Didier Auroux
Mathematics Institute of Toulouse
University of Toulouse
France
auroux@mip.ups-tlse.fr

Jacques Blum
Dieudonné Laboratory
University of Nice
France
jblum@math.unice.fr

# Back and forth nudging algorithm for data assimilation problems

SPARC Data Assimilation Workshop, Fields Institute,
University of Toronto, Canada.

September 4-7, 2007.

# Talk overview

1. Description of the Back and Forth Nudging algorithm
2. Numerical convergence and comparison on Lorenz' equations
3. Numerical experiments on 1D-Burger's equation
4. Numerical experiments on a 3-layered Quasi-Geostrophic ocean model
5. Preliminary results on a Shallow Water model
6. Conclusions

# Motivations

Classical data assimilation schemes (e.g. 4D-VAR, Kalman-like filters/smoothers) : implementation/development cost.

Main difficulties : model linearization (tangent linear, adjoint models), optimization algorithm, error covariance matrices.

$\rightsquigarrow$ Idea : develop a simple and efficient DA scheme.

# The BFN Algorithm (Back and Forth Nudging)

# Forward nudging

Let us consider a model governed by an ODE :

$$\frac{dX}{dt} = F(X), \quad 0 < t < T,$$

with an initial condition $X(0) = x_0$.

$X_{obs}(t)$ : observations of the system
$H$ : observation operator.

$$\begin{cases} \dfrac{dX}{dt} = F(X) + K(X_{obs} - HX), & 0 < t < T, \\ X(0) = x_0, \end{cases}$$

where $K$ is the nudging (or gain) matrix.

# Forward nudging

- Meteorology : Hoke-Anthes (1976)
- Oceanography (QG model) : Verron-Holland (1989)
- Atmosphere (mesoscale) : Stauffer-Seaman (1990)
- Optimal determination of the nudging coefficients : Zou-Navon-Le Dimet (1992), Stauffer-Bao (1993)

# Backward nudging

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 :

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

# BFN algorithm

Iterative algorithm (forward and backward resolutions) :

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

$$\begin{cases} \dfrac{d\tilde{X}_k}{dt} = F(\tilde{X}_k) - K(X_{obs} - H\tilde{X}_k), \\ \tilde{X}_k(T) = X_k(T), \end{cases}$$

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

Proof of convergence in a simple case (linear model) : Auroux-Blum (2005)

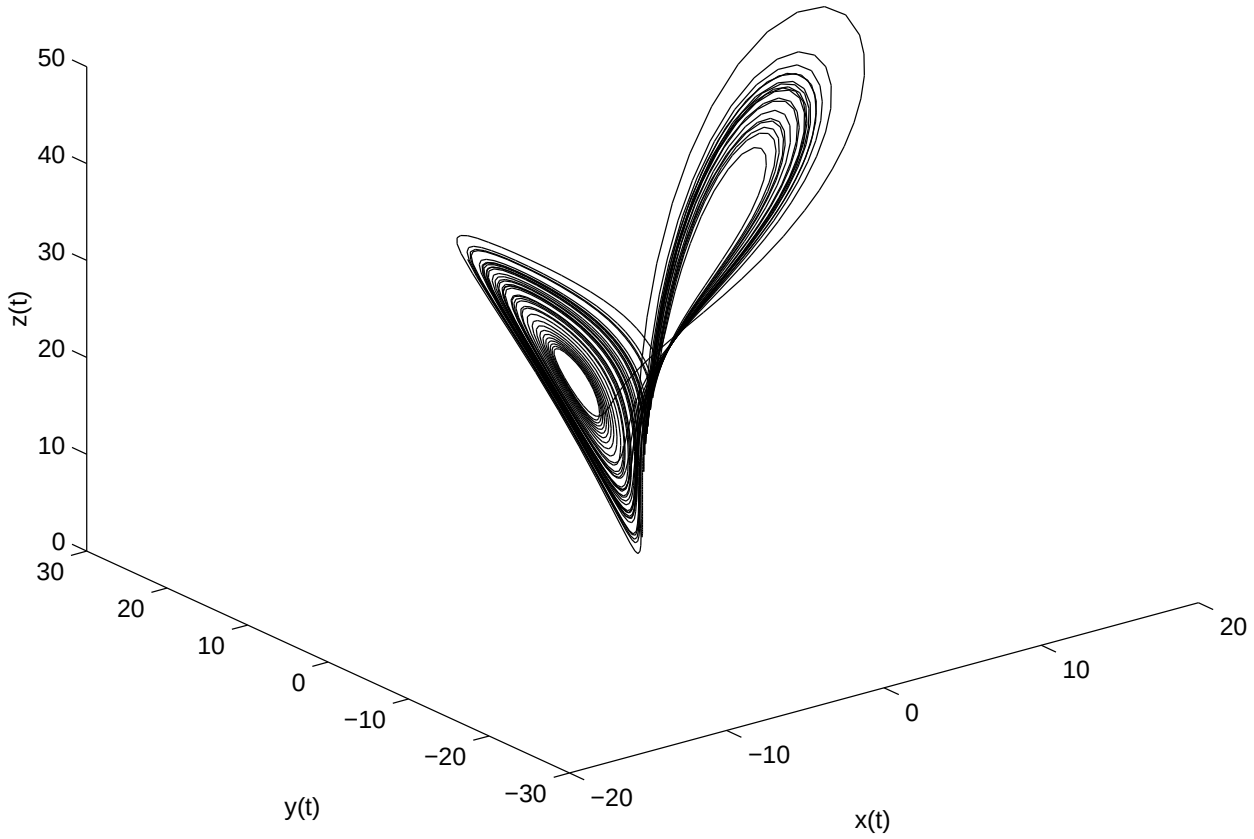
# Numerical Results
# Lorenz Equations

# Lorenz' equations

$$\begin{cases} \dfrac{dx}{dt} = 10\ (y - x), \\ \dfrac{dy}{dt} = 28\ x - y - xz, \\ \dfrac{dz}{dt} = -\dfrac{8}{3}\ z + xy. \end{cases}$$

- Assimilation period : $[0; 3]$, forecast : $[3; 6]$.
- Time step : 0.001.
- 31 observations (every 100 time steps).

# Lorenz' model



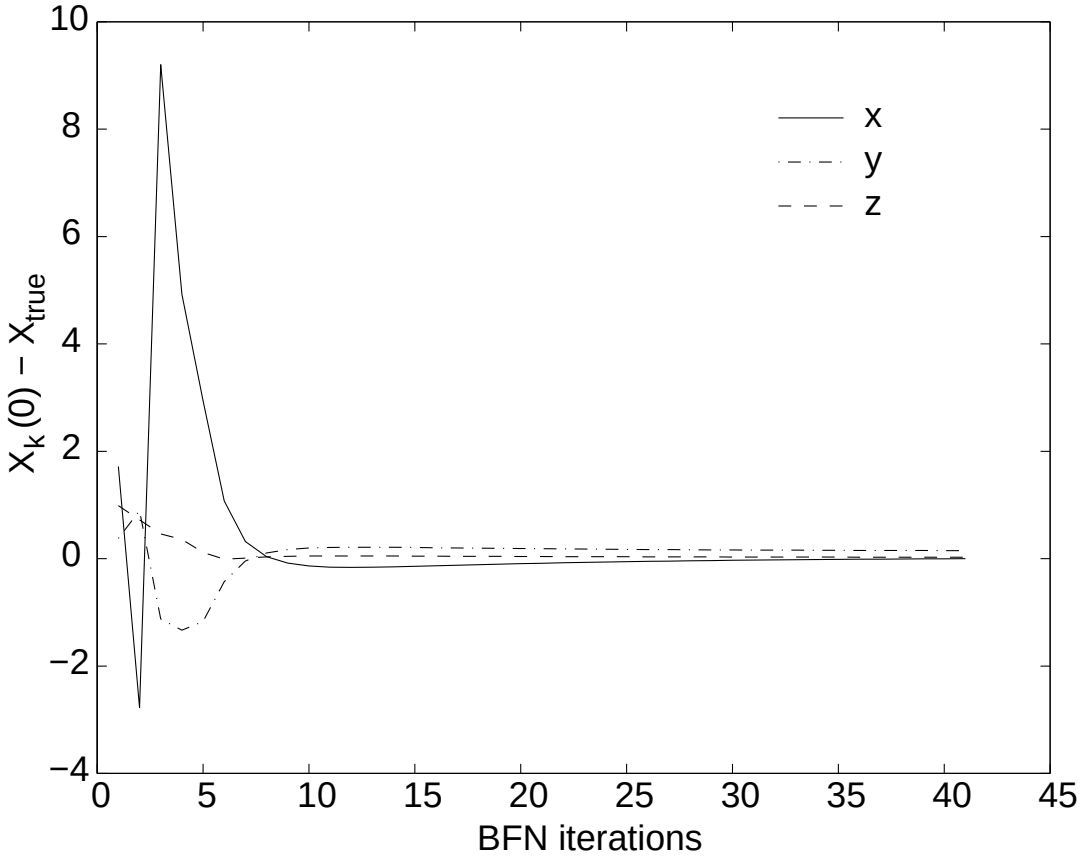



FIG. 1 – Difference between the k$^{th}$ iterate $X_k(0)$ and the exact initial condition $x_{true}$ for the 3 variables versus the number of BFN iterations.

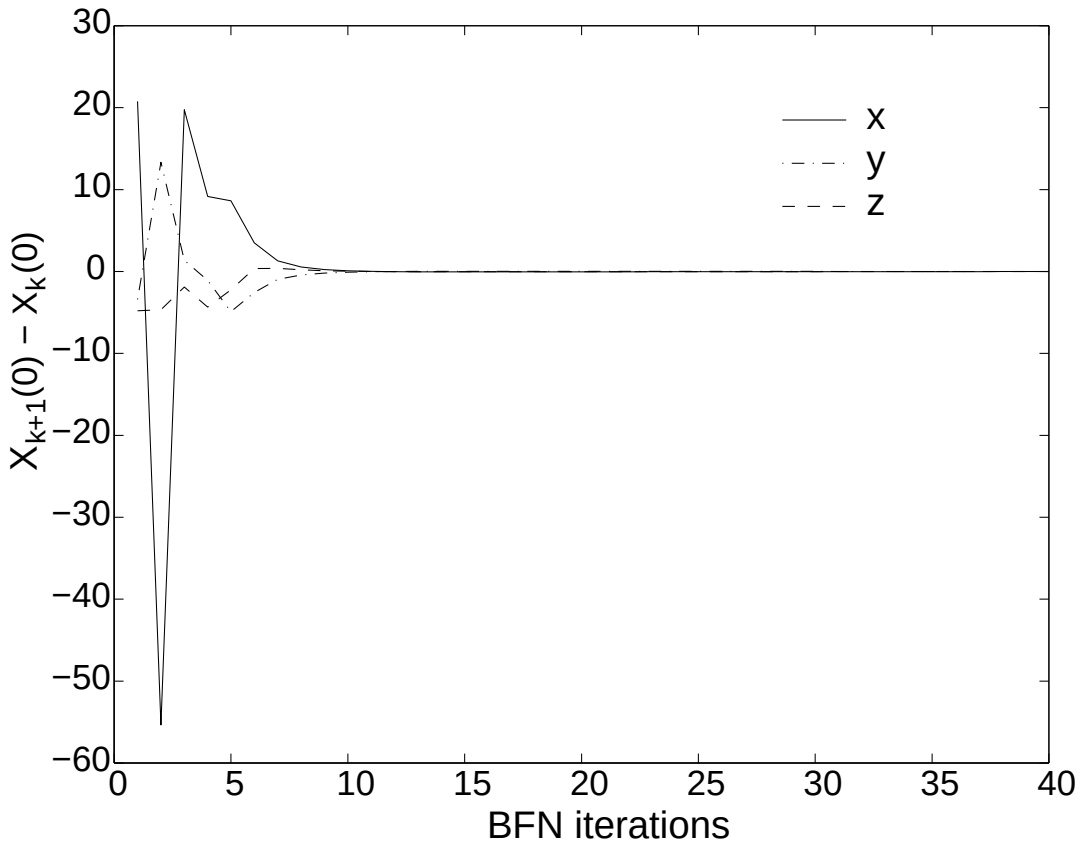



FIG. 2 – Difference between two consecutive BFN iterates for the 3 variables versus the number of BFN iterations.

