

**18th International Conference on
Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes
9-12 October 2017, Bologna, Italy**

**DATA ASSIMILATION AT LOCAL SCALE TO IMPROVE CFD SIMULATIONS OF
DISPERSION AROUND INDUSTRIAL SITES AND IN URBAN NEIGHBOURHOODS**

Cécile L. Defforge¹, Marc Bocquet¹, Raphaël Bresson¹, Patrick Armand², and Bertrand Carissimo¹

¹CEREA, Joint laboratory École des Ponts ParisTech / EDF R&D, Université Paris-Est, Marne-la-Vallée, France

²CEA, DAM, DIF, F-91297 Arpajon, France

Abstract: Wind fields around industrial sites and in urban neighbourhoods have very complex structures, which are sensitive to geometrical features such as topography and buildings. These wind field structures are difficult to simulate with CFD models. Yet, these simulations are important to address various issues related to micrometeorology and dispersion of pollutants. To perform small scale simulations, CFD models use inputs (initial and boundary conditions) that usually are meteorological data obtained from measurements or larger-scale model outputs. These data often lack precision, may not contain all necessary information, and are not adapted to the detailed features of local scale, especially the topography and the presence of buildings. A few measurements inside the domain, although very local, have the potential to greatly enhance the precision of the simulations and thus the prediction of pollutants concentrations. If some concentration measurements are available, they can also be used to improve the simulations. Using measurements to improve the estimation of the system state is the goal of data assimilation. Data assimilation techniques developed so far in meteorology are generally applied to larger scale simulations that are mainly driven by initial conditions and deal with simple geometries without obstacles. The present work aims at developing local-scale data assimilation techniques that focus on boundary conditions rather than initial conditions and may deal with very complex geometries. Two data assimilation techniques (back and forth nudging algorithm and iterative ensemble Kalman smoother) are evaluated at small scale, at first, with a simple flow, using 1D/2D solutions of the shallow water equations. This simple test-case allows in particular to verify that small-scale simulations are more sensitive to errors on boundary conditions than errors on initial conditions. The data assimilation techniques are adapted to take boundary conditions into account and their performance and limitations are compared using the simple case described above.

Key words: *Data assimilation, local scale simulation, boundary conditions, shallow water model, back and forth nudging algorithm, iterative ensemble Kalman smoother.*

INTRODUCTION

Context

Atmospheric dispersion modelling requires meteorological inputs, such as wind and turbulence fields, over local domains with complex topographies (e.g., urban neighbourhoods or surroundings of industrial sites). The local wind fields over these domains are difficult to simulate with CFD models because of their complex structures which are sensitive to geometrical features (topography, building set up, etc.). To perform small scale simulations, CFD models — for instance the atmospheric version of *Code_Saturne* (Archambeau et al., 2004) — use inputs (initial and boundary conditions) that are generally meteorological data obtained from measurements or from larger-scale model outputs. These external large scale data are generally imprecise and might not be adapted to the detailed features of local scale.

A few measurements inside the domain, although very local and potentially perturbed by, for instance, the buildings, have the potential to greatly enhance the precision of the simulations. Using measurements (e.g., wind velocity, concentration, etc.) to improve the estimation of the system state is the goal of data assimilation.

Data assimilation techniques developed so far in meteorology (Kalnay, 2003; Asch et al., 2016) are generally applied to larger scale simulations that are mainly driven by initial conditions (IC) and deal with simple geometries without obstacles. The theoretical analysis of shallow water equations — approximation of Navier-Stokes equations for shallow flows — shows how small scale simulations are dominantly influenced by boundary conditions (BC). Consequently, the present work aims at developing local-scale data assimilation techniques that focus on BC rather than IC and may deal with very complex geometries.

Introduction to data assimilation

The goal of data assimilation is to compute the best estimate of some control variables using all available information (model, observations, and prior information).

