\ \theta_N(\Psi) \end{pmatrix} = [\Delta - [W]] \begin{pmatrix} \Psi_1 \\ \vdots \\ \Psi_N \end{pmatrix}$$

  where $[W]$ is a $N \times N$ tridiagonal matrix;

- f is the Coriolis force;
- $\dfrac{D_k .}{Dt}$ is the Lagrangian derivative in layer k, given by :

$$\frac{D_k .}{Dt} = \frac{\partial .}{\partial t} - \frac{\partial \Psi_k}{\partial y}\frac{\partial .}{\partial x} + \frac{\partial \Psi_k}{\partial x}\frac{\partial .}{\partial y} = \frac{\partial .}{\partial t} + J(\Psi_k, .),$$

  where $J(.,.)$ is the Jacobian operator $J(\varphi, \xi) = \dfrac{\partial \varphi}{\partial x}\dfrac{\partial \xi}{\partial y} - \dfrac{\partial \varphi}{\partial y}\dfrac{\partial \xi}{\partial x}$;

- F_k is a forcing term. In this model only the tension due to the wind, denoted τ, is taken into account : $F_1 = \mathrm{Rot}\tau$ and $F_k = 0, \ \forall k \geq 2$.

Example

We consider altimetric measurement of the surface of the ocean given by satellite observations (Topex-Poseidon, Jason). The observed data is the change in the surface of the ocean. According to the quasi geostrophic approximation it is proportional to the stream function in the surface layer :

$$h^{obs} = \frac{f_0}{g} \Psi_1^{obs}$$

Therefore we will assimilate surface data in order to retrieve the fluid circulation especially in the deep ocean layers.

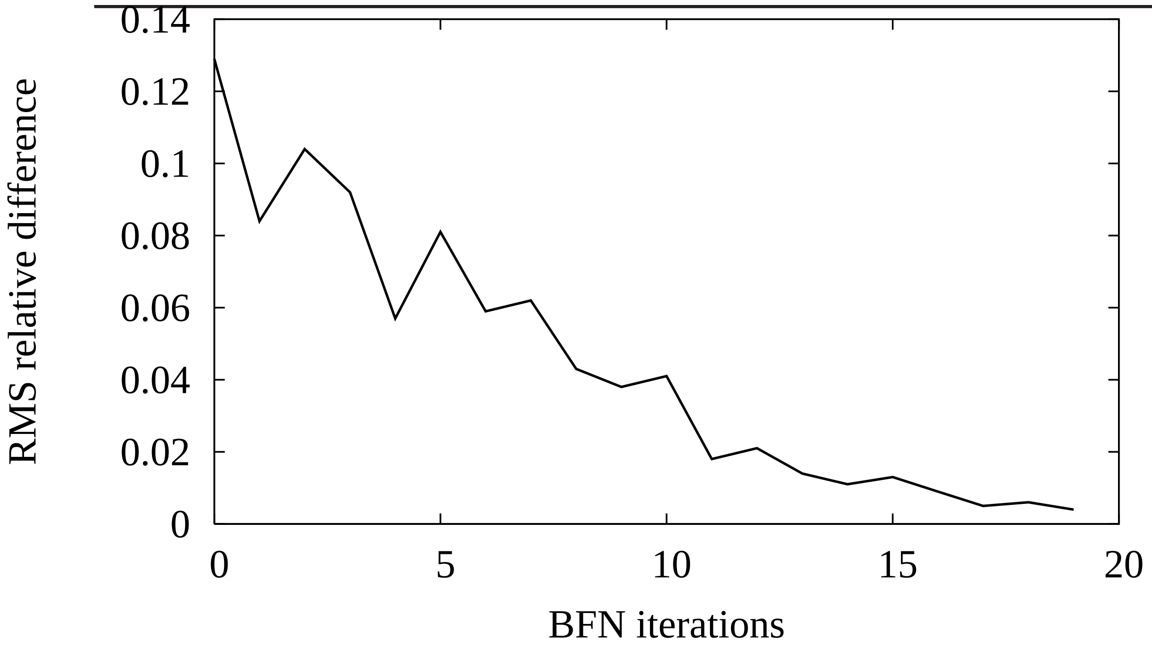
The control vector is the initial state on the N layers :

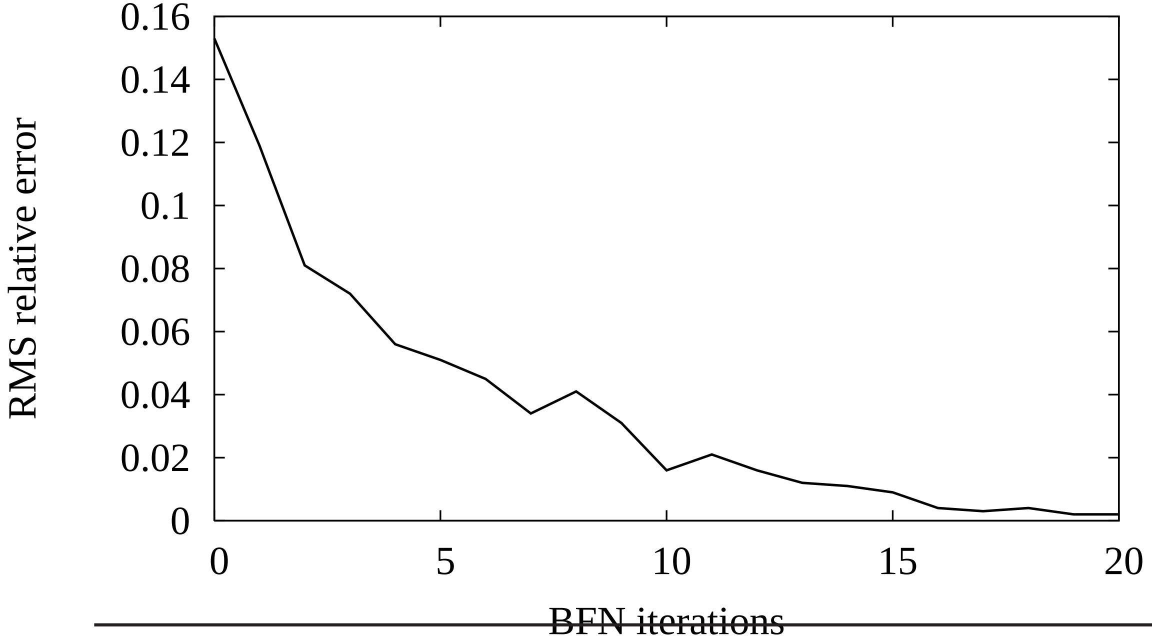
$$u = \Big(\Psi_k(t = 0) \Big)_{k=1,\dots,N} \in \mathcal{U}_{\text{ad}}$$

The state vector is

$$\Big(\Psi_k(t) \Big)_{k=1,\dots,N}$$

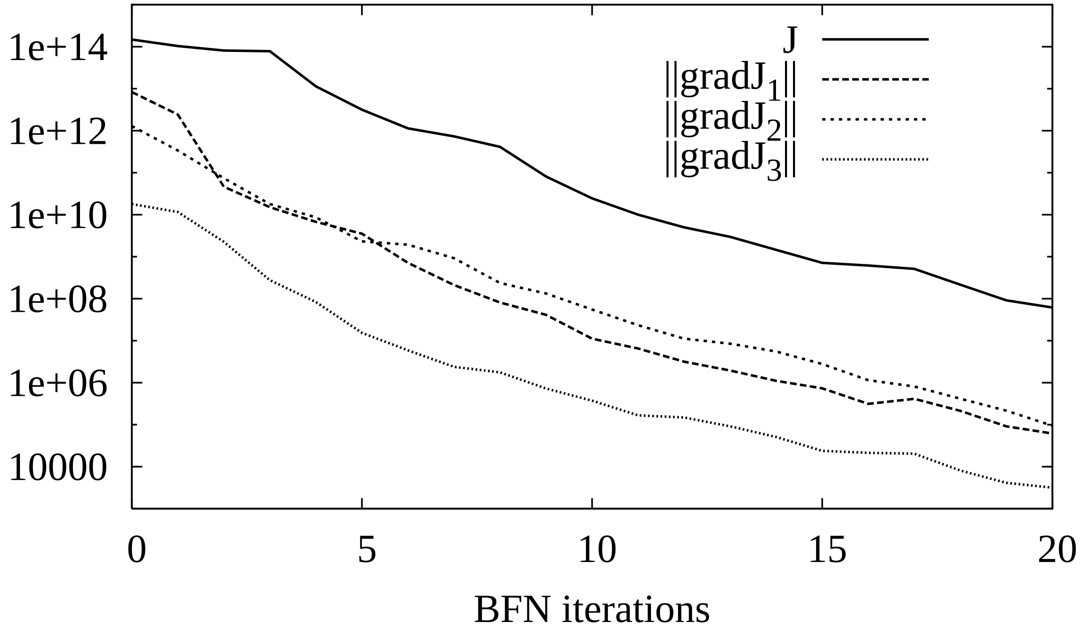
QG - BFN convergence

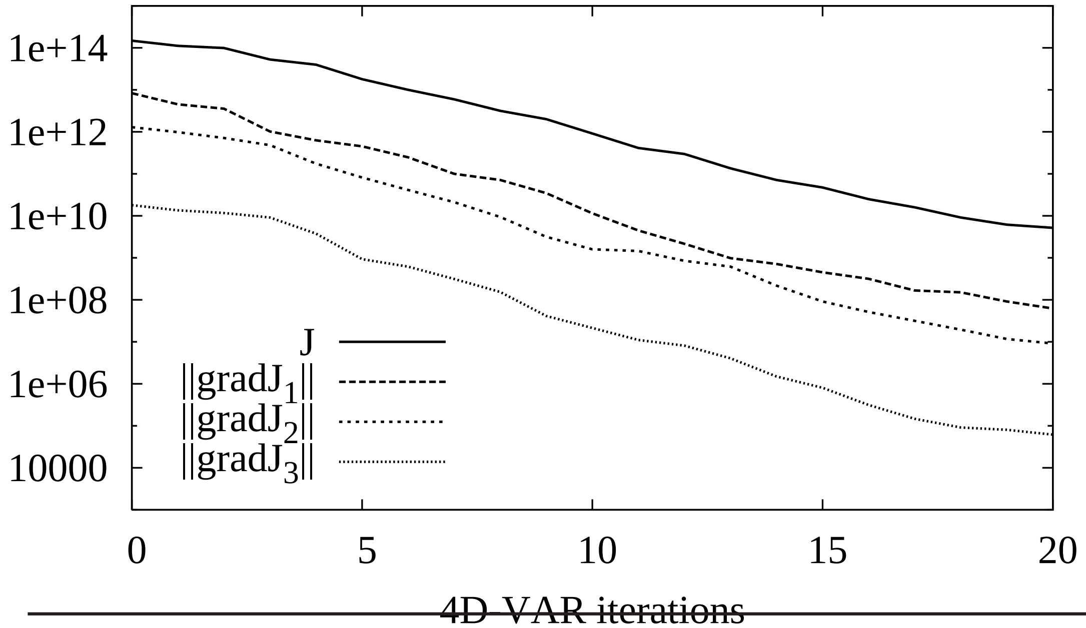




RMS relative difference between two consecutive BFN iterates (a) and between the BFN iterates and the exact solution (b) versus the number of BFN iterations.