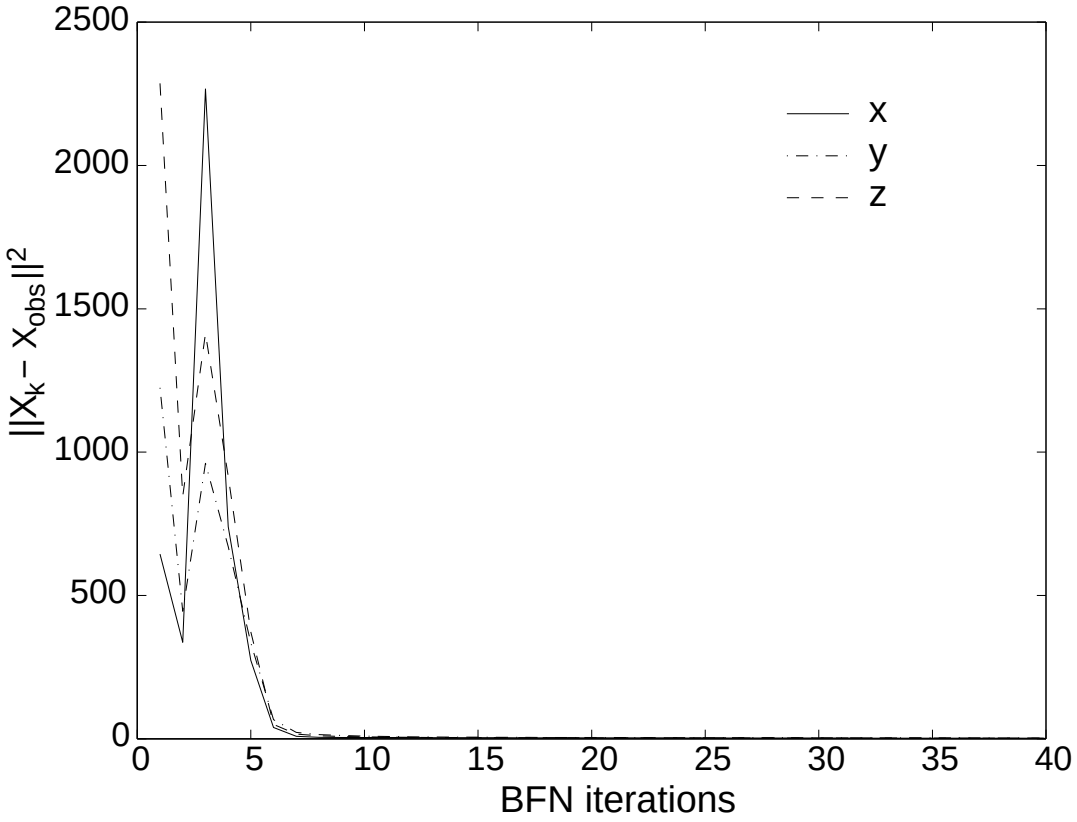



FIG. 3 – RMS difference between the observations and the BFN identified trajectory versus the BFN iterations.

# Comparison with 4D-VAR

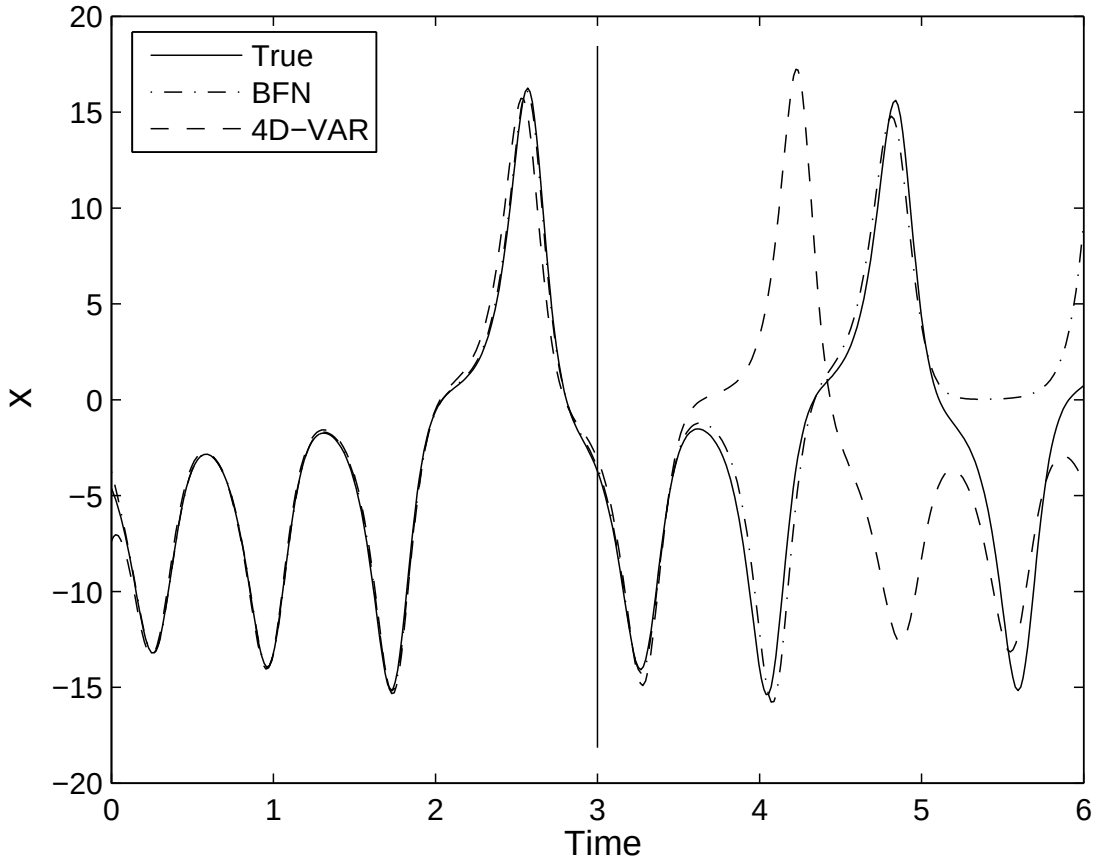



FIG. 4 – Evolution in time of the reference trajectory (plain line), and of the trajectories identified by the 4D-VAR (dashed line) and the BFN (dash-dotted line) algorithms, in the case of perfect observations and for the first Lorenz variable $x$.

# Comparison with 4D-VAR

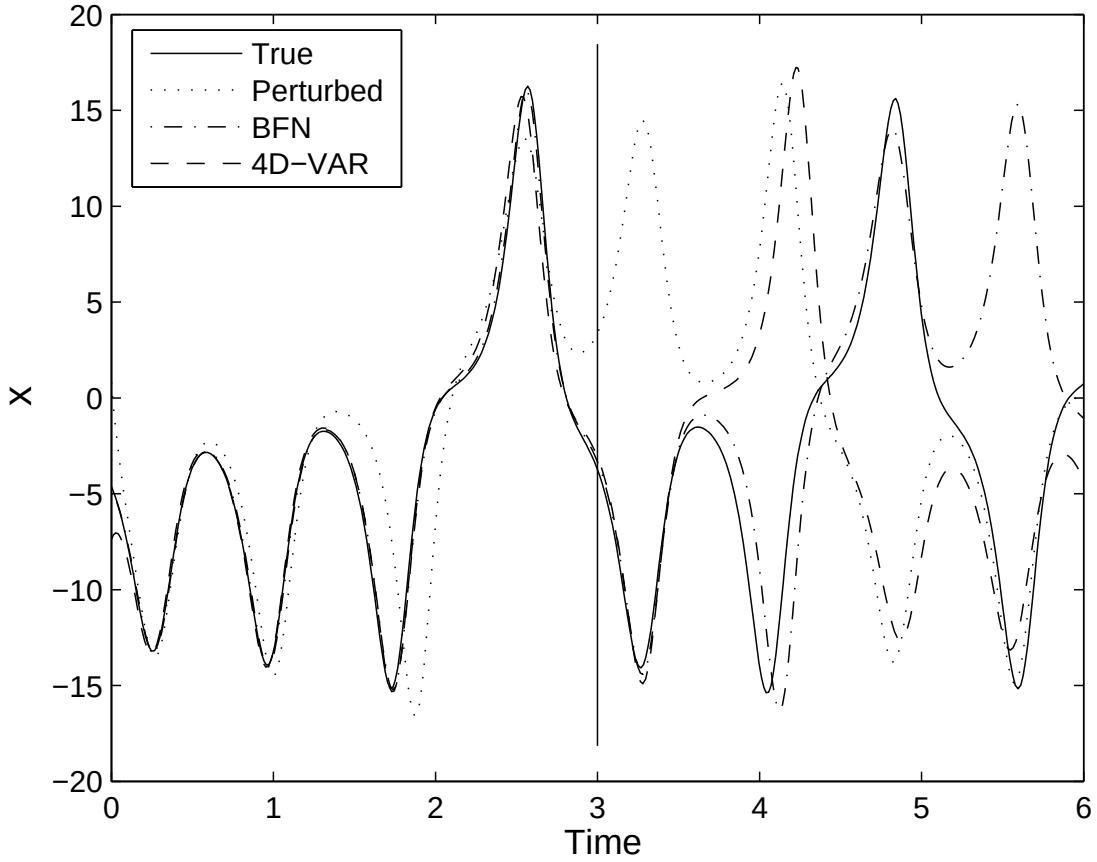



FIG. 5 – Evolution in time of the reference trajectories (plain line), and of the trajectories identified by the 4D-VAR (dashed line) and the BFN (dash-dotted line) algorithms, in the case of noised observations (with a 10% gaussian blank noise) and for the first Lorenz variable $x$.