One of the first and most basic data assimilation method is *nudging*, which consists in adding a relaxation term to the dynamical equations (Kalnay, 2003; Asch et al., 2016). The feedback term is proportional to the distance between the observations and the projection of the system state into the observation space. An improvement of this method and a potential substitute for 4D variational data assimilation has been recently developed: the back and forth nudging (BFN) algorithm (Auroux and Blum, 2005, 2008; Auroux et al., 2013). It has been tested and proved to converge on simple cases of shallow water model (Auroux, 2009), diffusion (Auroux et al., 2011), and transport (Auroux and Nodet, 2012).

Apart from nudging, data assimilation techniques developed so far and operationally implemented are generally divided into two classes: statistical (or filtering methods) and variational methods.

Recently, hybrid methods such as the iterative ensemble Kalman filter and smoother (IEnKF and IEnKS) (Sakov et al., 2012; Bocquet and Sakov, 2014) have been developed. These methods are based on the iterative minimisation of a cost function using Gauss-Newton method. In the linear model case, the filtering (i.e. at the end of the data assimilation time window) solution is equivalent to that obtained with the EnKF.

Both the BFN algorithm and the IEnKS have the great advantage to avoid using the model adjoint and its tangent linear, and to handle non-linear analyses. Consequently, they have been adapted in the present study so as to include boundary conditions in the control variables.

METHODS

Shallow water model

The shallow water equations (SWE) are derived from Navier-Stokes equations in the approximation of small fluid height compared to horizontal spatial scales. These equations can be used to represent atmospheric flows if the gravity g is replaced by the reduced gravity $g' = \frac{\rho - \rho_a}{\rho} g$ to account for the small difference of density between the simulated boundary layer (ρ) and the free atmosphere above (ρ_a). Moreover, the often used discretised 'level' models — which use vertical finite-difference approximations — to simulate continuously stratified fluid, are equivalent to multi-layer models — where SWE are applied to each layer. Therefore the simple model described below can be considered as a crude vertical description of the atmosphere into two layers. This equivalence between 'level' models and layer models, together with the relative simplicity of the SWE, motivate the use of a shallow water model to first test the data assimilation methods for atmospheric local scale.

In one dimension, the two state variables involved in the SWE are the thickness of the fluid layer (h) (here, the boundary layer), and the mean horizontal velocity (u) which are related as follows:

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbf{M} \frac{\partial \mathbf{X}}{\partial x} = \mathbf{S} \quad (1)$$

where $\mathbf{X} = \begin{pmatrix} h \\ u \end{pmatrix}$, $\mathbf{M} = \begin{pmatrix} u & h \\ g' & u \end{pmatrix}$, and $\mathbf{S} = \begin{pmatrix} 0 \\ -g' \frac{\partial z_f}{\partial x} \end{pmatrix}$ with z_f the bottom topography. The problem is well-posed if two boundary conditions are prescribed, one on h and one on u . For subcritical flows, it is necessary to impose a boundary condition on each side of the domain. In what follows, we thus prescribe the velocity on the left of the domain: $u(x=0) = u_L$, and the fluid height on the right: $h(x=L) = h_R$. We consider the steady state obtained with these two boundary conditions.

Back and forth nudging algorithm

The back and forth nudging (BFN) algorithm is an iterative algorithm: loops of forward and backward integrations, both with nudging, are performed over a time period T during which observations are available (Fig. 1). The evolution of the system is governed by the two equations:

$$\begin{cases} \text{(F)} & \frac{\partial \mathbf{X}_k^f}{\partial t} + \mathbf{M}^f \frac{\partial \mathbf{X}_k^f}{\partial x} = \mathbf{S} + \mathbf{K} [\mathbf{y}^o - \mathcal{H}(\mathbf{X}_k^f)] & \text{for } 0 \leq t \leq T, \delta t > 0 \\ \text{(B)} & \frac{\partial \mathbf{X}_k^b}{\partial t} + \mathbf{M}^b \frac{\partial \mathbf{X}_k^b}{\partial x} = \mathbf{S} - \tilde{\mathbf{K}} [\mathbf{y}^o - \mathcal{H}(\mathbf{X}_k^b)] & \text{for } T \geq t \geq 0, \delta t < 0 \end{cases} \quad (2)$$

where \mathbf{K} and $\tilde{\mathbf{K}}$ are gain matrices, the superscript f and b refer to forward and backward variables, and the subscript k refers to the index of the BFN iteration.

