Improving CFD atmospheric simulations at local scale for wind resource assessment using the iterative ensemble Kalman smoother

Cécile L Defforge, B. Carissimo, M. Bocquet, R. Bresson, P. Armand

Improving CFD atmospheric simulations at local scale for wind resource assessment using the iterative ensemble Kalman smoother

Cécile L. Defforge[a,*], B. Carissimo[a], M. Bocquet[a], R. Bresson[a], P. Armand[b]

[a] *CEREA, Joint laboratory École des Ponts ParisTech / EDF R&D, Université Paris-Est, Marne-la-Vallée, France*
[b] *CEA, DAM, DIF, F-91297 Arpajon, France*

Abstract

Accurate wind fields simulated by CFD models are necessary for many environmental and safety micrometeorological applications, such as wind resource assessment. Atmospheric simulations at local scale are largely determined by boundary conditions (BCs), which are generally provided by outputs of mesoscale models (e.g., WRF). In order to improve the accuracy of the BCs, especially in the lowest levels, data assimilation methods might be used to take available observations into account. Data assimilation methods have generally been developed for larger scale meteorology and deal with initial conditions. Among the existing methods, the iterative ensemble Kalman smoother (IEnKS) has been chosen and adapted to micrometeorology by taking BCs into account. In the present study, we assess the ability of the adapted IEnKS to improve wind simulations over a very complex topography in a context of wind resource assessment, by assimilating a few *in situ* observations. The IEnKS is tested with the CFD model *Code_Saturne* in 2D and 3D using both twin experiments and real observations. We propose a method to determine the first estimate of the BCs and to construct the associated background error covariance matrix, from the statistical analysis of three years of WRF simulations. The IEnKS is proved to greatly reduce the error and the uncertainty of the BCs and thus of the simulated wind field over the small-scale domain. As a consequence, the wind resource estimate is also much more accurate.

Highlights

- This article provides a framework to perform ensemble variational data assimilation of *in situ* observations to improve local scale simulations with a CFD model.
- The iterative ensemble Kalman smoother is adapted to correct boundary conditions and is tested with twin experiments and real data experiments in 2D and 3D with a CFD model over very complex topography.
- The adapted IEnKS is proved to enhance the accuracy of boundary conditions and local scale simulations in operationally affordable conditions.

Keywords **-** Data assimilation, micrometeorology, computational fluid dynamics, wind resource assessment, wind potential, iterative ensemble Kalman smoother, local scale simulation, boundary conditions.

1. Introduction

Local scale atmospheric simulations are necessary to address some micro-meteorological issues. For instance, estimating the wind potential of a given site requires accurate simulations of the wind field over this site. To perform such atmospheric simulations at local scale, CFD models (e.g., *Code_Saturne*, Archambeau

*Corresponding author: cecile.defforge@enpc.fr

et al., 2004) are increasingly used (Landberg et al., 2003; Franke et al., 2004; Blocken, 2014), especially over complex terrain (e.g., Cattin et al., 2006; Palma et al., 2008; Blocken et al., 2015) and in built environment (e.g., Kalmikov et al., 2010; Simões and Estanqueiro, 2016; Wang et al., 2017).

The local wind fields may be difficult to simulate with CFD models because they are very sensitive to geometrical features, such as topography, and boundary conditions (BCs) (Yang et al., 2009). In fact, the time needed for any information to propagate into a small scale domain is much smaller than the characteristic integration time scale. Consequently the influence of initial conditions quickly vanishes with integration time, and the BCs eventually control the solution. The BCs thus account for a major source of uncertainty in local scale atmospheric simulations and it is a challenge to accurately estimate them.

To first estimate the initial and BCs for CFD simulations at small scale, the results of mesoscale simulations are generally used (e.g. Yamada, 2004; Probst et al., 2010). However, due to the large difference in spatial and temporal resolution between mesoscale and CFD domains, such micro-scale imbrications raise many issues. In particular, the BCs provided by mesoscale simulations are generally incorrect and do not always contain all the information, such as turbulence estimations. Consequently, mesoscale simulations provide useful information that is not sufficient and should be combined to another source of information, if available, to estimate the BCs as accurately as possible.

The prospective sites for implantation of wind farms are often equipped with meteorological instruments, such as anemometers or lidars, which supply wind measurements inside the domain of interest (Landberg et al., 2003). These observations provide supplementary information that could be used with data assimilation (DA) methods to improve the knowledge of BCs, and thus increase the precision of atmospheric simulations over small domains.