# Numerical Results
# 1D-Viscous Burgers Equation

# 1D viscous Burgers' equation

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

where $X$ is the state variable, $s$ represents the distance in meters around the $45^o$N constant-latitude circle and $t$ is the time.

- Period of the domain : $28.3 \times 10^6 m$.
- Diffusion coefficient : $\nu = 10^5\ m^2.s^{-1}$.
- Time step : one hour.
- Assimilation period : one month (700 time steps).

Data : every 10 time steps (10 hours), every 5 gridpoints, 5% RMS blank gaussian error.

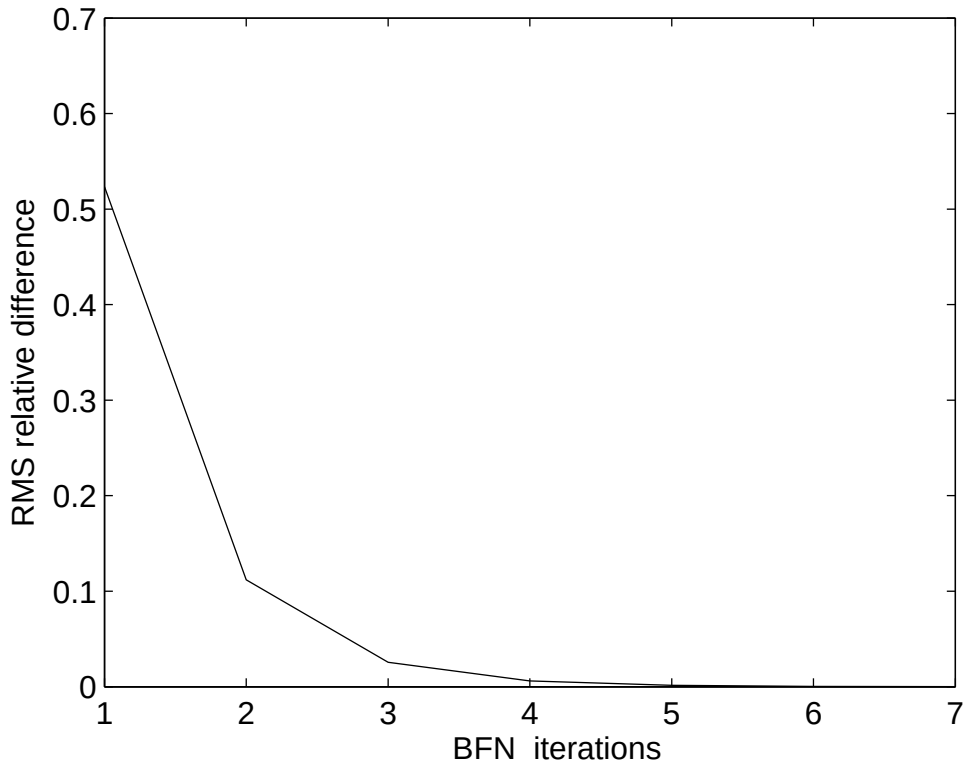



FIG. 6 – RMS relative difference between two consecutive iterates of the BFN algorithm versus the number of iterations.

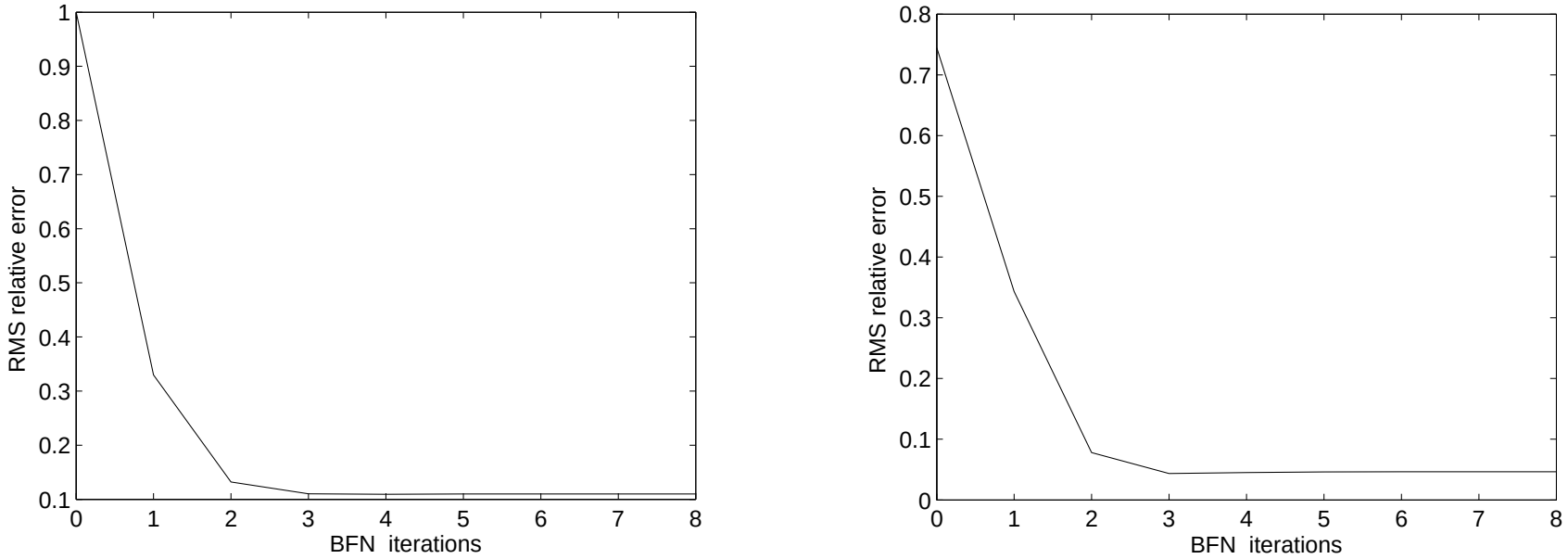



FIG. 7 – RMS relative difference between the BFN iterates and the exact solution versus the number of iterations, at time $t = 0$ (a) and at time $t = T$ (b).

# Comparison with 4D-VAR

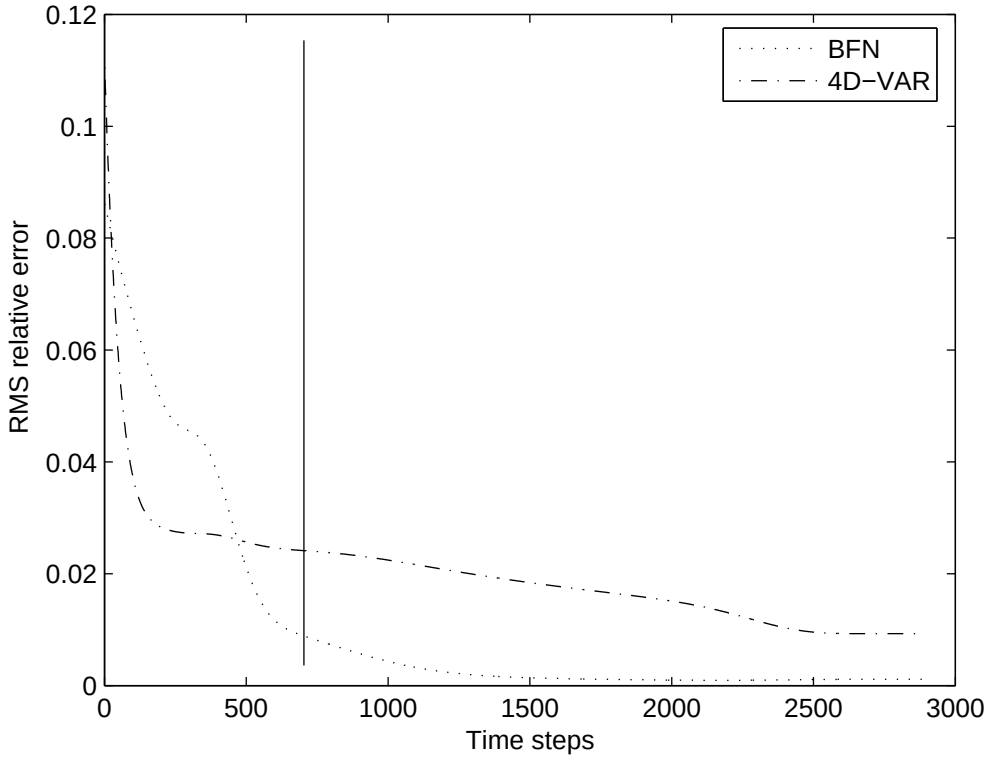



FIG. 8 – Evolution in time of the RMS difference between the reference trajectory and the identified trajectories for the BFN (dotted line) and the 4D-VAR (dash-dotted line) algorithms, in the case of perfect observations.

# BFN preprocessing

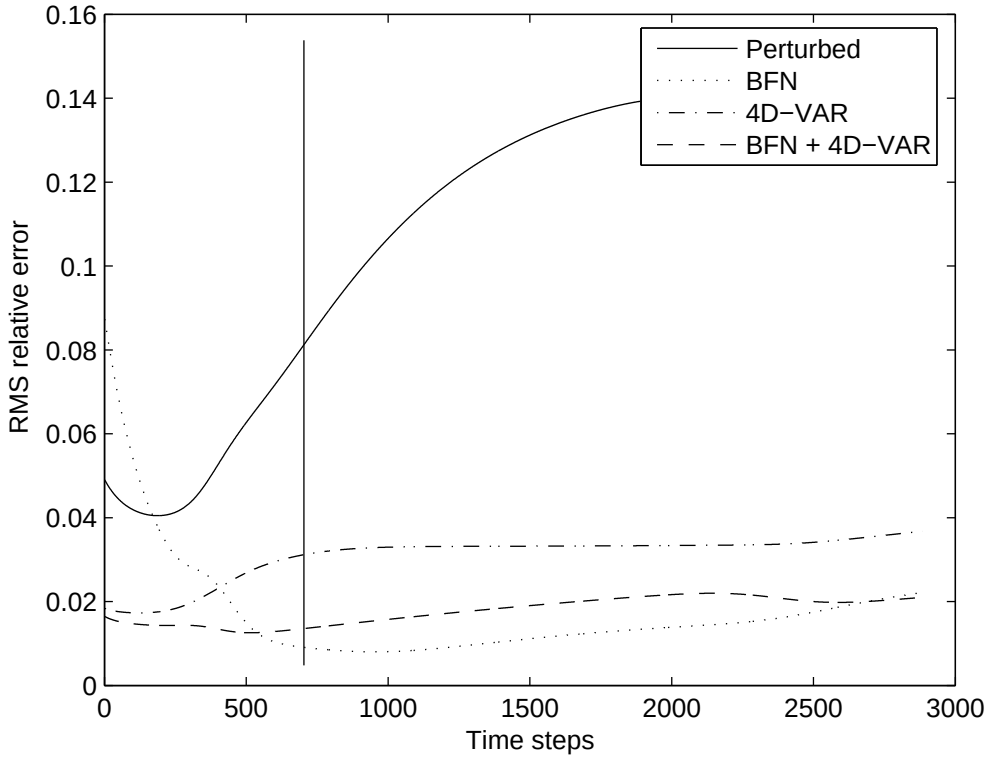



FIG. 9 – Evolution in time of the RMS difference between the reference trajectory and the identified trajectories for the BFN (dotted line), the 4D-VAR (dash-dotted line) and the BFN-preprocessed 4D-VAR (dashed line) algorithms, in the case of noised observations (with a 5% RMS error).