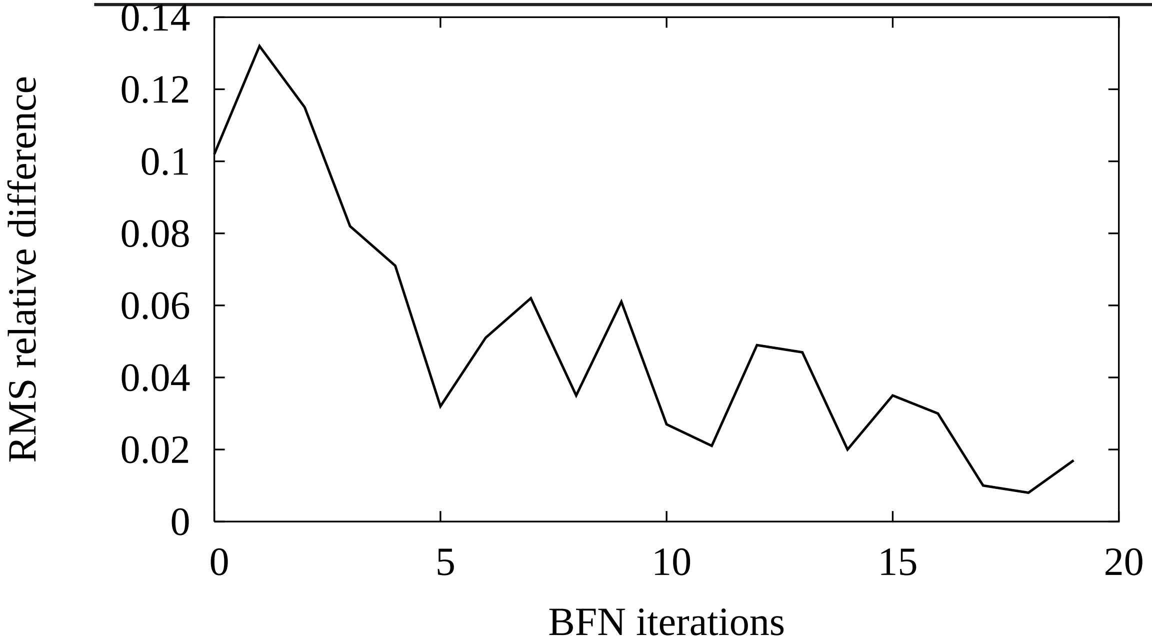
QG - BFN vs 4D-VAR

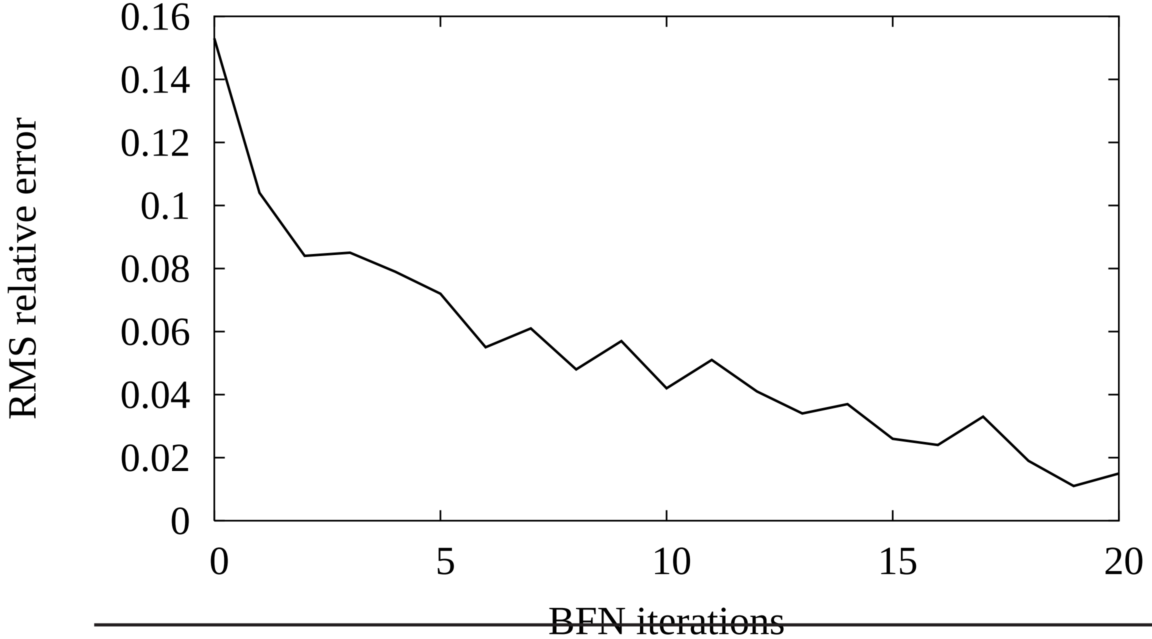




Evolution of the 4D-VAR cost function and of the 3 gradients of the cost function in the 3 ocean layers for the BFN iterates versus the number of BFN iterations (top) and for the 4D-VAR iterates versus the number of 4D-VAR iterations (b).

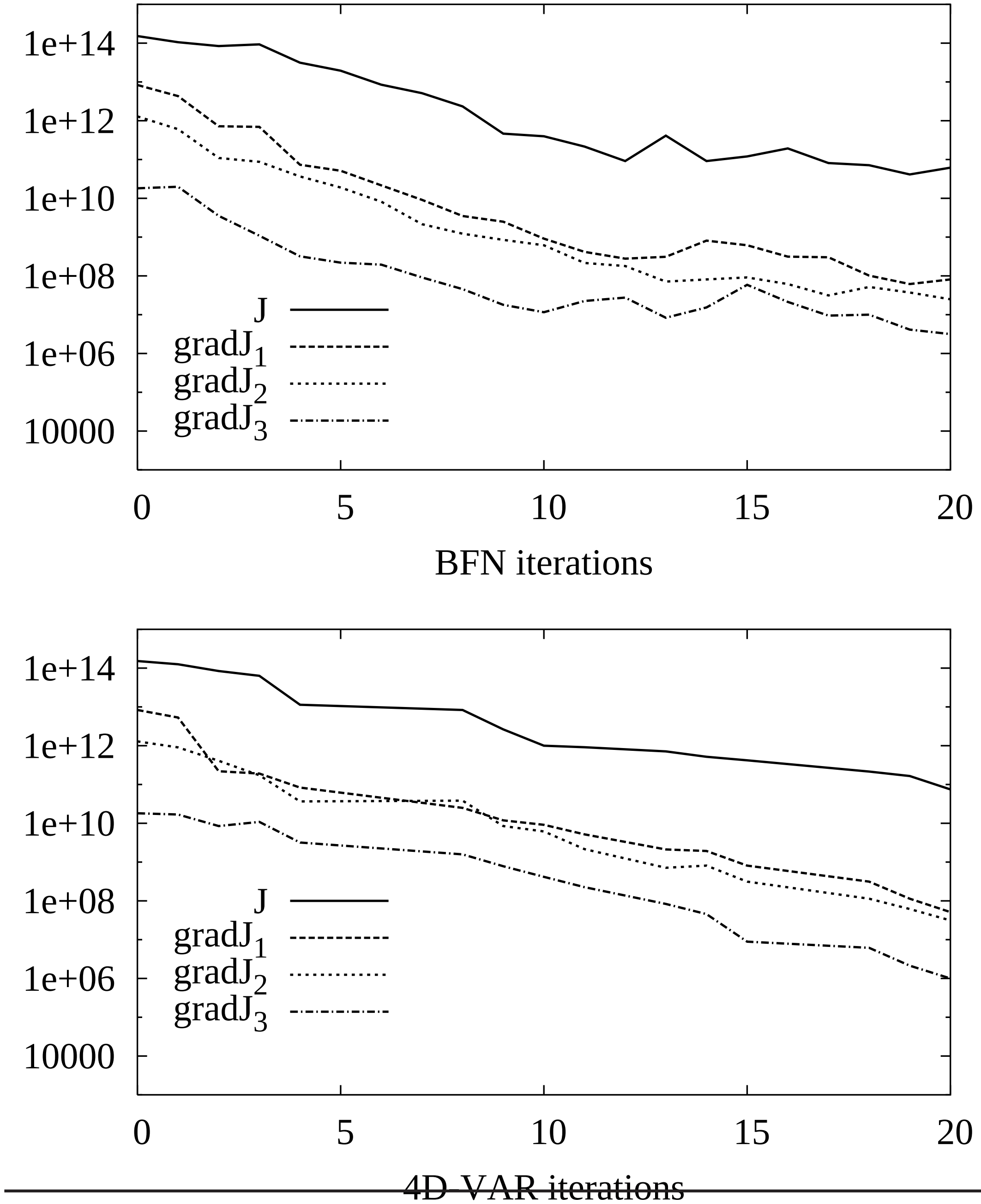
QG - BFN convergence (noisy obs.)





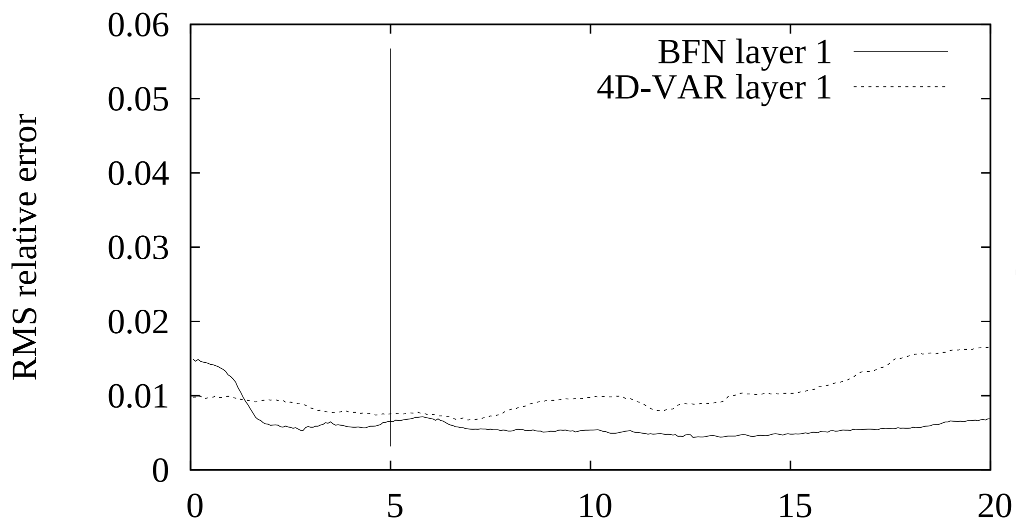
RMS relative difference between two consecutive BFN iterates (a) and between the BFN iterates and the exact solution (b) versus the number of BFN iterations in the case of noised observations.