With the changes of variables $\tilde{t} = T - t$, $\tilde{\mathbf{X}} = \begin{pmatrix} h \\ \tilde{u} \end{pmatrix}$, and $\tilde{\mathbf{M}} = \begin{pmatrix} \tilde{u} & h \\ g & \tilde{u} \end{pmatrix}$, where $\tilde{u} = -u$, the backward equation is exactly the same as the forward equation (1) where u is replaced with $\tilde{u} = -u$.

During each integration (either forward or backward), there might be a conflict between the prescribed external boundary conditions and the internal state of the system, modified by the nudging. To avoid this conflict, the length of the data assimilation time window, T , should be chosen equal to the time needed for the nudging information to propagate into the whole domain.

Applying the method of characteristics to the backward equation, we have shown that the boundary conditions must be reversed between forward and backward integration (Fig. 1). The boundary condition that is prescribed upstream in the forward integration (here the velocity on the left of the domain, u_L) must be prescribed downstream in the backward integration (here \tilde{u}_R on the right). Similarly, the boundary condition prescribed downstream in the forward integration (here the height on the right, h_R) must be prescribed upstream in the backward integration (here h_L on the left). The values of the boundary conditions \tilde{u}_R and h_L for the backward integration are given by the system state on the right and left of the domain, respectively, at the end of the forward integration: $\tilde{u}_R^k = -u^k(x = L, t = T)$ and $h_L^k = h^k(x = 0, t = T)$.

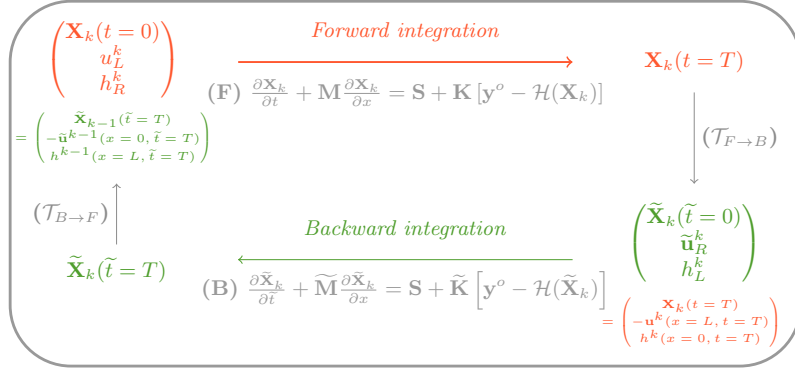


Figure 1: Diagram of the k^{th} cycle of the BFN algorithm. The transformations $(\mathcal{T}_{F \rightarrow B})$ and $(\mathcal{T}_{B \rightarrow F})$ shift from forward system to backward system and vice versa. They include a change of variable $\tilde{t} = T - t$ and an inversion of boundary conditions. In the case of shallow water equations, the backward integration is equivalent to perform an integration with positive time steps but with $\tilde{\mathbf{u}} = -\mathbf{u}$.

Iterative ensemble Kalman smoother

The iterative ensemble Kalman smoother (IEnKS) (Bocquet and Sakov, 2014) considers assimilation time windows of size $L\Delta t$ with $L \geq 1$. The iterative ensemble Kalman filter (Sakov et al., 2012) is the particular case where $L = 1$. If $L > 1$, the analysis is performed at a time t_L and focuses on the system state at the beginning of the assimilation time window, i.e. at $t_0: \mathbf{x}_0$ (Fig. 2).