# Numerical Results : Layered Quasi-geostrophic Ocean Model

$$\frac{D_1\left(\theta_1(\Psi)+f\right)}{Dt}-\beta\Delta^2\Psi_1=F_1 \qquad \text{in } \Omega\times]0,T[, \tag{1}$$

at the surface layer $(k=1)$;

$$\frac{D_k\left(\theta_k(\Psi)+f\right)}{Dt}-\beta\Delta^2\Psi_k=0 \qquad \text{in } \Omega\times]0,T[, \tag{2}$$

at the intermediate layers $(k=2,\ldots,n-1)$;

$$\frac{D_n\left(\theta_n(\Psi)+f\right)}{Dt}+\alpha\Delta\Psi_n-\beta\Delta^2\Psi_n=0 \qquad \text{in } \Omega\times]0,T[, \tag{3}$$

at the bottom layer $(k=n)$.

# QG notations

- $\Omega \subset \mathbb{R}^2$ is the circulation basin and $]0, T[$ is the time interval,
- $n$ is the number of layers, $f$ is the Coriolis force,
- $\Psi_k$ is the stream function at layer $k$, $\Psi$ is the vector $(\Psi_1, \ldots, \Psi_n)^T$,
- $\theta_k$ is the sum of the dynamical and thermal vorticities at layer $k$ :

$$\theta_k(\Psi) = \Delta\Psi_k - (W\Psi)_k,$$

with $-(W\Psi)_k = \dfrac{f_0^2 \rho}{H_k g}\left(\dfrac{\Psi_{k+1} - \Psi_k}{\rho_{k+1} - \rho_k} - \dfrac{\Psi_k - \Psi_{k-1}}{\rho_k - \rho_{k-1}}\right)$.

- $g$ represents the constant of gravity, $\rho_k$ the fluid density at layer $k$ (and $\rho$ the average fluid density), and $H_k$ the depth of layer $k$,
- $\dfrac{D_k}{Dt}$ is the Lagrangian particular derivative : $\dfrac{D_k}{Dt} = \dfrac{\partial}{\partial t} + J(\Psi_k, .)$,

where $J$ is the Jacobian operator $J(f, g) = \dfrac{\partial f}{\partial x}\dfrac{\partial g}{\partial y} - \dfrac{\partial f}{\partial y}\dfrac{\partial g}{\partial x}$,

- $\Delta\Psi_n$ represents the bottom friction dissipation, $\Delta^2\Psi_k$ represents the lateral friction dissipation,
- and $F_1$ is the forcing term, the wind stress applied to the ocean surface.

# QG notations

- Mesh : $200 \times 200 \times 3$ grid ;
- Time step : 1.5 hour ;
- Assimilation period : 5 days ;
- Time/space data sampling : every 5 steps ;

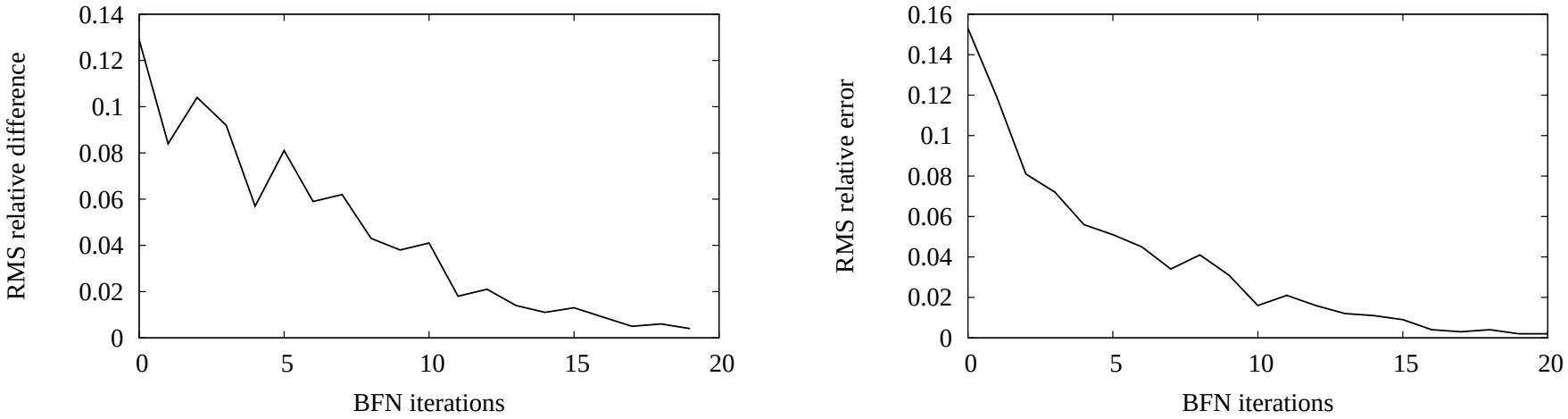



FIG. 10 – RMS relative difference between two consecutive BFN iterates (a) and between the BFN iterates and the exact solution (b) versus the number of BFN iterations.

# Comparison with 4D-VAR

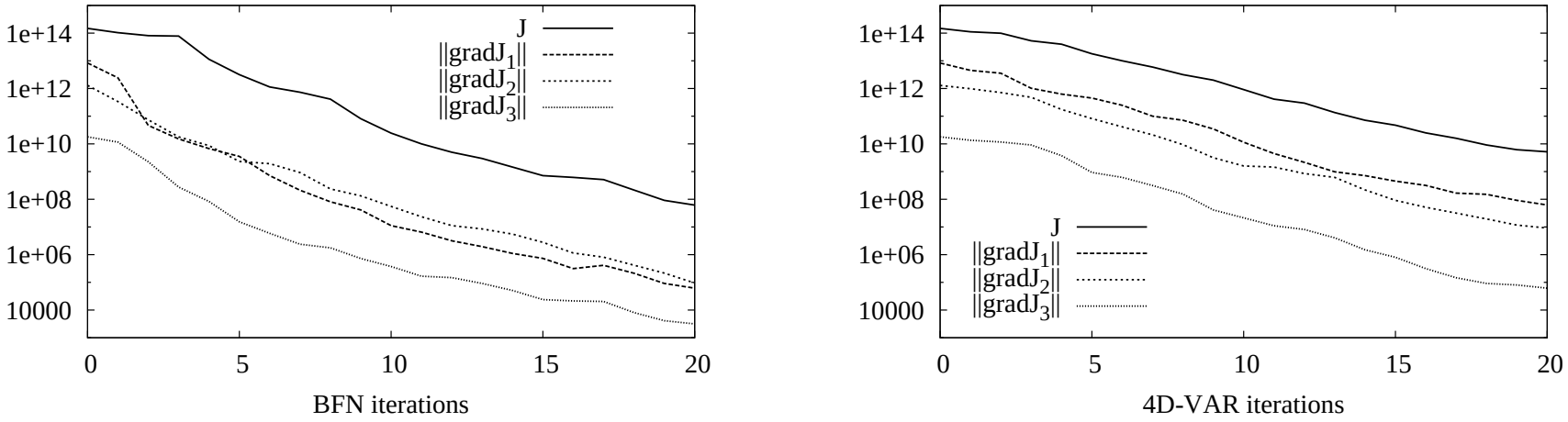



FIG. 11 – Evolution of the 4D-VAR cost function and of the 3 gradients of the cost function in the 3 ocean layers for the BFN iterates versus the number of BFN iterations (a) and for the 4D-VAR iterates versus the number of 4D-VAR iterations (b).

# Convergence - noised observations

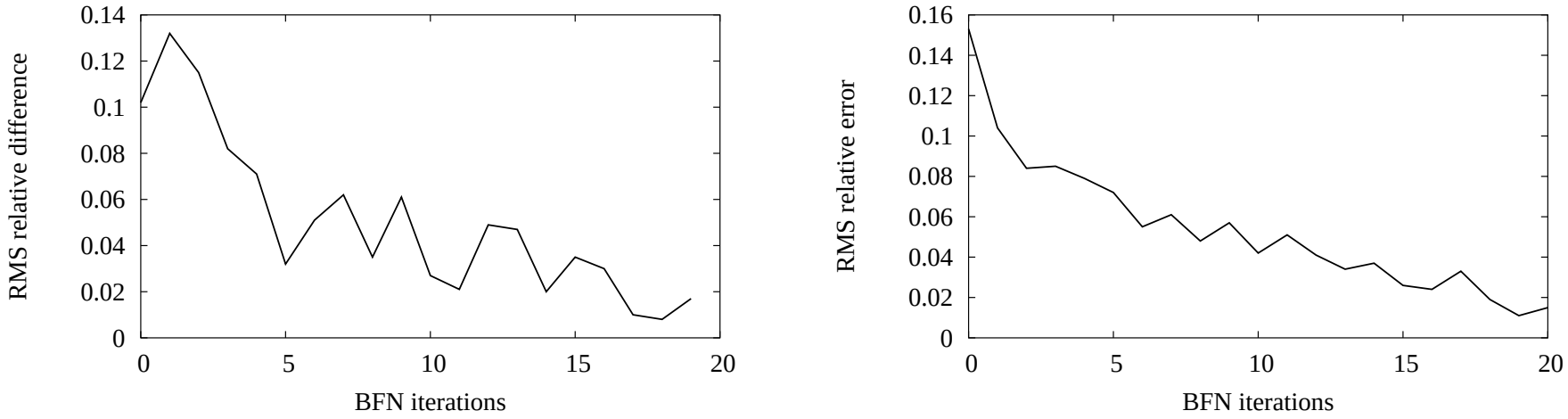



FIG. 12 – RMS relative difference between two consecutive BFN iterates (a) and between the BFN iterates and the exact solution (b) versus the number of BFN iterations in the case of noised observations.