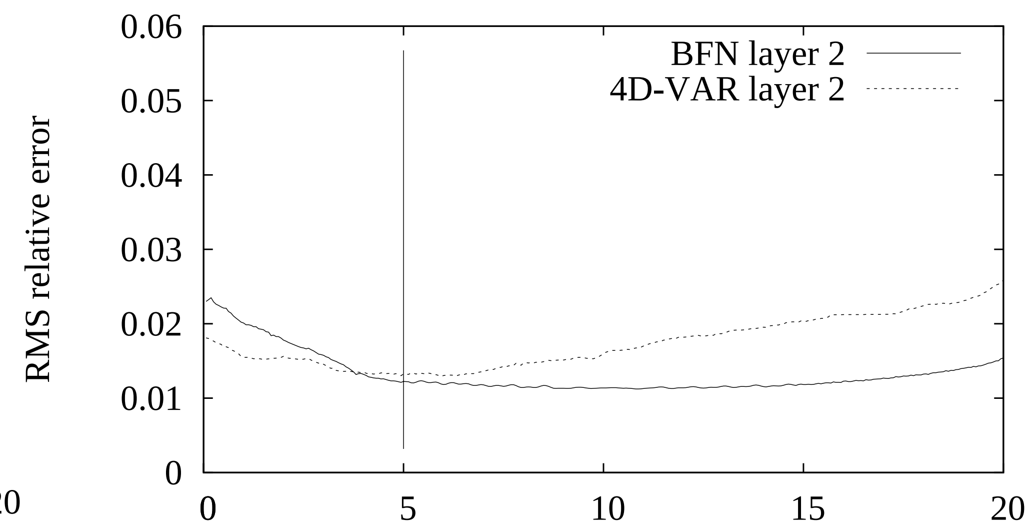
QG - BFN vs 4D-VAR (noisy obs.)

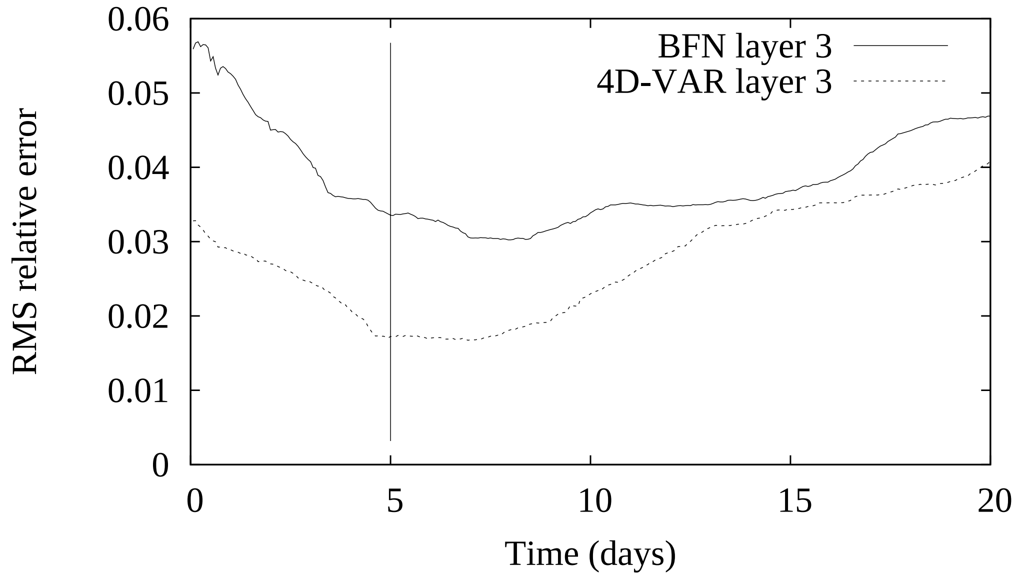


Evolution of the 4D-VAR cost function and of the 3 gradients of the cost function in the 3 ocean layers for the BFN iterates versus the number of BFN iterations (top) and for the 4D-VAR iterates versus the number of 4D-VAR iterations (b), in the case of noisy observations.

QG - BFN vs 4D-VAR (noisy obs.)

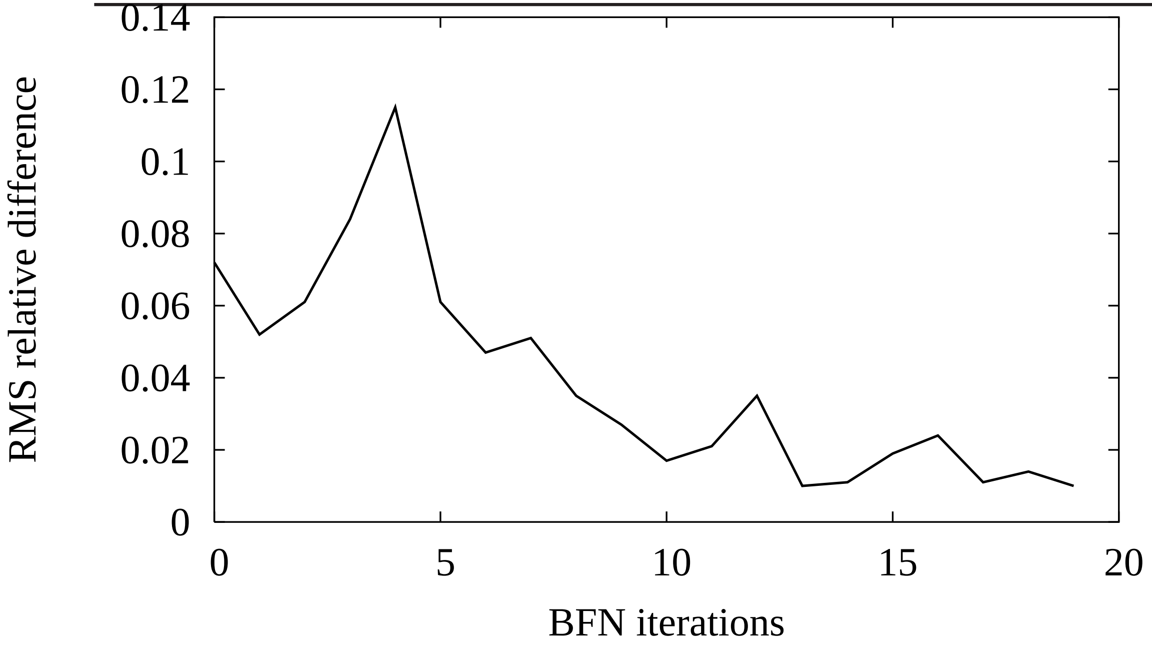


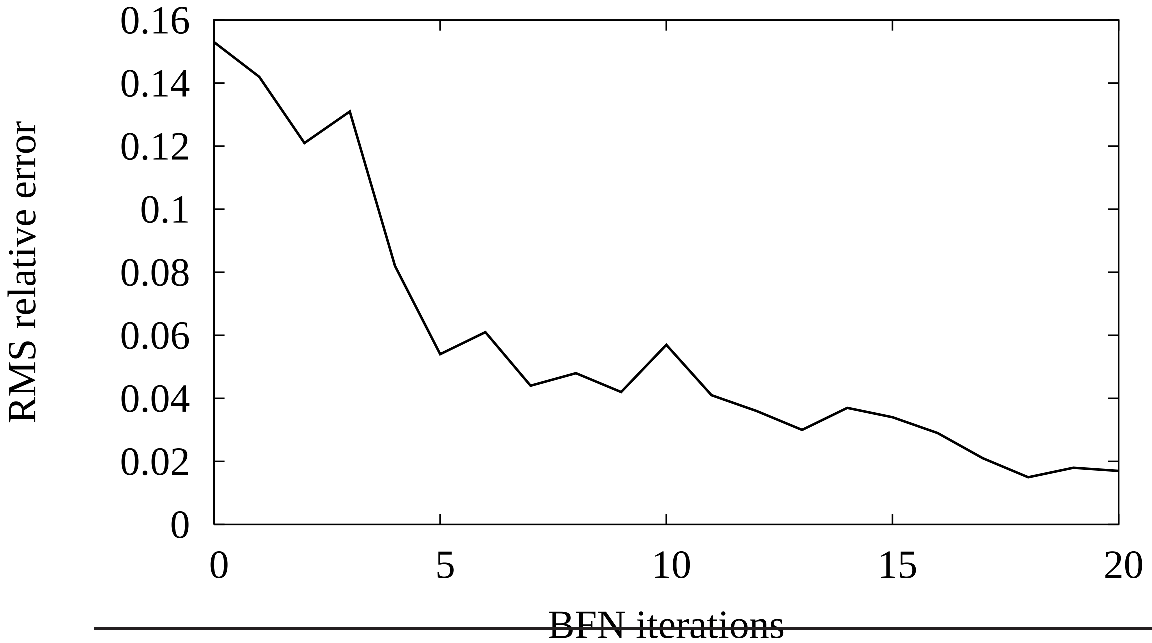




Evolution in time of the RMS relative difference between the reference trajectory and the identified trajectories for the BFN (solid line) and 4D-VAR (dotted line) algorithms, versus time, for each layer : from surface to bottom.

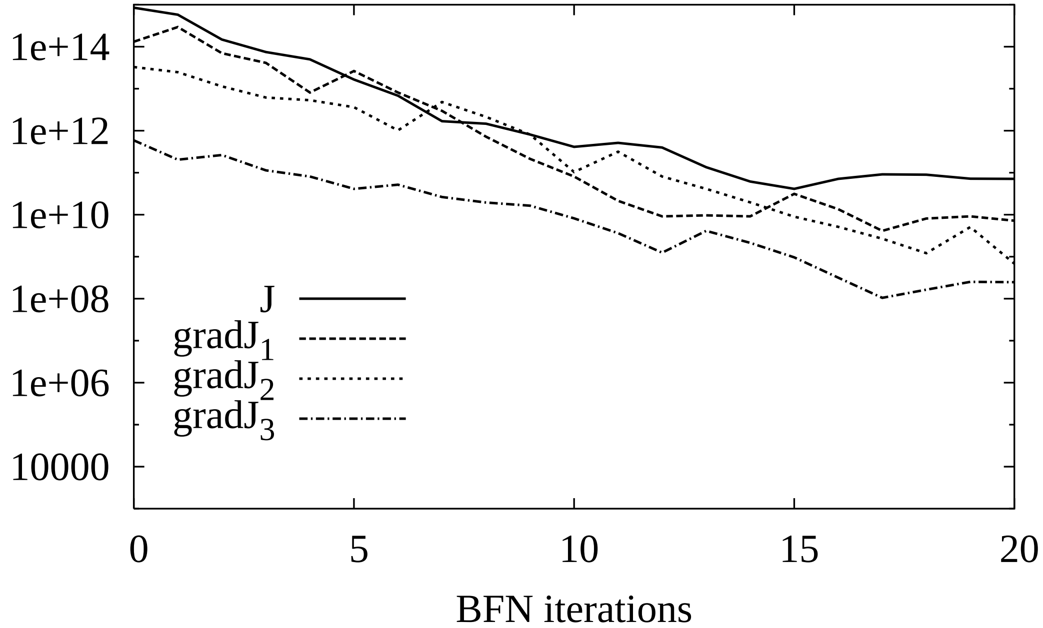
QG - BFN convergence (model error)

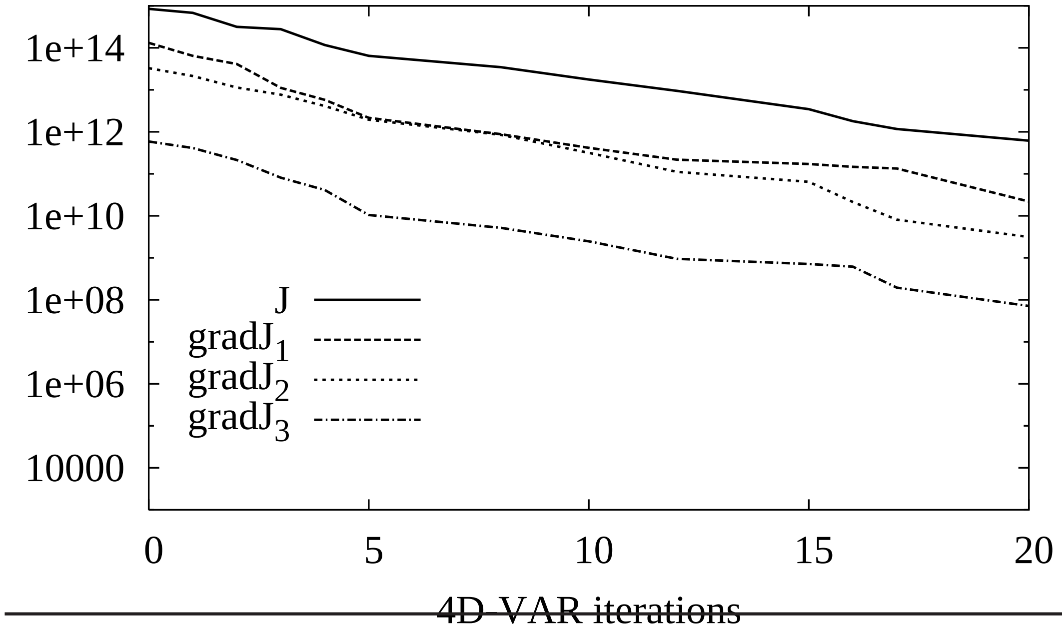




RMS relative difference between two consecutive BFN iterates (a) and between the BFN iterates and the exact solution (b) versus the number of BFN iterations in the case of imperfect model.