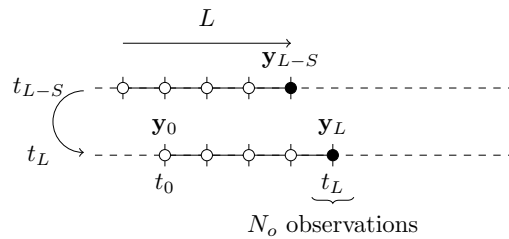


Figure 2: Two cycles of IEnKS analysis. An analysis is performed every S time steps, with $1 \leq S \leq L + 1$. The number of new observation vectors included in the analysis (and thus the cost function) at each analysis cycle is referred to as N_o . Here the data assimilation time window includes $L = 4$ time steps and we consider the simple case where $S = N_o = 1$ for which only the last observation of each assimilation window is newly assimilated. It is a single data assimilation scheme: each observation is assimilated only once.

The evolution between t_0 and t_L depends on the initial state \mathbf{x}_0 and all the boundary conditions between these two times, stored in the vector $\mathbf{z}_{0,L:1}$. It is governed by the model $\mathcal{M}_{L:0}$:

$$\mathbf{x}_L = \mathcal{M}_{L:0}(\mathbf{x}_0, \mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_L) = \mathcal{M}_{L:0}(\mathbf{z}_{0,L:1}) \quad (3)$$

$$\mathbf{z}_{0,L:1} = \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{b}_{L:1} \end{pmatrix} = (\mathbf{x}_0 \quad \mathbf{b}_1 \quad \mathbf{b}_2 \quad \dots \quad \mathbf{b}_L)^\top \quad (4)$$

The model is assumed perfect and all past observations up to t_L are given by the vector \mathbf{y}_L . The observation errors — defined as $\mathbf{y}_L - \mathcal{H}_L(\mathbf{x}_L)$, where \mathcal{H}_L is the observation operator — are assumed to be Gaussian with zero mean and covariance matrix \mathbf{R}_L . We consider an ensemble of N^x initial state vectors and an ensemble of N^b boundary condition vectors: $\mathbf{E}_0^x = \{\mathbf{x}_{0,[i]}\}_{i=1\dots N^x}$ and $\mathbf{E}_{L+1:1}^b = \{\mathbf{b}_{L+1:1,[i]}\}_{i=1\dots N^b}$, centred on the background estimate $\mathbf{z}_{0,L+1:1}^{(0)}$. The smoothing pdf of $\mathbf{z}_{0,L:1}$, $p(\mathbf{z}_{0,L:1}|\mathbf{y}_{L-1})$, is assumed to follow a Gaussian distribution of mean $\mathbf{z}_{0,L:1}^{(0)}$ and covariance matrix $\tilde{\mathbf{P}}_{0,L:1}$, given by the previous cycle of the IEnKS.

We seek a combination of the ensemble members, defined by the weight $\mathbf{w}_{0,L+1:1}$, that gives the best estimate of $\mathbf{z}_{0,L+1:1}$. Using a Bayesian approach, we have derived a cost function $\tilde{\mathcal{J}}$ in the ensemble space:

$$\tilde{\mathcal{J}}(\mathbf{w}_{0,L+1:1}) = \frac{1}{2}\|\mathbf{w}_{0,L+1:1}\|^2 + \frac{1}{2}\|\mathbf{y}_L - \mathcal{H}_L \circ \mathcal{M}_{L:0}(\mathbf{z}_{0,L+1:1}^{(0)} + \mathbf{A}_{0,L+1:1}\mathbf{w}_{0,L+1:1})\|_{\mathbf{R}_L^{-1}}^2 \quad (5)$$

where $\mathbf{A}_{0,L+1:1}$ is the anomaly matrix. The minimum of this cost function is reached for the most likely $\mathbf{w}_{0,L+1:1}$, referred to as $\mathbf{w}_{0,L+1:1}^*$. Due to pre-conditioning by \mathbf{A} , the calculation of the gradient of the cost function does not require the adjoint nor the tangent linear of $\mathcal{H}_L \circ \mathcal{M}_{L:0}$. Consequently the cost function is pretty easily minimised, following, for instance, a Gauss-Newton algorithm. The best estimate of $\mathbf{z}_{0,L+1:1}$ is thus given by $\mathbf{z}_{0,L+1:1}^* = \mathbf{z}_{0,L+1:1}^{(0)} + \mathbf{A}_{0,L+1:1}\mathbf{w}_{0,L+1:1}^*$.