# BFN vs 4D-VAR - noised observation

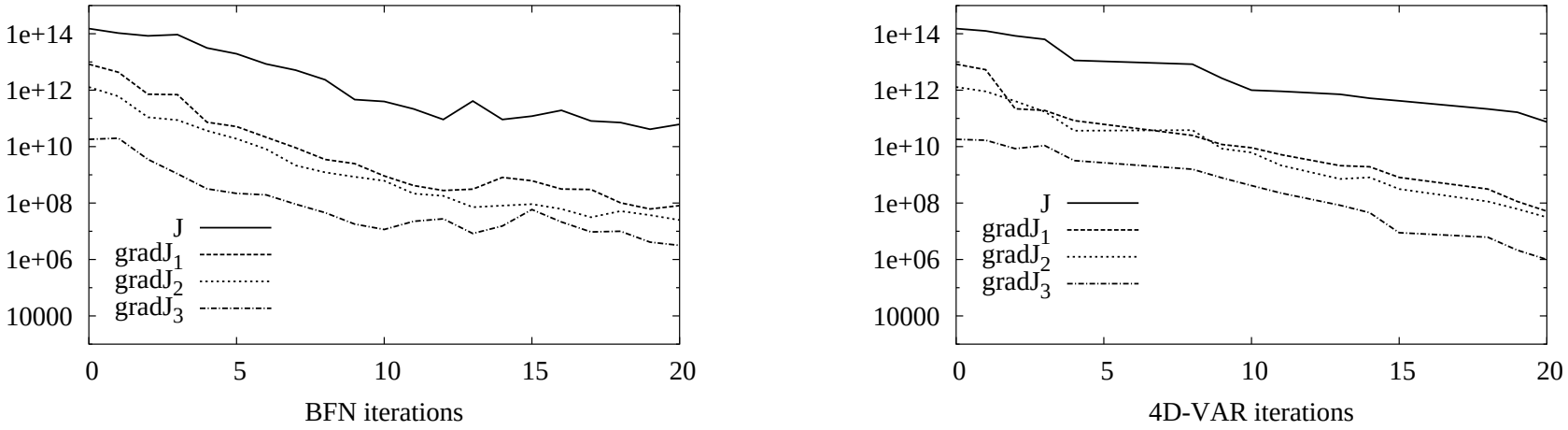



FIG. 13 – Evolution of the 4D-VAR cost function and of the 3 gradients of the cost function in the 3 ocean layers for the BFN iterates versus the number of BFN iterations (a) and for the 4D-VAR iterates versus the number of 4D-VAR iterations (b) in the case of noised observations.

# Forecast - noised observations

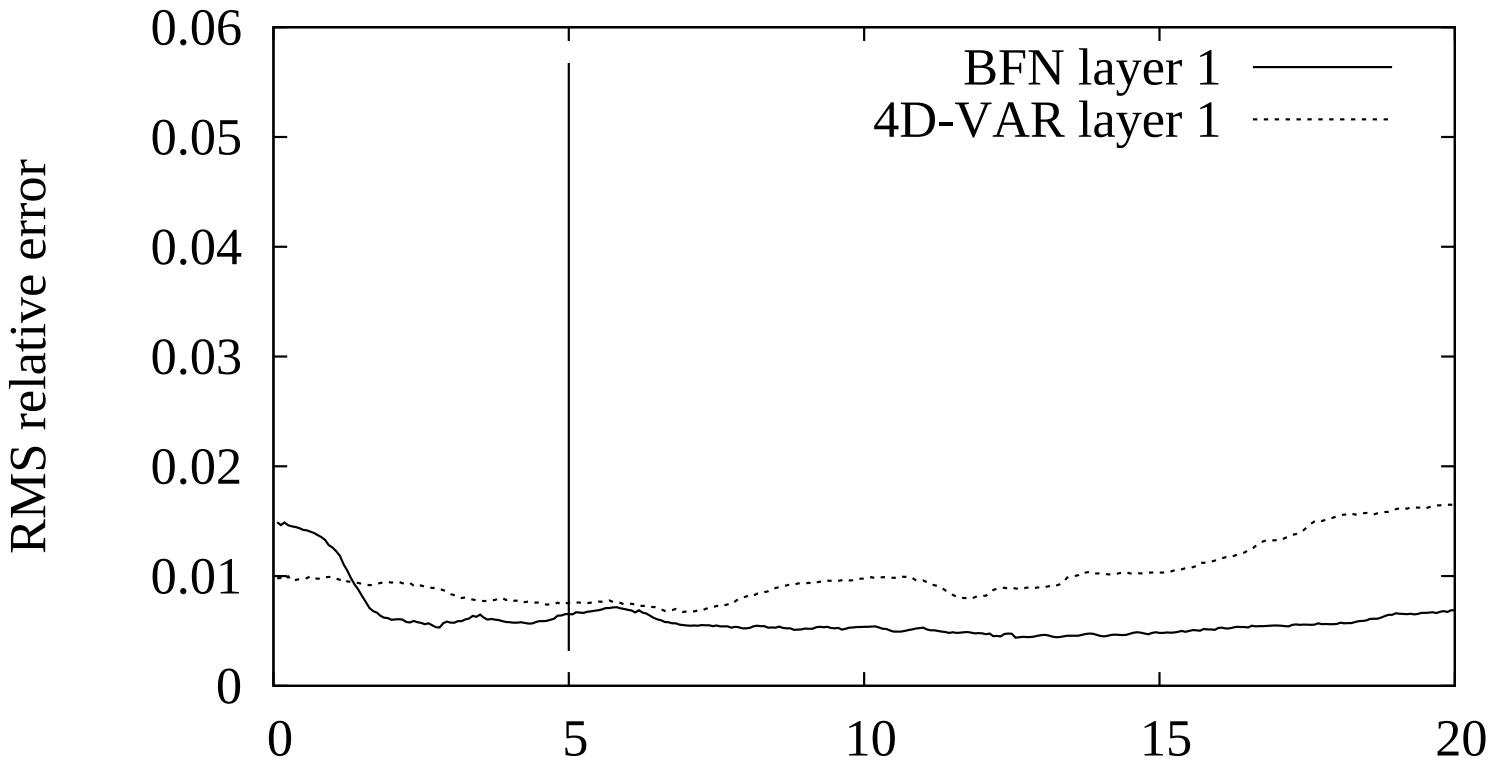



FIG. 14 – Evolution in time of the RMS relative difference between the reference trajectory and the identified trajectories for the BFN (plain line) and the 4D-VAR (dotted line) algorithms, versus time ; Surface layer.

# Forecast - noised observations

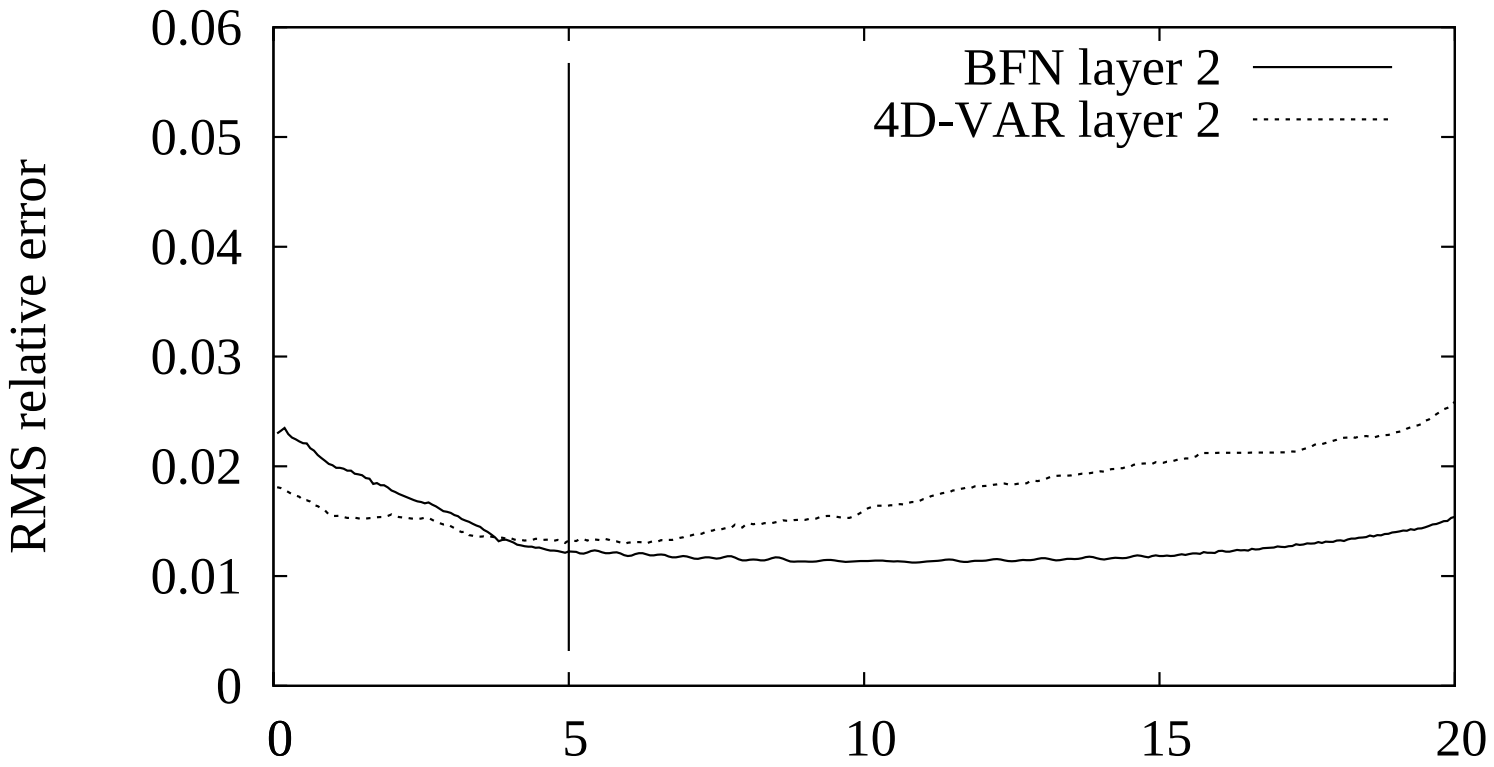



FIG. 15 – Evolution in time of the RMS relative difference between the reference trajectory and the identified trajectories for the BFN (plain line) and the 4D-VAR (dotted line) algorithms, versus time ; Intermediate layer.