QG - BFN vs 4D-VAR (model error)





Evolution of the 4D-VAR cost function and of the 3 gradients of the cost function in the 3 ocean layers for the BFN iterates versus the number of BFN iterations (a) and for the 4D-VAR iterates versus the number of 4D-VAR iterations (b) in the case of imperfect model.

Full primitive ocean model

Primitive equations : Navier-Stokes equations (velocity-pressure), coupled with two active tracers (temperature and salinity).

Momentum balance :

$$\frac{\partial U_h}{\partial t} = -\left[(\nabla \wedge U) \wedge U + \frac{1}{2}\nabla(|U|^2) \right]_h - f.z \wedge U_h - \frac{1}{\rho_0}\nabla_h p + D^U + F^U$$

Incompressibility equation : $\nabla.U = 0$

Hydrostatic equilibrium : $\frac{\partial p}{\partial z} = -\rho g$

Heat and salt conservation equations :

$$\frac{\partial T}{\partial t} = -\nabla.(TU) + D^T + F^T \quad \text{(+ same for S)}$$

Equation of state : $\rho = \rho(T, S, p)$

Full primitive ocean model

Free surface formulation : the height of the sea surface η is given by

$$\frac{\partial \eta}{\partial t} = -div_h((H+\eta)\bar{U_h}) + [P-E]$$

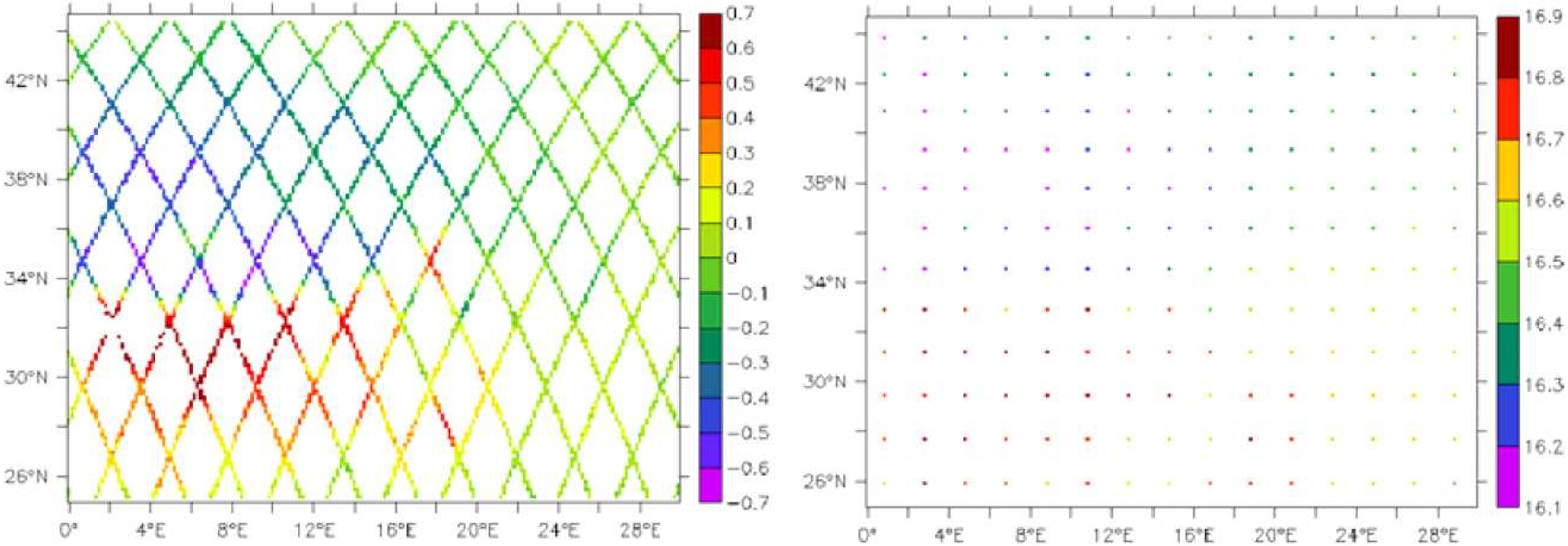
The surface pressure is given by : $p_s = \rho g \eta.$

This boundary condition is then used for integrating the hydrostatic equilibrium and calculating the pressure.

Numerical experiments : double gyre circulation confined between closed boundaries (similar to the shallow water model). The circulation is forced by a sinusoidal (with latitude) zonal wind.

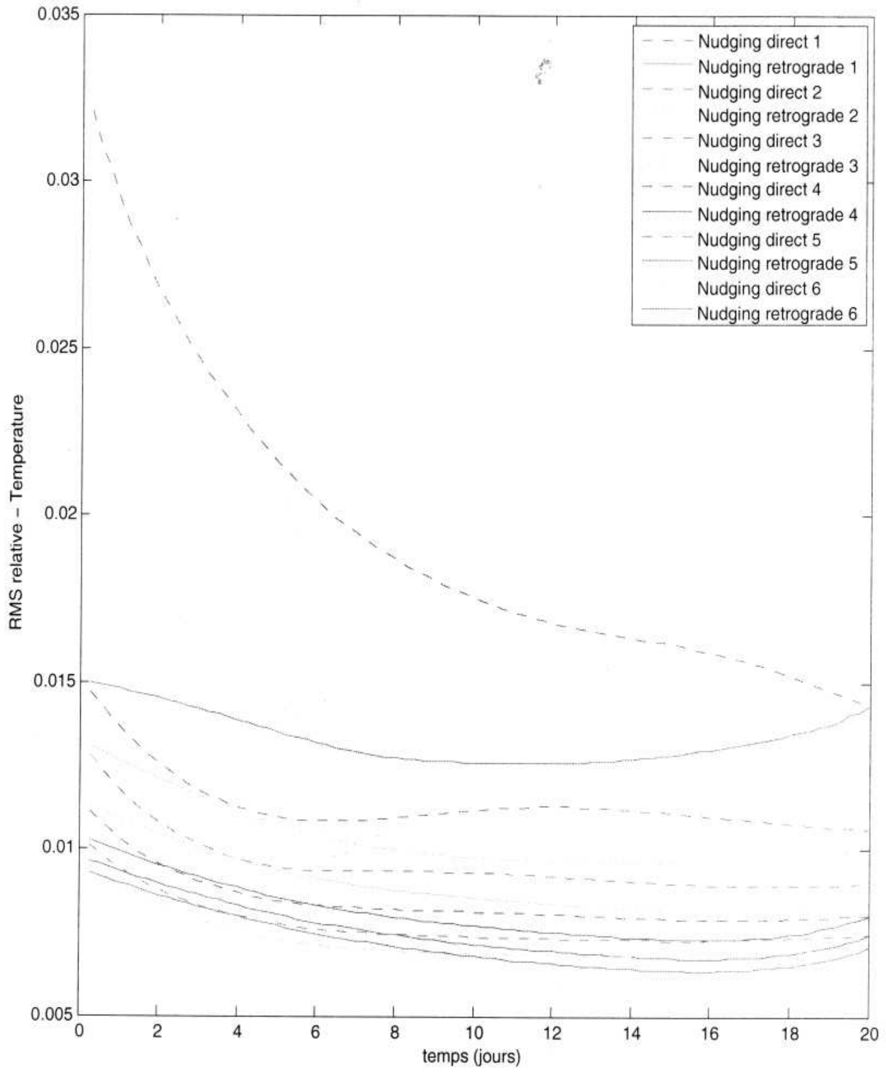
Twin experiments : observations are extracted from a reference run, according to networks of realistic density : SSH is observed similarly to TOPEX/POSEIDON, and temperature is observed on a regular grid that mimics the ARGO network density. [P. Bansart - PhD Thesis 2011]

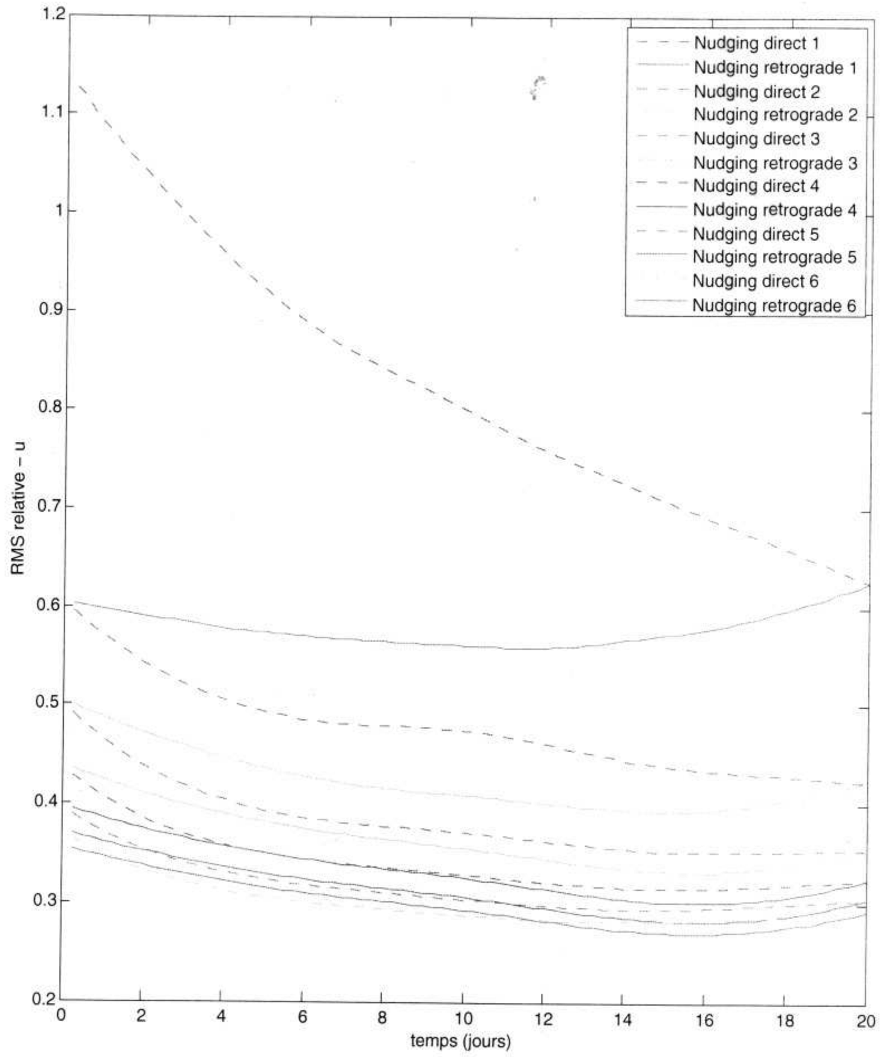
Full primitive ocean model



Example of observation network used in the assimilation : along-track altimetric observations (Topex-Poseidon) of the SSH every 10 days ; vertical profiles of temperature (ARGO float network) every 18 days.