RESULTS

The BFN algorithm and the IEnKS have been tested on a shallow water two-layer system for two one-dimensional twin experiments ("observations" are extracted from the reference simulation). Both experiments correspond to a channel with a realistic topography profile but without dissipation. The true state is obtained with an upstream boundary condition equal to $u_L = u^t = 5.5\text{m/s}$ and the prior estimate (or background) corresponds to the boundary value $u_L = u^b = 4.4\text{m/s}$. The value of the downstream boundary condition is unchanged between the reference simulation and the prior estimate ($h_R = 617\text{m}$). Five observations, equally distributed, are extracted from the reference simulation at every time step. In the first experiment, the observations are kept exact, whereas in the second experiment they are perturbed with a white noise.

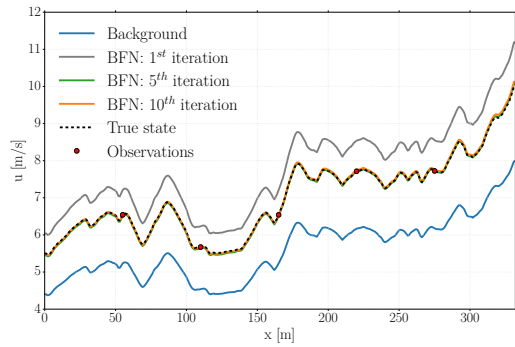
Figure 3a (respectively Fig. 3c) shows the state of the system after 1, 5, and 10 BFN iterations in the experiment with perfect (respectively noisy) observations. The nudging matrix is the same in forward and backward integration: $\mathbf{K} = k\mathbf{H}^\top$ where $k\Delta t = 0.1$. Less than 10 BFN iterations are required to reach the convergence, i.e. the model has to be integrated less than 20 times. Figures 3b and 3d show the IEnKS analysis state, obtained with $\tilde{\mathbf{P}} = \mathbf{I}$, $\mathbf{R} = 0.1\mathbf{I}$, and an ensemble of 3 members. The minimisation of the cost function only needs 2 or 3 iterations of the Gauss-Newton algorithm, i.e. this method requires less than 10 model integrations. Both the BFN algorithm and the IEnKS help correcting the boundary conditions, though the IEnKS is slightly more efficient in this case as it requires less integrations of the model.

CONCLUSION

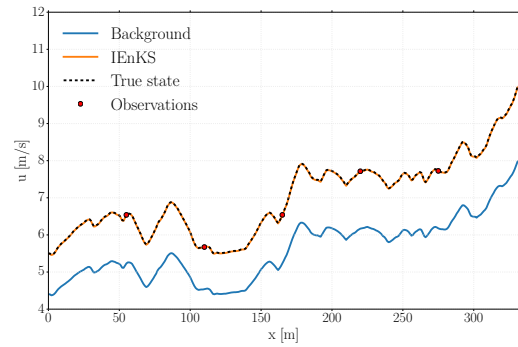
Variants of the BFN algorithm and the IEnKS have been derived to take boundary conditions into account in addition to the initial conditions. The first results show the efficiency of these methods to assimilate a few observations, perfect or noisy, on cases with a complex one-dimensional geometry. Further investigation on cases with two horizontal dimensions and vertical profiles of velocity will be performed, as well as sensibility analysis to the different parameters involved in each method. Eventually, we will apply these methods to more realistic cases with *Code_Saturne*.