# Forecast - noised observations

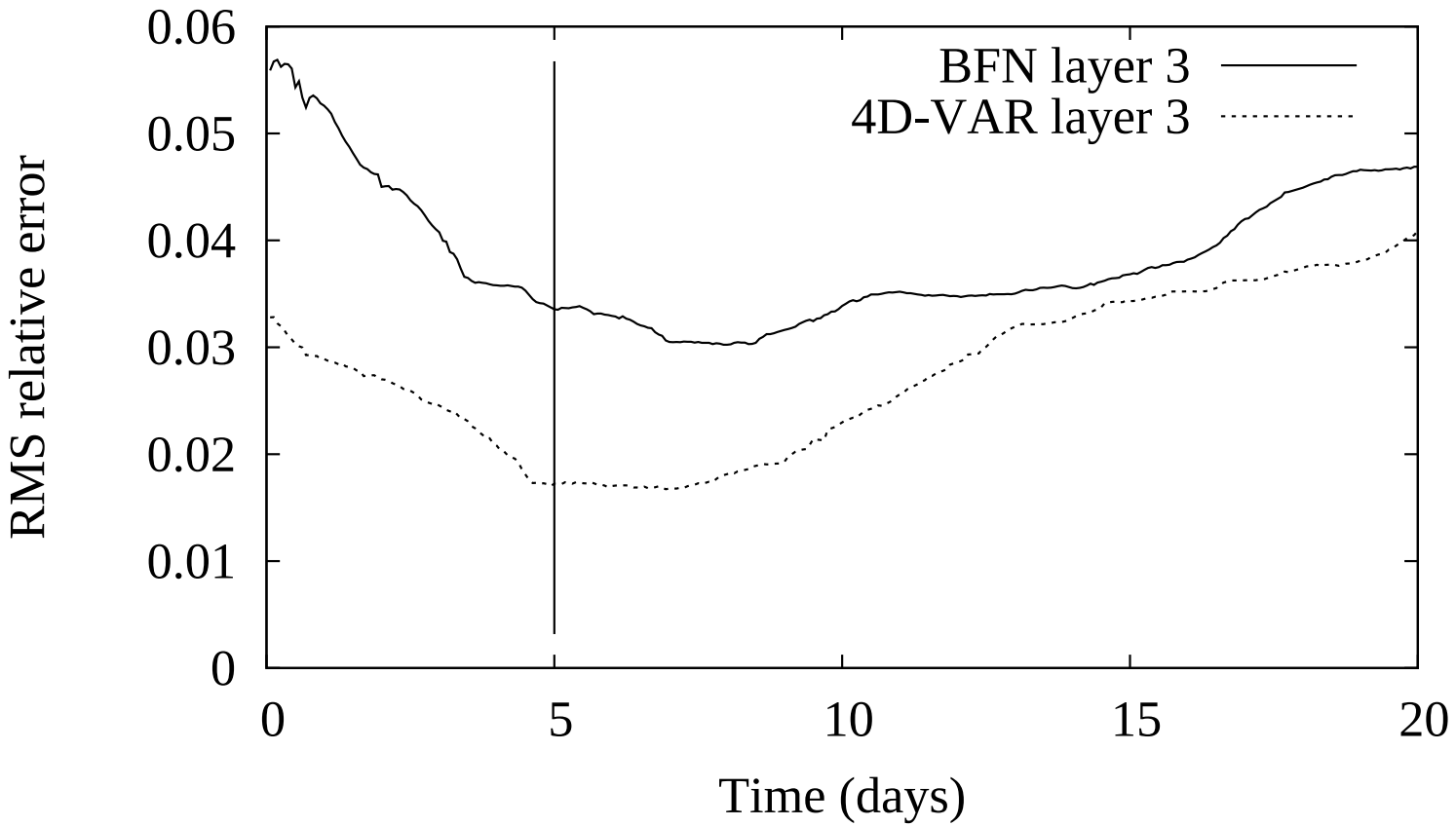



FIG. 16 – Evolution in time of the RMS relative difference between the reference trajectory and the identified trajectories for the BFN (plain line) and the 4D-VAR (dotted line) algorithms, versus time ; Bottom layer.

# Convergence - model error

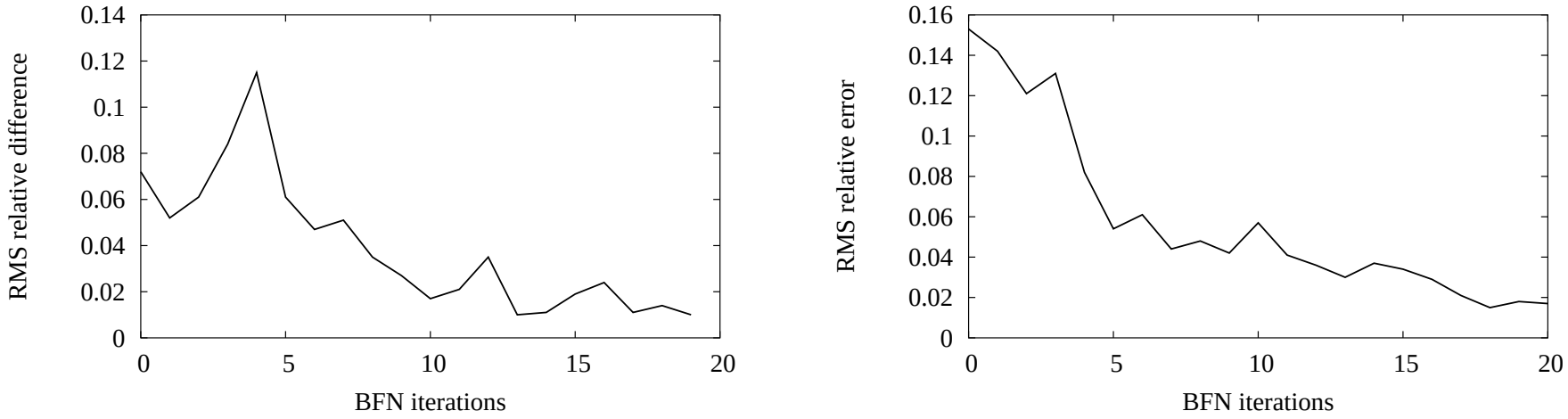



FIG. 17 – RMS relative difference between two consecutive BFN iterates (a) and between the BFN iterates and the exact solution (b) versus the number of BFN iterations in the case of imperfect model.

# BFN vs 4D-VAR - model error

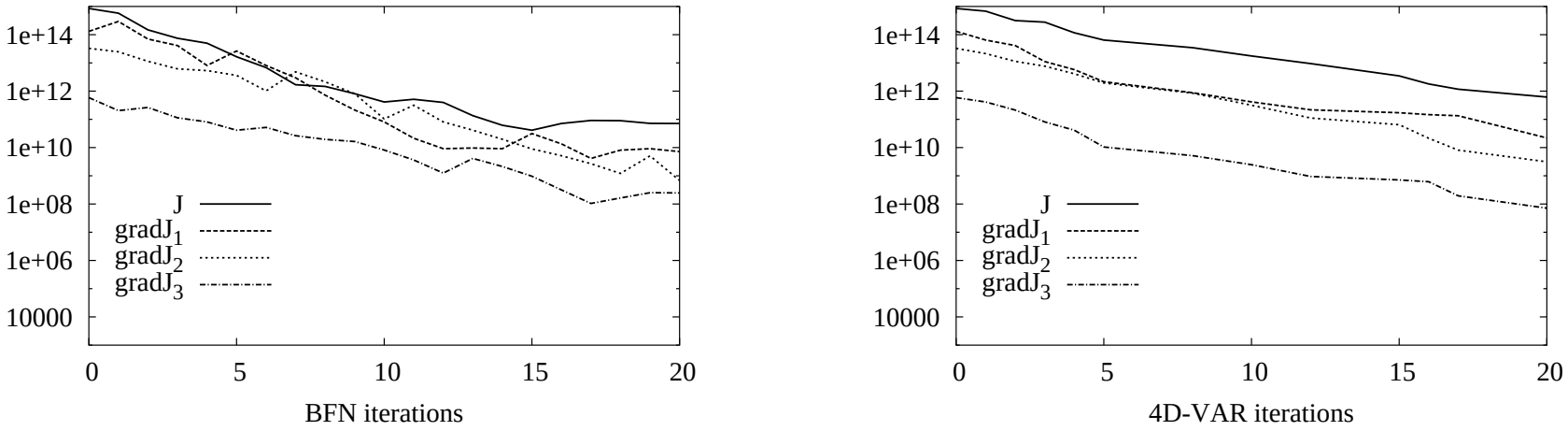



FIG. 18 – Evolution of the 4D-VAR cost function and of the 3 gradients of the cost function in the 3 ocean layers for the BFN iterates versus the number of BFN iterations (a) and for the 4D-VAR iterates versus the number of 4D-VAR iterations (b) in the case of imperfect model.

# Numerical Results
# Shallow Water Model

# 2D shallow-water model

$$\begin{cases} \partial_t u - (f+\zeta)v + \partial_x B = \dfrac{\tau}{\rho_0 h} - ru + \nu\Delta u, \\ \partial_t v + (f+\zeta)u + \partial_y B = \dfrac{\tau}{\rho_0 h} - rv + \nu\Delta v, \\ \partial_t h + \partial_x(hu) + \partial_y(hv) = 0. \end{cases}$$

$f$ : Coriolis force, $\zeta = \partial_x v - \partial_y u$ : vorticity, $B = gh + \frac{1}{2}(u^2+v^2)$ : Bernouilli potential, $r$ : bottom friction.

Domain : $L = 2.10^6$ m, $T = 1440 \times \delta t = 30$ days.

Observations of **h** only, every 24 time steps, **no observation error**.

# BFN data assimilation - h

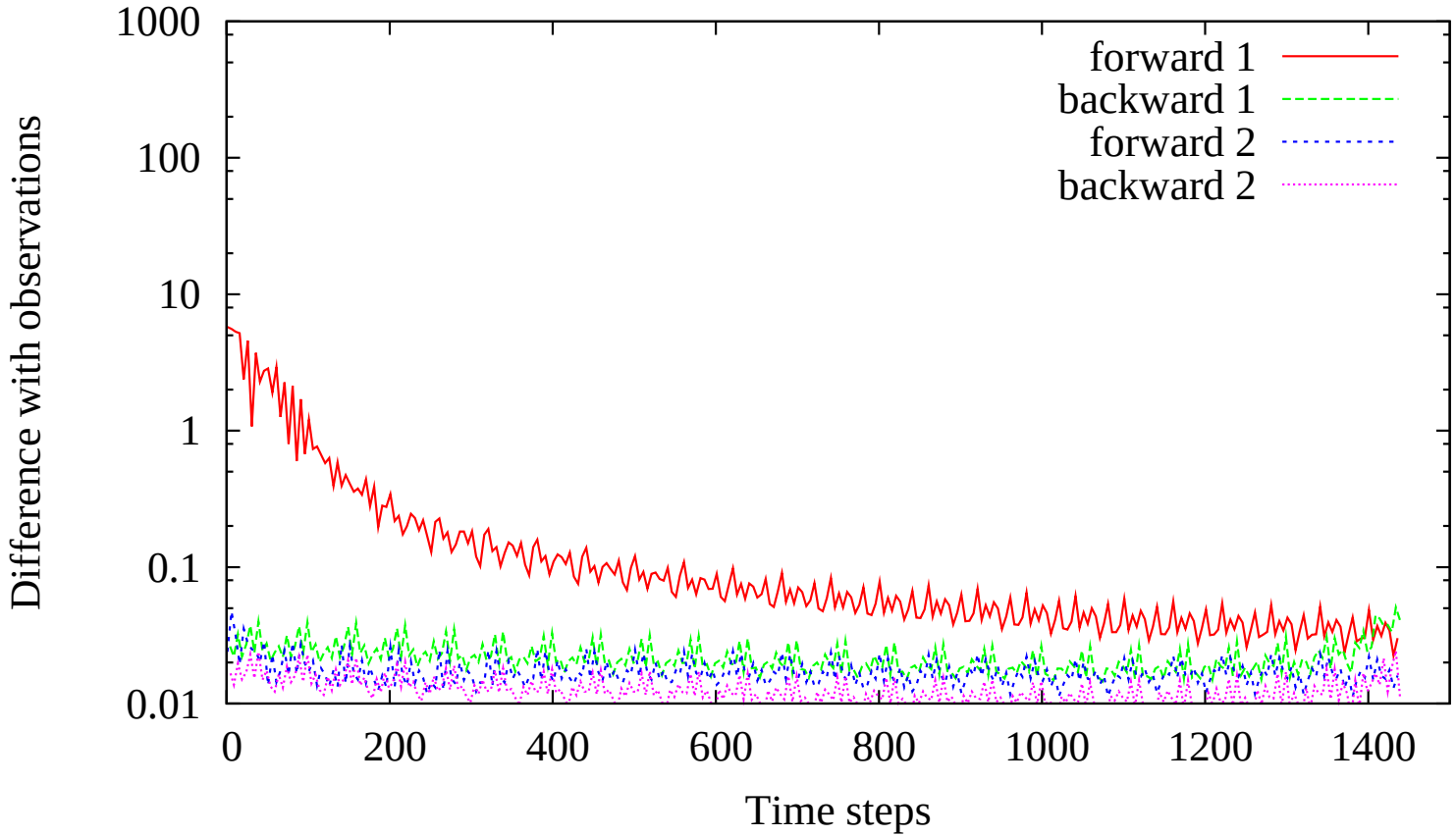



FIG. 19 – RMS difference between the backward nudging iterates and the true solution for the height **h**, versus the number of time steps.