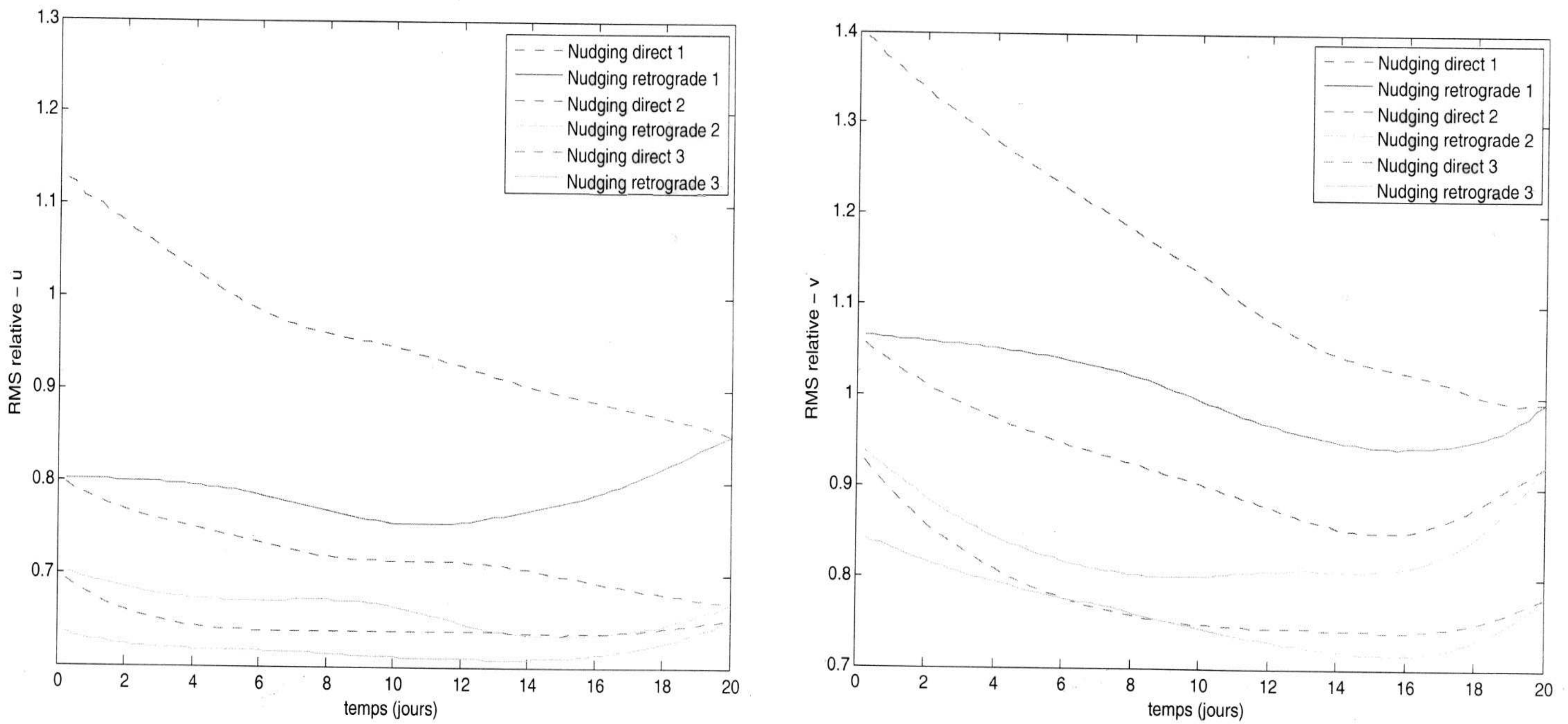
Numerical results





Relative RMS error of the temperature (left) and longitudinal velocity (right), 6 iterations of BFN (nudging terms in the temperature and SSH equations only), with full SSH observations every day.

Numerical results



Relative RMS error of the longitudinal and transversal velocities, 3 iterations of BFN (nudging terms in the temperature and SSH equations only), with "realistic" SSH observations.

Talk overview

1. Back and forth nudging algorithm
2. Numerical experiments : shallow water, QG model, NEMO model

⇒ 3. Extension to back and forth Kalman filters

4. Diffusive back and forth nudging

SEEK filter

Singular Evolutive Extended Kalman filter :

Consider low rank approximations of the covariance matrices P_n^a, while preserving the stability of the filter : $|Sp((I - K_{n+1}H_{n+1})M_n)| < 1$.

Initialization of the filter : any covariance matrix P is real and symmetric $\rightsquigarrow$ diagonalization : $P = LDL^T$, D being diagonal positive.

Initialization :

$$P_0^f = S_0^f D_0 S_0^{fT} = [\sqrt{\lambda_1}L_1, \dots, \sqrt{\lambda_r}L_r]\,[\sqrt{\lambda_1}L_1, \dots, \sqrt{\lambda_r}L_r]^T$$

where D_0 is the truncated diagonal matrix with the few first eigenvalues.

At any time, the covariance matrices of forecast error can also be decomposed in a similar way, because their rank is r : $\quad P_n^f = S_n^f S_n^{fT}$
where S_n^f and S_n^a are $dim(x_0) \times r$ matrices.

Error covariances matrices

Propagation of errors and gain matrices similar to Kalman filter :

$$P_n^a = S_n^f[I + (H_n S_n^f)^T R_n^{-1} (H_n S_n^f)]^{-1} S_n^{fT} = S_n^a S_n^{aT}$$

$(rank(P_n^a) = r)$

As one usually considers 10 to 30 different model modes (number of vectors of large variability, corresponding to the columns of S_n), it is possible to consider their time evolution (which was difficult in the full filter). For $1 \leq j \leq r$:

$$[S_{n+1}^f]_j = M(x_n^a + [S_n^a]_j) - M(x_n^a) \simeq M'[S_n^a]_j$$

The last approximation is equivalent to neglect the nonlinear propagation of errors.

[Pham-Verron-Roubaud (98)]

SEEK Filter algorithm

- Initialization :

$$x_0^f \text{ and } P_0^f \text{ given (rank = r, or if not, it will be approximated)}$$

- Analysis :

$$\begin{aligned}
P_n^f &= S_n^f S_n^{fT} \\
K_n &= S_n^f [I_r + (H_n S_n^f)^T R_n^{-1} (H_n S_n^f)]^{-1} (H_n S_n^f)^T R_n^{-1} \\
x_n^a &= x_n^f + K_n [x_{obs\,n} - H_n(x_n^f)] \\
P_n^a &= S_n^f [I_r + (H_n S_n^f)^T R_n^{-1} (H_n S_n^f)]^{-1} S_n^{fT}
\end{aligned}$$

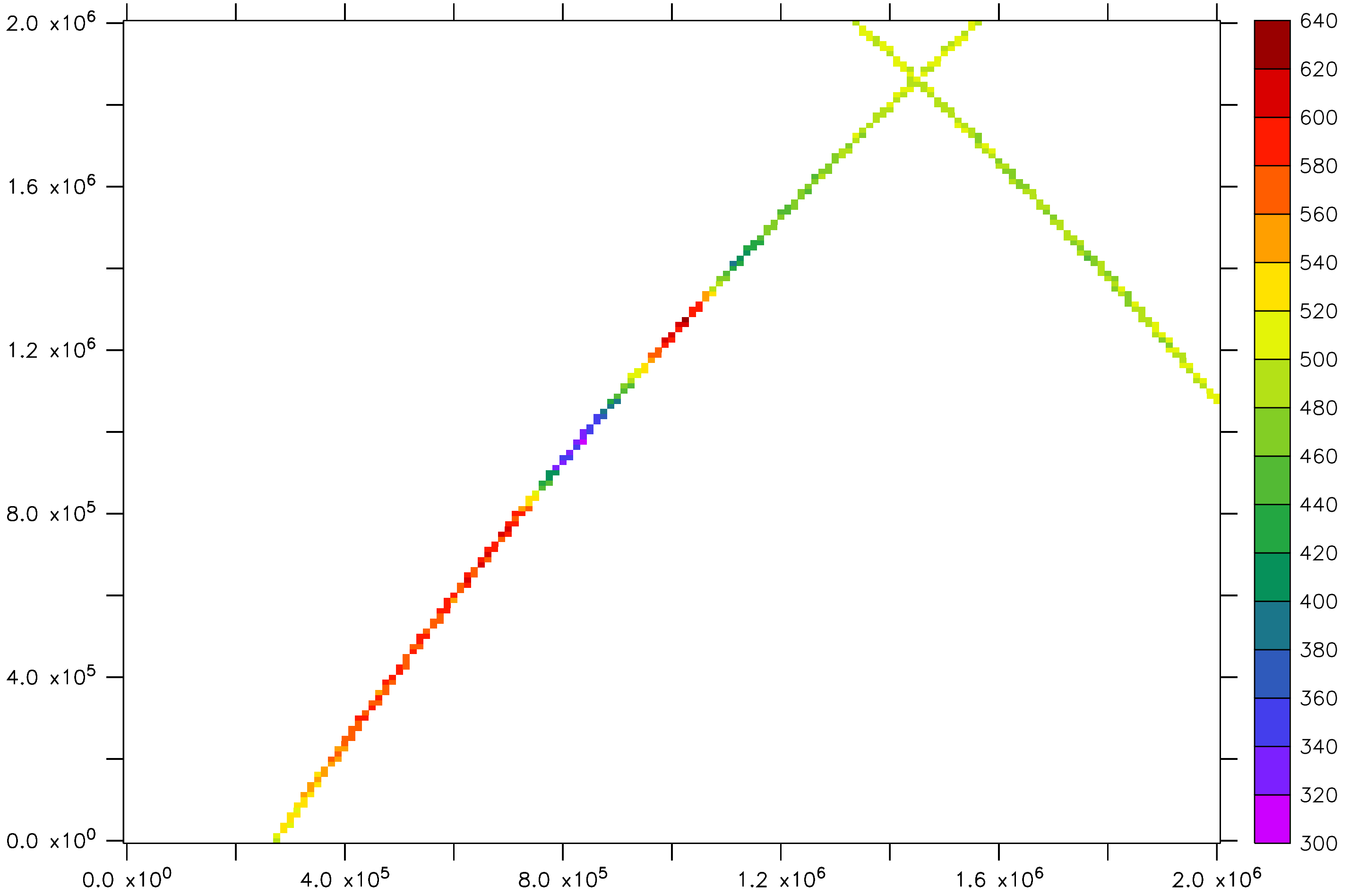
- Forecast :

$$\begin{aligned}
x_{n+1}^f &= M_{n;n+1}(x_n^a) \\
[S_{n+1}^f]_j &= M(x_n^a + [S_n^a]_j) - M(x_n^a), \quad 1 \leq j \leq r \\
P_{n+1}^f &= S_{n+1}^f (S_{n+1}^f)^T
\end{aligned}$$