ACKNOWLEDGEMENT

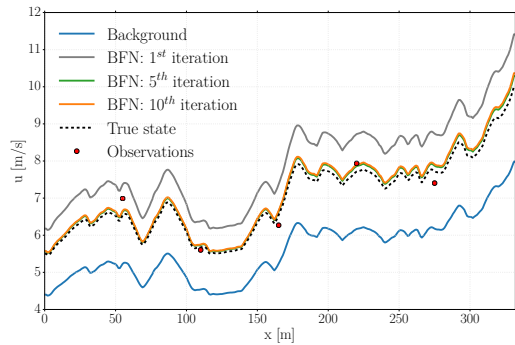
The CERA is a member of the Pierre-Simon Laplace Institute (IPSL).



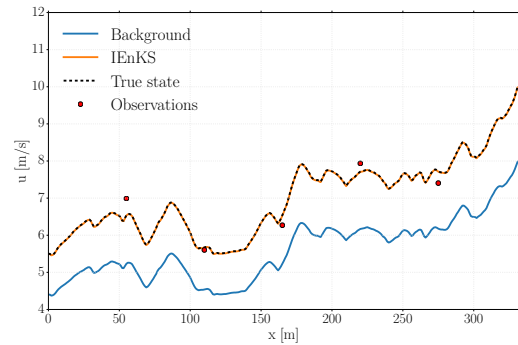
(a) Results of the BFN algorithm with perfect observations



(b) Results of the IEnKS with perfect observations



(c) Results of the BFN algorithm with noisy observations



(d) Results of the IEnKS with noisy observations

Figure 3: Profiles of the velocity obtained after some BFN iterations (a and c) and at the end of the IEnKS analysis (b and d) for experiments with perfect (a and c) and noisy observations (b and d). The simulations are performed with a shallow water model without dissipation.

REFERENCES

- Archambeau, F., N. Méchitoua, and M. Sakiz, 2004: Code Saturne: A Finite Volume Code for the computation of turbulent incompressible flows - Industrial Applications. *Int. J. on Finite Vol.*, **1**.
- Asch, M., M. Bocquet, and M. Nodet, 2016: *Data assimilation: methods, algorithms, and applications*. Society for Industrial and Applied Mathematics.
- Auroux, D., 2009: The back and forth nudging algorithm applied to a shallow water model, comparison and hybridization with the 4D-VAR. *Int. J. for Numer. Methods Fluids*, **61**, 911–929.
- Auroux, D., P. Bansart, and J. Blum, 2013: An evolution of the back and forth nudging for geophysical data assimilation: application to burgers equation and comparisons. *Inverse Probl. Sci. Eng.*, **21**, 399–419.
- Auroux, D., and J. Blum, 2005: Back and forth nudging algorithm for data assimilation problems. *Comptes Rendus Math.*, **340**, 873–878.
- Auroux, D., and J. Blum, 2008: A nudging-based data assimilation method: the Back and Forth Nudging (BFN) algorithm. *Nonlin. Process. Geophys.*, **15**, 305–319.
- Auroux, D., J. Blum, and M. Nodet, 2011: Diffusive Back and Forth Nudging algorithm for data assimilation. *Comptes Rendus Math.*, **349**, 849–854.
- Auroux, D., and M. Nodet, 2012: The Back and Forth Nudging algorithm for data assimilation problems : theoretical results on transport equations. *ESAIM: Control. Optimisation Calc. Var.*, **18**, 318–342.
- Bocquet, M., and P. Sakov, 2014: An iterative ensemble Kalman smoother. *Quart. J. Royal Meteor. Soc.*, **140**, 1521–1535.
- Kalnay, E., 2003: *Atmospheric modeling, data assimilation, and predictability*. Cambridge University Press.
- Sakov, P., D. S. Oliver, and L. Bertino, 2012: An Iterative EnKF for Strongly Nonlinear Systems. *Mon. Wea. Rev.*, **140**, 1988–2004.