# BFN data assimilation - u

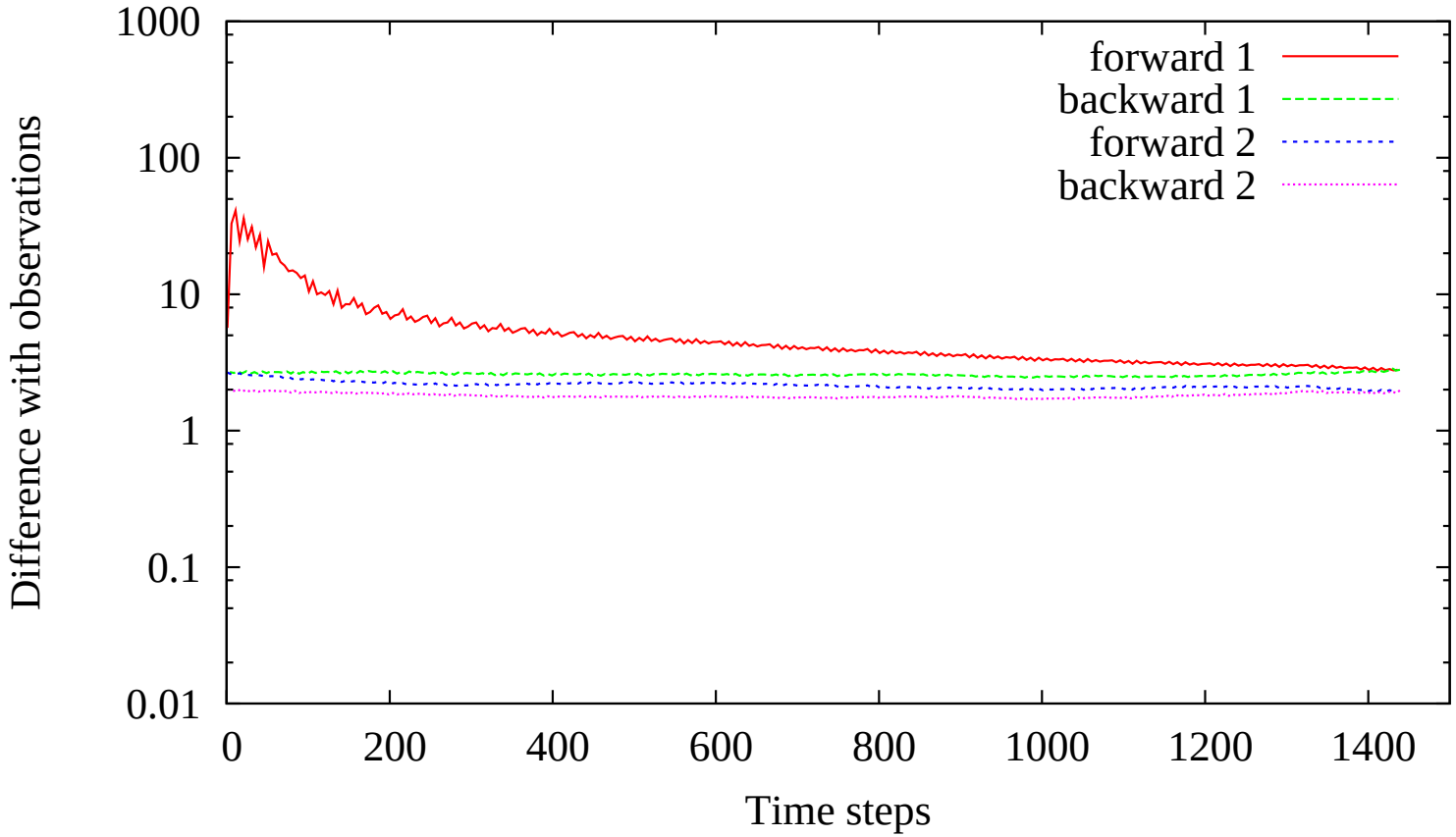



FIG. 20 – RMS difference between the backward nudging iterates and the true solution for the velocity **u**, versus the number of time steps.

# BFN data assimilation - v

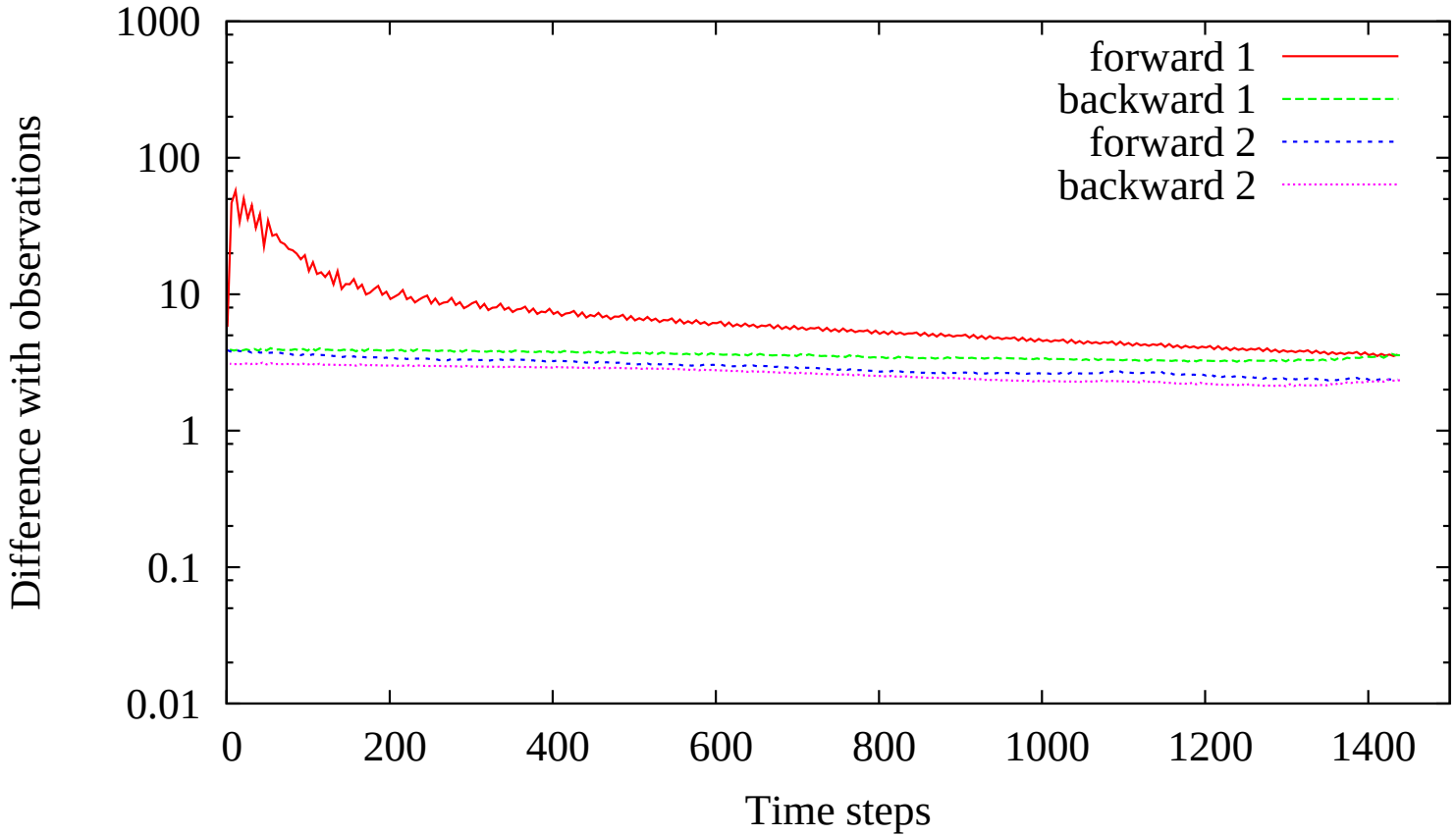



FIG. 21 – RMS difference between the backward nudging iterates and the true solution for the velocity $\mathbf{v}$, versus the number of time steps.