BF-SEEK

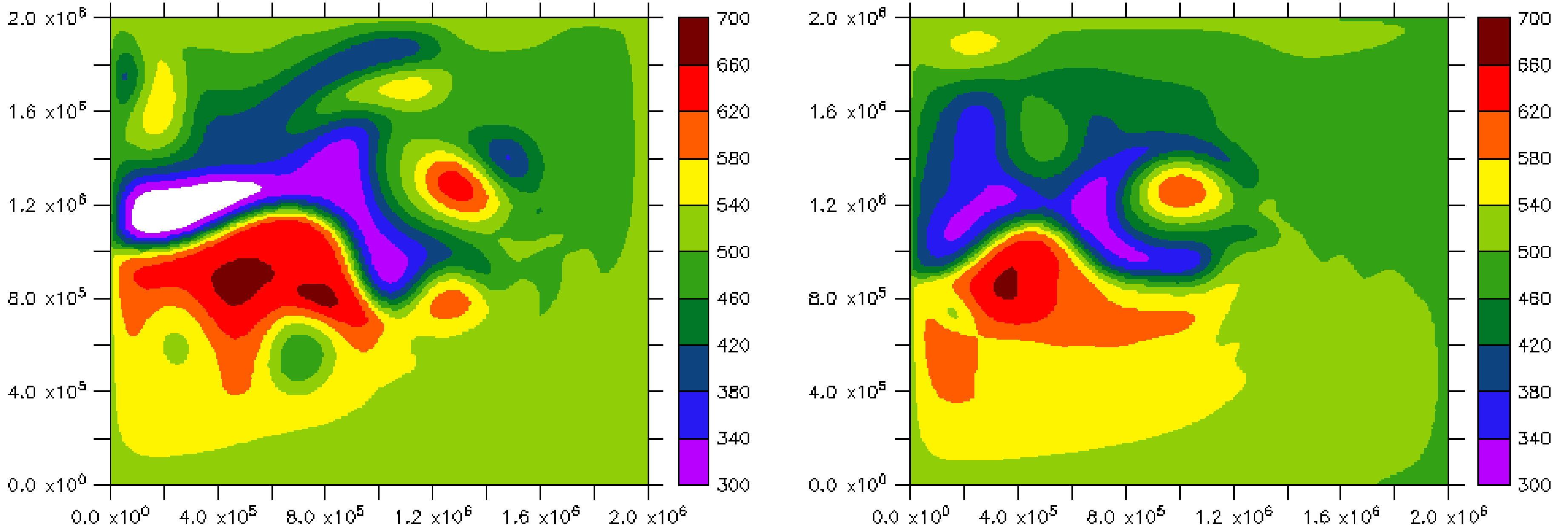
Numerical configuration :

- Shallow water model (2D), domain : 2000×2000 km
- Assimilation window : 30 days, daily assimilation
- SSH observation lines along Topex/Poseidon ground tracks
- 4 iterations of BF-SEEK
- But no dynamical propagation of the errors (still in progress) ($\rightsquigarrow$ SEEK $\simeq$ optimal interpolation)



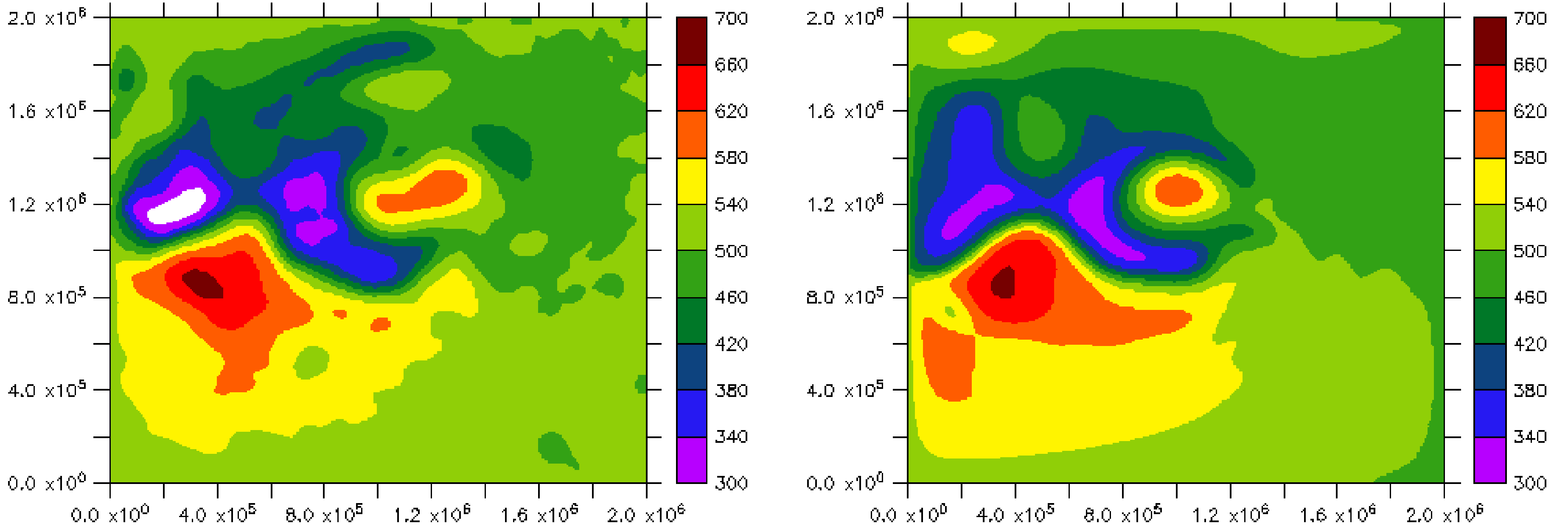
Example of assimilated data every day

BF-SEEK



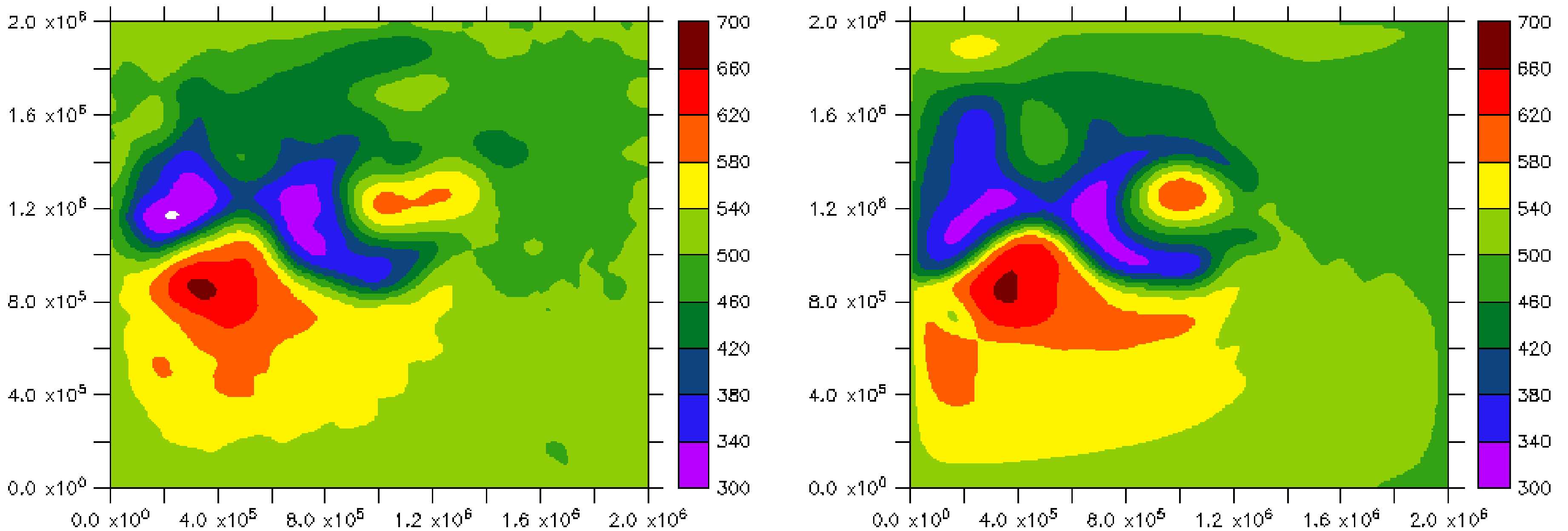
Identified SSH (left) and "true" SSH (right) after : 0 iteration of BF-SEEK

BF-SEEK



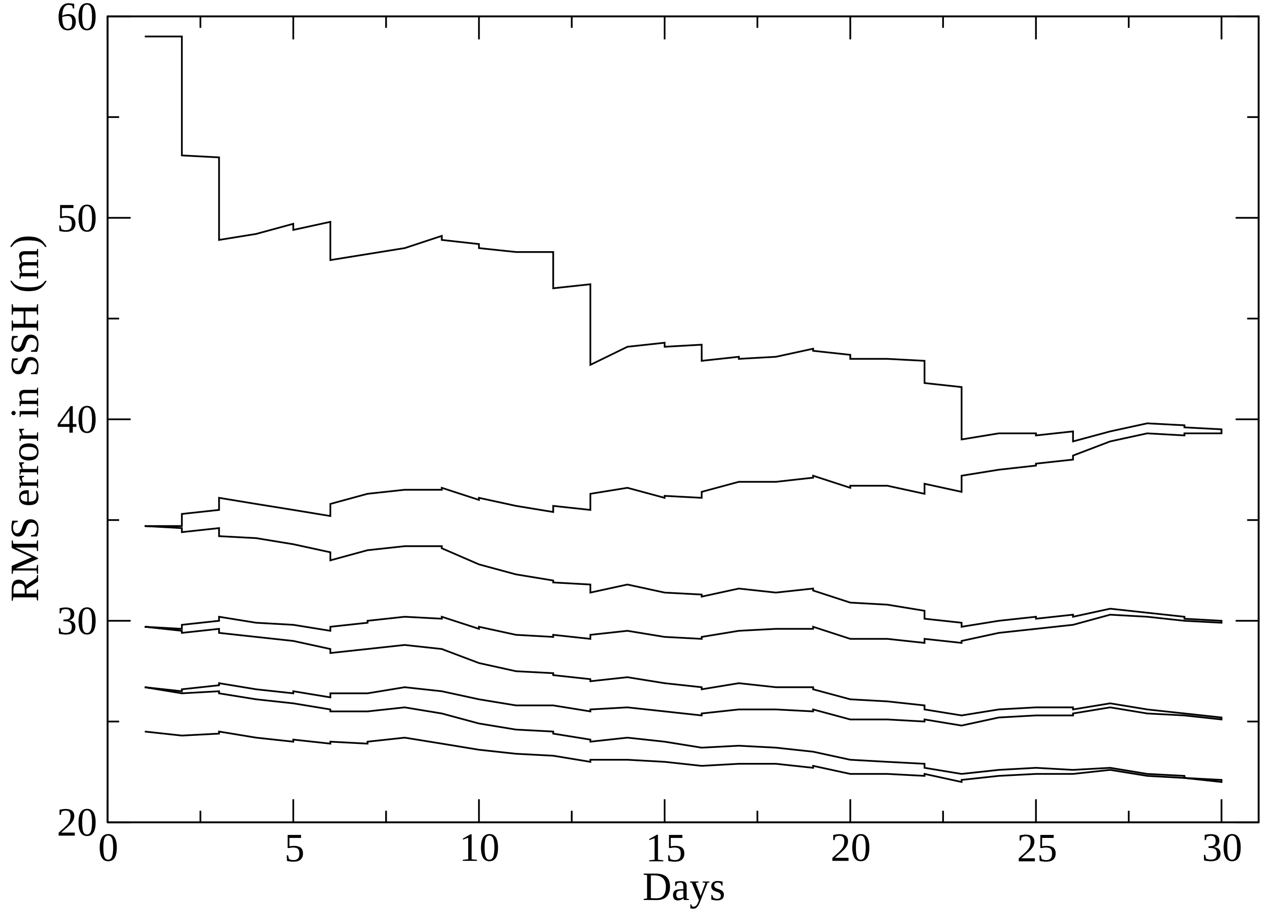
Identified SSH (left) and “true” SSH (right) after : 2 iterations of BF-SEEK

BF-SEEK



Identified SSH (left) and “true” SSH (right) after : 4 iterations of BF-SEEK

BF-SEEK



Evolution of the RMS error on the SSH during the BF-SEEK iterations.