# BFN forecast - h

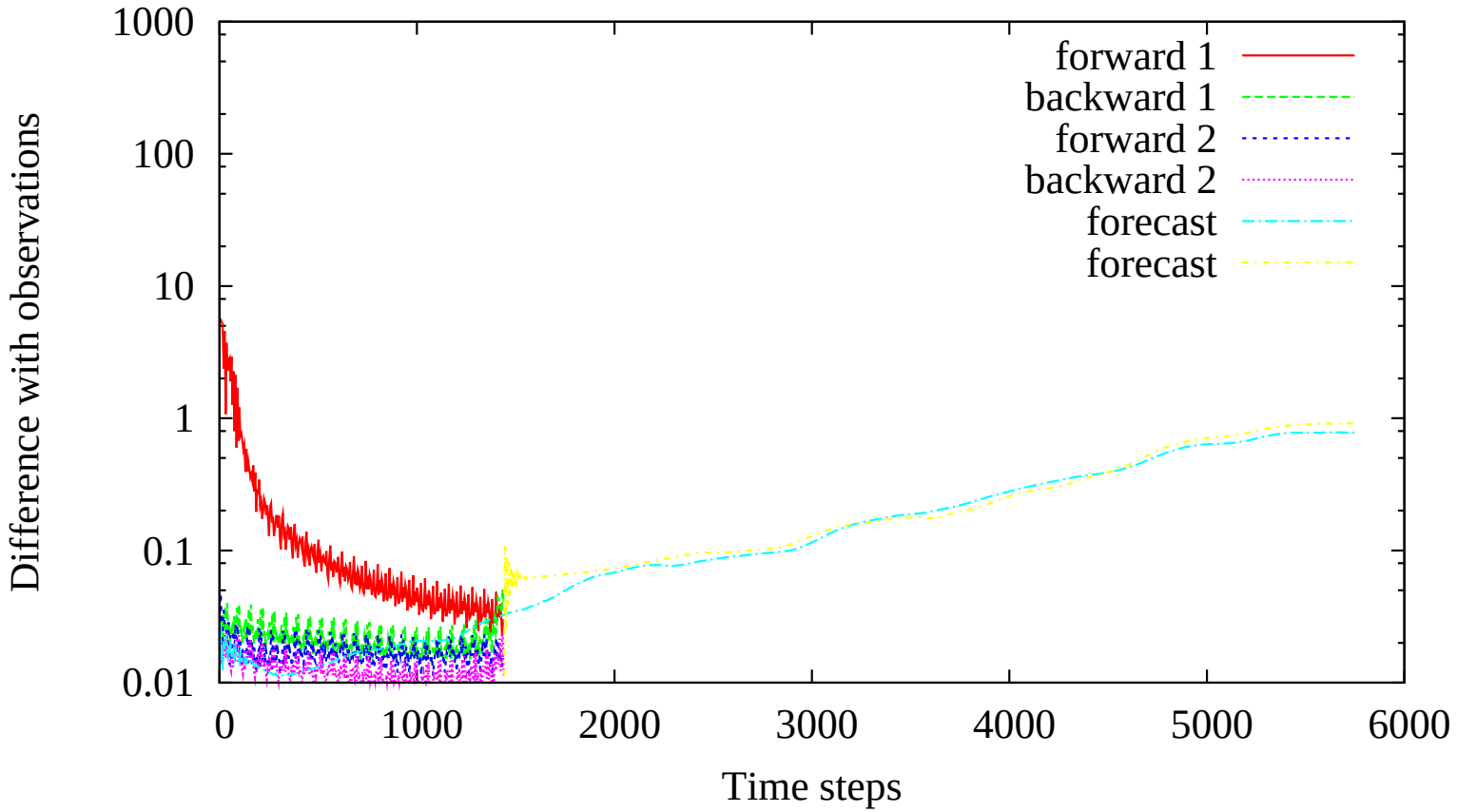



FIG. 22 – RMS difference between the backward nudging iterates and the true solution for the height **h**, and corresponding predictions using the last initial and final identified states, versus the number of time steps.

# BFN data assimilation - u

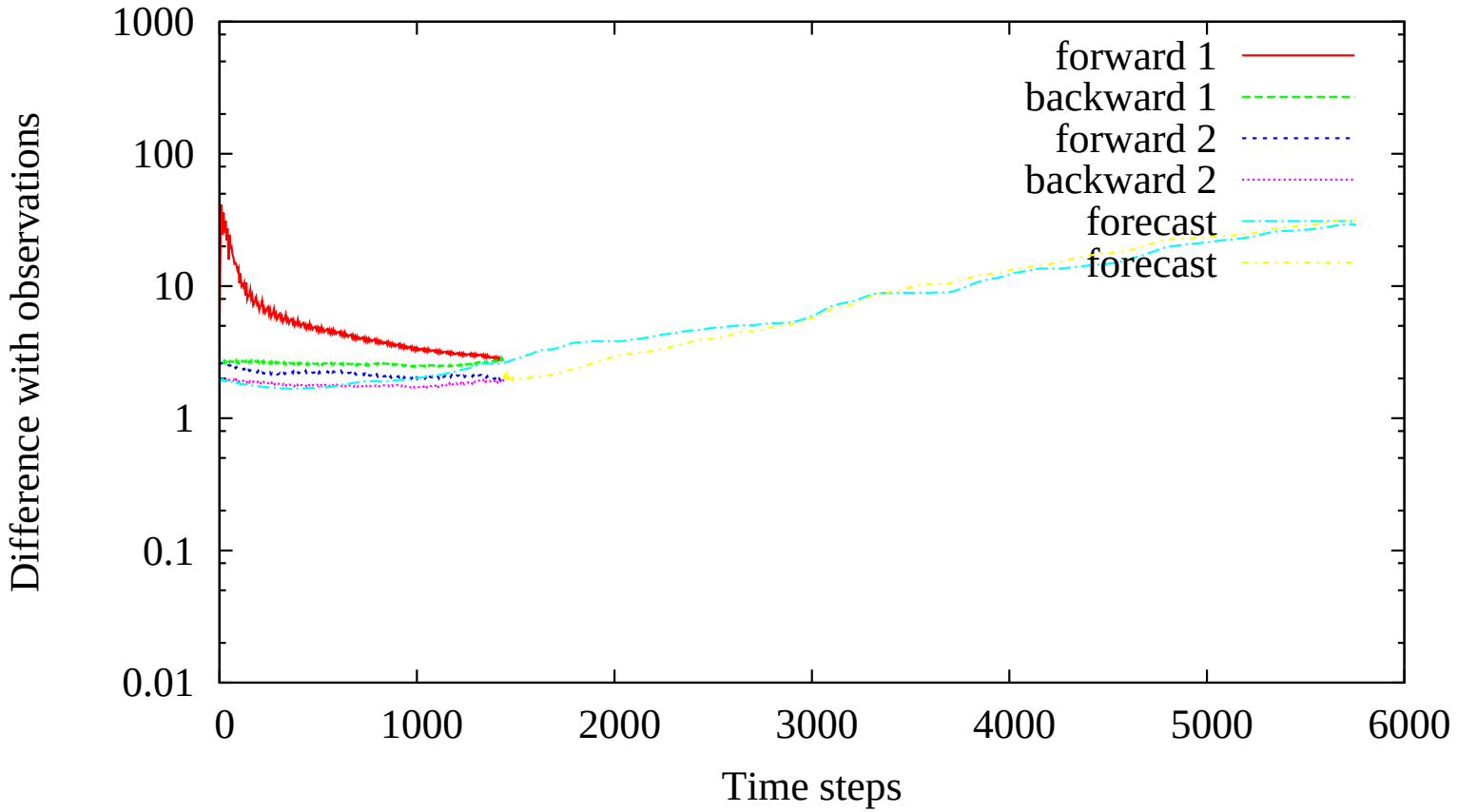



FIG. 23 – RMS difference between the backward nudging iterates and the true solution for the velocity **u**, and corresponding predictions using the last initial and final identified states, versus the number of time steps.

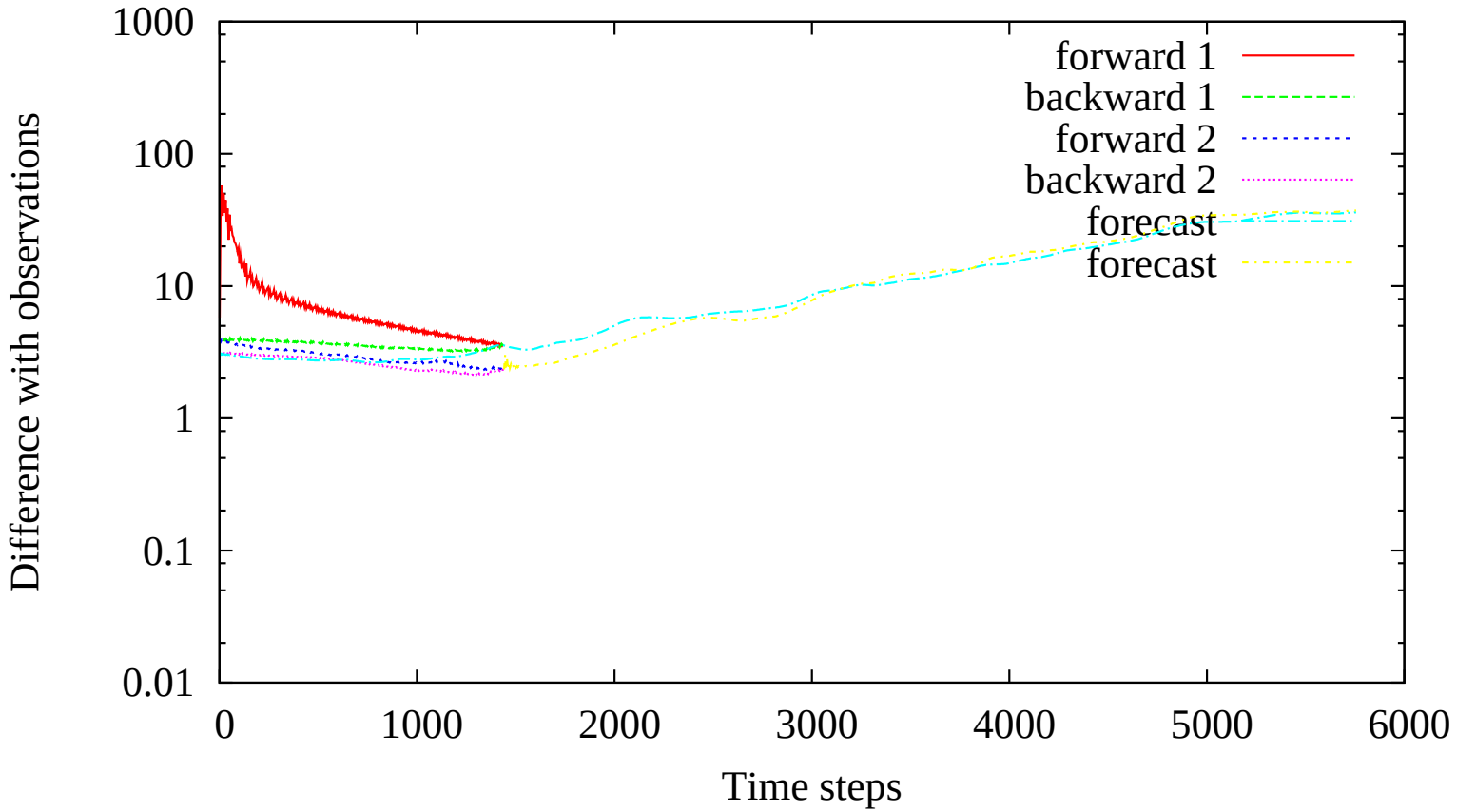



FIG. 24 – RMS difference between the backward nudging iterates and the true solution for the velocity $\mathbf{v}$, and corresponding predictions using the last initial and final identified states, versus the number of time steps.

# BFN data assimilation and forecast

- True solution
  - Height **h**
  - Velocity **u**
  - Velocity **v**
- BFN scheme
  - Height **h**
  - Velocity **u**
  - Velocity **v**

Domain : $L = 2.10^6$ m, $T = 1440 \times \delta t = 30$ days.

Observations of **h** only, every 24 time steps, no observation error.

2 BFN iterations (forward-backward resolutions) on $[0; T]$, forecast on $[0; 4T]$.

# Conclusions

# Conclusions

- Easy implementation (no linearization, no adjoint state, no minimization process)
- Very efficient on the first iterations
- Lower computational and memory costs than 4D-VAR
- Works well on several geophysical models.

  $\Longrightarrow$ Could also be an excellent preconditioner for 4D-VAR

This work has been performed within the INRIA Rhône-Alpes MOISE project, and supported by the INSU-LEFE BFN grant.

# Thank you for your Attention