Talk overview

1. Back and forth nudging algorithm

2. Numerical experiments : shallow water, QG model, NEMO model

3. Extension to back and forth Kalman filters

⇒ 4. Diffusive back and forth nudging

Diffusive BFN

Diffusive free equations in the geophysical context :

In meteorology or oceanography, theoretical equations are usually diffusive free (e.g. Euler's equation for meteorological processes).

In a numerical framework, a diffusive term is added to the equations (or a diffusive scheme is used), in order to both stabilize the numerical integration of the equations, and take into consideration some subscale phenomena.

Original model and corresponding BFN algorithm :

$$\partial_t X = F(X), \quad 0 < t < T.$$

$$\begin{cases} \partial_t X_k = F(X_k) + K(Y_{obs} - H(X_k)), \\ X_k(0) = \tilde{X}_{k-1}(0), \quad 0 < t < T, \end{cases} \begin{cases} \partial_t \tilde{X}_k = F(\tilde{X}_k) - K'(Y_{obs} - H(\tilde{X}_k)), \\ \tilde{X}_k(T) = X_k(T), \quad T > t > 0, \end{cases}$$

with the notation $\tilde{X}_0(0) = x_0$.

Diffusive BFN

Addition of a diffusion term :

$$\partial_t X = F(X) + \nu\Delta X, \quad 0 < t < T,$$

where F has no diffusive terms, ν is the diffusion coefficient, and we assume that the diffusion is a standard second-order Laplacian.

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

$$\begin{cases} \partial_t X_k = F(X_k) + \nu\Delta X_k + K(Y_{obs} - H(X_k)), \\ X_k(0) = \tilde{X}_{k-1}(0), \quad 0 < t < T, \end{cases}$$

$$\begin{cases} \partial_t \tilde{X}_k = F(\tilde{X}_k) - \nu\Delta \tilde{X}_k - K'(Y_{obs} - H(\tilde{X}_k)), \\ \tilde{X}_k(T) = X_k(T), \quad T > t > 0. \end{cases}$$

Diffusive BFN

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

$$\partial_{t'} \tilde{X}_k = -F(\tilde{X}_k) + \nu \Delta \tilde{X}_k + K'(Y_{obs} - H(\tilde{X}_k)), \quad \tilde{X}_k(t' = 0) = X_k(T),$$

where $\tilde{X}$ is evaluated at time t'.

Then the backward equation can easily be solved, 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.

Convergence of D-BFN

We assume here that the model F and the observation operator H are linear. Let us define the following operator that corresponds to one forward and one backward integrations :

$$\psi \quad : \quad (X_1(0), Y_{obs}(0)) \mapsto \tilde{X}_1(0).$$

This operator is linear in the initial conditions, so that there exist C and D linear operators such that

$$X_2(0) = \psi(X_1(0), Y_{obs}(0)) = \psi(X_1(0), 0) + \psi(0, Y_{obs}(0)) = CX_1(0) + DY_{obs}(0).$$

So that the initial state $X_{k+1}(0)$ of the $(k+1)^{\text{th}}$ D-BFN iteration satisfies :

$$X_{k+1}(0) = C^k x_0 + \left(\sum_{m=0}^{k-1} C^m \right) DY_{obs}(0)$$

Convergence of D-BFN

If the spectrum of C is included in $[-\rho;\rho]$, with $\rho < 1$, then $C^k \to 0$ and $\sum_{m=0}^{k} C^m \to (I-C)^{-1}$ when $k \to \infty$. Therefore, in that case, $X_k(0)$ converges as k goes to infinity to X_∞ solution of

$$X_\infty = (I-C)^{-1} D Y_{obs}(0)$$

Linear transport equation

$$\partial_t u + a(x)\,\partial_x u = 0, \quad t \in [0,T],\ x \in \Omega, \quad u(t=0) = u_0 \in L^2(\Omega)$$

with periodic boundary conditions, and we assume that $a \in W^{1,\infty}(\Omega)$.

Numerically, for both stability and subscale modelling, the following equation would be solved :

$$\partial_t u + a(x)\,\partial_x u = \nu \partial_{xx} u, \quad t \in [0,T], x \in \Omega, \quad u(t=0) = u_0 \in L^2(\Omega),$$

where $\nu \geq 0$ is assumed to be constant and rather small.

Linear transport equation

Let us assume that the observations satisfy the physical model (without diffusion) :

$$\partial_t u_{obs} + a(x)\,\partial_x u_{obs} = 0, \quad t \in [0,T], x \in \Omega, \quad u_{obs}(t=0) = u^0_{obs} \in L^2(\Omega).$$

We assume in this idealized situation that the system is fully observed (and H is then the identity operator).

Then the D-BFN algorithm applied to this problem gives, for $k \geq 1$:

$$\begin{cases} \partial_t u_k + a(x)\,\partial_x u_k = \nu \partial_{xx} u_k + K(u_{obs,k} - u_k), \\ t \in [2(k-1)T, 2(k-1)T + T], x \in \Omega \\ u_k(2(k-1)T, x) = \tilde{u}_{k-1}(2(k-1)T, x) \end{cases}$$

$$\begin{cases} \partial_t \tilde{u}_k - a(x)\,\partial_x \tilde{u}_k = \nu \partial_{xx} \tilde{u}_k + K(\tilde{u}_{obs,k} - \tilde{u}_k), \\ t \in [2kT - T, 2kT], x \in \Omega \\ \tilde{u}_k(2kT - T, x) = u_k(2kT - T, x). \end{cases}$$

Energy estimate

Energy estimate :

$$d_t\|u\|^2 \leq -2\nu\|\partial_x u\|^2 - (2K - \|\partial_x a\|_\infty)\|u\|^2 \leq -\delta\|u\|^2,$$

where $\delta = 2K - \|\partial_x a\|_\infty$ is non negative for K large enough. Therefore $\|Cu_0\|^2 \leq e^{-2\delta T}\|u_0\|^2$, so that $\|C\| < 1$ and convergence is assured.

In the special case where $a(x) = a \in \mathbb{R}$, we can change variables to straighten characteristics as follows. Setting $v_k(t,y) = u_k(t, y + a(t - 2(k-1)T))$ and $\tilde{v}_k(t,z) = \tilde{u}_k(t, z - a(t - 2kT))$ leads to

$$\partial_t v_k = \nu\partial_{yy} v_k + K(u^0_{obs}(y) - v_k), \qquad \partial_t \tilde{v}_k = \nu\partial_{zz}\tilde{v}_k + K(u^0_{obs}(z) - \tilde{v}_k).$$

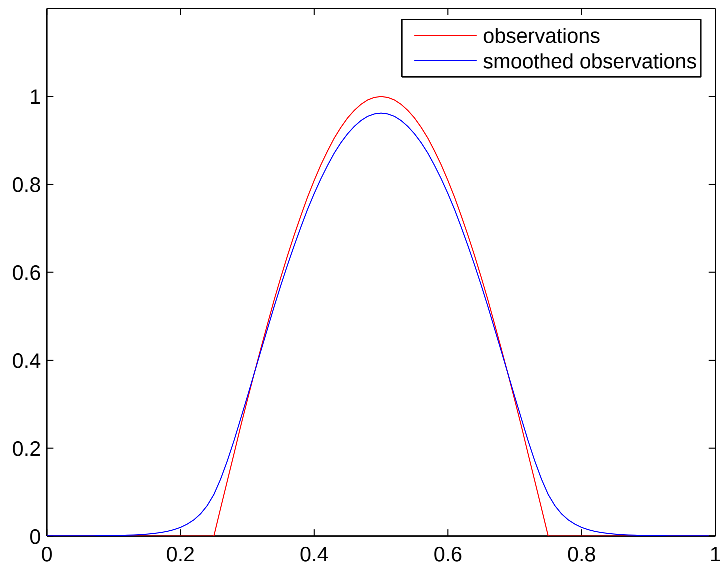
Smoothing equation

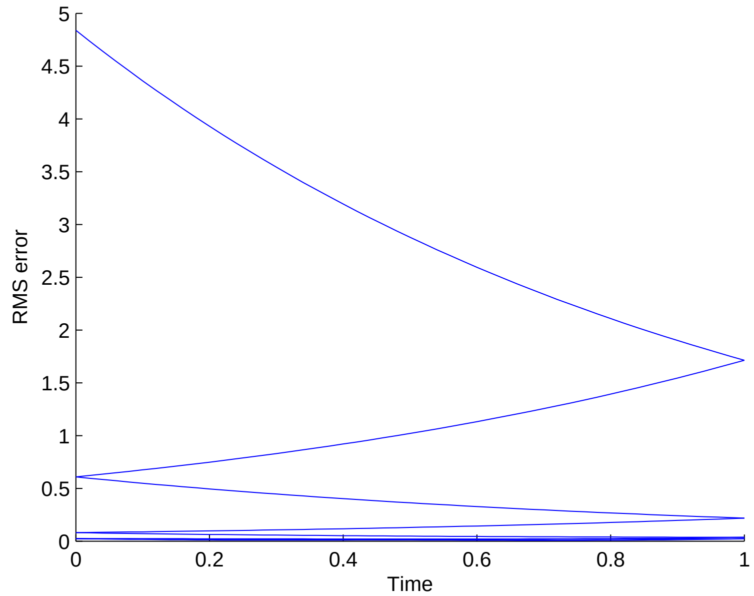
At the limit $k \to \infty$, v_k and $\tilde{v}_k$ tend to $v_\infty(x)$ solution of

$$\nu \partial_{xx} v_\infty + K(u_{obs}^0(x) - v_\infty) = 0, \quad x \in \mathbb{R}.$$

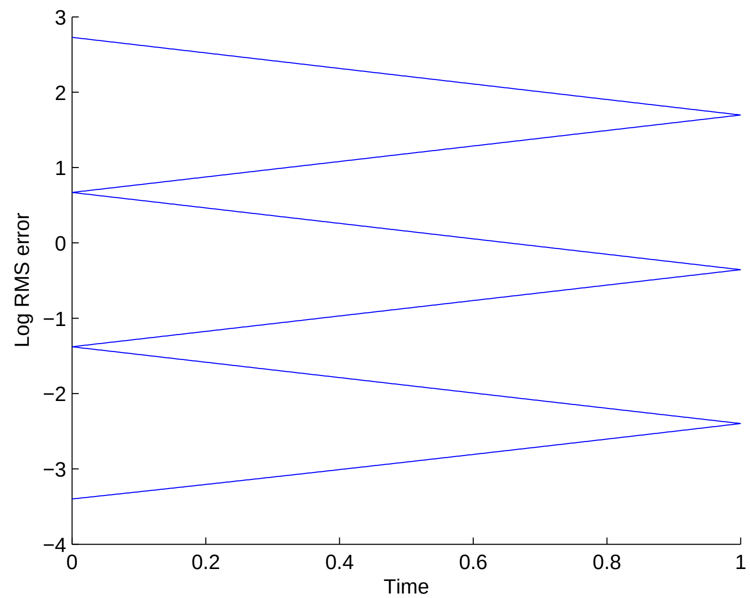
This equations is well known in signal or image processing, as being the standard linear diffusion restoration equation. In some sense, v_∞ is the result of a smoothing process on the observations u_{obs}, where the degree of smoothness is given by the ratio $\frac{\nu}{K}$.

Numerical experiments





Initial condition of the observation and corresponding smoothed solution ; RMS difference between the BFN iterates and the smoothed observations ; same in semi-log scale.



Burgers equation

1D inviscid Burgers equation :

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = 0,$$

with a given initial condition $u(x,0)$ and periodic boundary conditions.

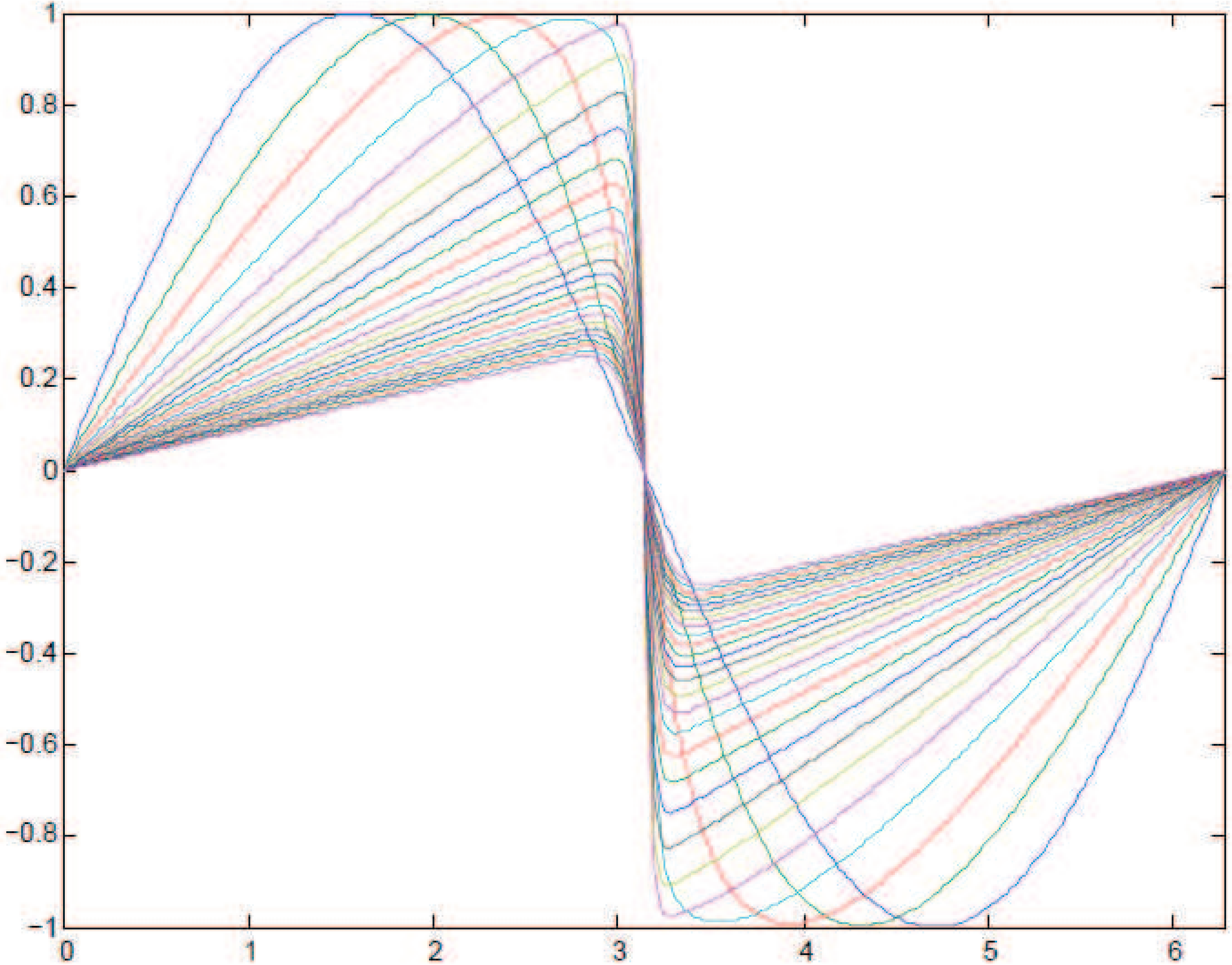
Diffusive BFN :

$$\begin{cases} \dfrac{\partial u_k}{\partial t} + \dfrac{1}{2}\dfrac{\partial u_k^2}{\partial x} = \nu \dfrac{\partial^2 u_k}{\partial x^2} + K(u_{obs} - H(u_k)), \quad 0 < t < T,\ 0 < x < L, \\ u_k(x,0) = \tilde{u}_{k-1}(x,0), \quad 0 < x < L, \end{cases}$$

$$\begin{cases} \dfrac{\partial \tilde{u}_k}{\partial t} + \dfrac{1}{2}\dfrac{\partial \tilde{u}_k^2}{\partial x} = -\nu \dfrac{\partial^2 \tilde{u}_k}{\partial x^2} - K'(u_{obs} - H(\tilde{u}_k)), \quad 0 < t < T,\ 0 < x < L, \\ \tilde{u}_k(x,T) = u_k(x,T), \quad 0 < x < L. \end{cases}$$

Numerical experiments

Inviscid Burgers equation : creation of shocks in finite time



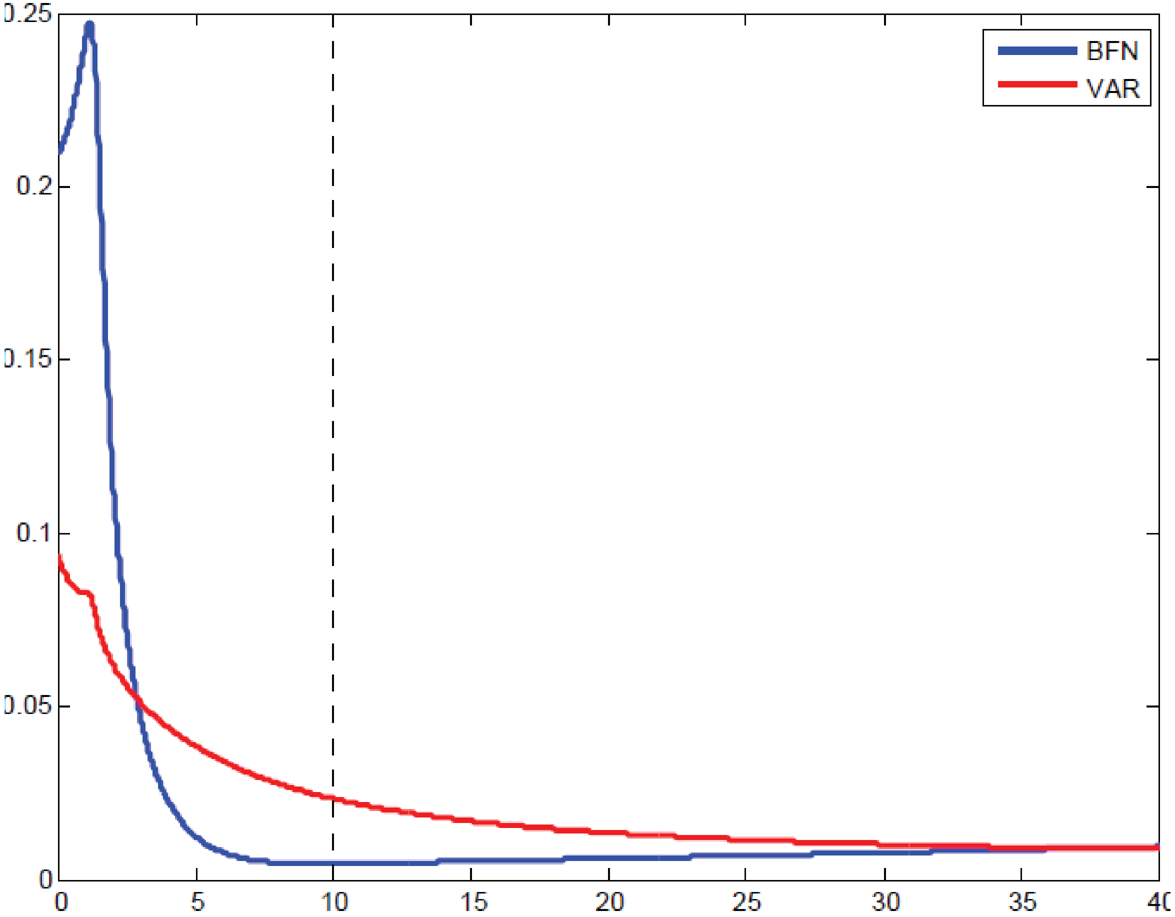
Numerical experiments

Comparison with a variational method :

		$n_x = 4$ $n_t = 4$ unnoisy	$n_x = 10$ $n_t = 10$ unnoisy	$n_x = 10$ $n_t = 10$ noisy (15%)
VAR	Number of iterations	18	20	15
	Relative RMS (%)	0.49	1.64	10.74
BFN2	Number of iterations	2	2	2
	Relative RMS (%)	0.34	0.69	3.50
		$(K = 30,\ K' = 60)$	$(K = 40,\ K' = 80)$	$(K = 10,\ K' = 20)$

Comparison between D-BFN and variational algorithms in the case of sparse and noisy observations on Burgers' equation with shock.

Numerical experiments



Forecast error (difference between the true trajectory and the solutions of the direct model initialized with the identified solutions) for D-BFN and VAR algorithms, with sparse ($n_x = 4 = n_t$) and noisy observations (15% noise).

Conclusions and perspectives

- Easy implementation (no linearization, no adjoint state, no minimization process)
- Very efficient in the first iterations (faster convergence)
- Lower computational and memory costs than other DA methods
- Stabilization of the backward model
- Excellent preconditioner for 4D-VAR (or Kalman filters)

Under investigation :

- Identification of the entire solution of a full primitive model from the knowledge of the SSH (or SST) only
- Efficient resolution of Riccati-like equations for a better backward nudging matrix