Up to now, DA methods have generally been developed in the framework of large scale meteorology and focus on initial conditions (e.g., Kalnay, 2003; Asch et al., 2016). Since large scale simulations are largely influenced by initial conditions, DA methods are mostly used to improve the accuracy of the initial conditions. Recently, some aeronautic studies have applied DA methods with CFD simulations to estimate some inflow parameters (Misaka et al., 2008; Kato et al., 2015). In other CFD studies, simple nudging has been used for downscaling from mesoscale to microscale atmospheric models (Zajaczkowski et al., 2011; Duraisamy et al., 2014), though differences in resolution between the models might be problematic. More recently, Mons et al. (2017) and Sousa et al. (2018) have used DA to estimate the incoming wind direction and velocity used as inputs for CFD simulations. Here we propose to describe the BC with several vertical profiles of velocity, which correspond to hundreds of control variables. We thus use DA to enhance the accuracy of small scale atmospheric simulations through the correction of a large control vector of BCs. In order to do so, we first need to chose the DA method most adapted to the constraints inherent to CFD models.

Existing DA methods are generally divided into two classes: variational methods (Le Dimet and Talagrand, 1986) and statistical (or filtering) methods. Filtering methods (e.g., ensemble Kalman filter, Evensen, 1994) are based on the minimisation of the variance of analysis error, which depends on the background error and the observation error statistics. The derivation of these methods assumes the linearity of the model and the observation operator, which is not necessarily verified. However filtering methods are well adapted to parallel computation. Variational methods (e.g., 3D-Var, 4D-Var, Courtier and Talagrand, 1990) are based on the minimisation of a cost function that measures misfit to the available data. It is generally the departure of the control variable from the background and of its image from the observations. Variational methods have the advantage of better handling nonlinear analyses but they require the minimisation of a scalar function defined over a large-dimensional space. This is usually achieved by using the adjoint of the forward operator or using finite difference approximation. However, the adjoint of CFD models is rarely available and finite differences require a very large number of model integrations. Consequently, 3D-Var and 4D-Var are not well adapted for an operational use of DA with a CFD model. Moreover, these methods do not easily yield an error estimate of the analysis.

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 combine the advantages of both ensemble-based and variational methods: they work for nonlinear systems, do not require the use of the adjoint nor the tangent linear of the forward operator, and they can easily be parallelised. The IEnKS

and IEnKF consist in defining a cost function in the ensemble space, which is iteratively minimised using, for instance, Gauss-Newton method. With a linear model, the filtering solution obtained at the end of the DA window with the IEnKF or the IEnKS is equivalent to the one obtained with the ensemble Kalman Filter.

In the context of CFD simulations for operational purposes, the adjoint model is rarely available and the DA method must be adapted to parallel computation and to nonlinear systems. Since the IEnKS fulfills these conditions and is well suited for parameter estimation, it has been chosen here to perform DA with *Code_Saturne*. The IEnKS has already been tested, among many other models, with a 1D shallow layer model and proved to help correct BCs in a simple case (Defforge et al., 2018). We analyse here its performance on more realistic cases.

In the present work, the usual derivation of the cost function used in the variational ensemble methods is adapted to take BCs into account. In what follows, we describe two DA methods in their versions adapted for local scale simulations: 3D-Var – which is used here as a reference method for comparison purposes – and the iterative ensemble Kalman smoother. We also present the method used to estimate the background error covariance matrix related to the control vector of BCs.

The goal of this study is to assess the ability of the adapted IEnKS to improve 3D wind simulations in a context of wind resource assessment, by assimilating a few *in situ* observations. We consider a real domain with very complex topography prospected for the installation of a wind farm and on which there has been a field campaign of wind observations. We first present the results obtained over a 2D section of this domain in the vertical plane, with twin experiments. Besides reducing the computation cost – and thus allowing us to perform more sensitivity test – this 2D case has the advantage to require the definition of BCs on one side only and to allow comparison with 3D-Var. In a second time, we consider the whole 3D domain with four lateral boundaries. To gradually increase the complexity of the DA problem, the IEnKS is tested over this 3D domain with twin experiments and then with real observations in a cross validation process.

2. Methods

2.1. 3D-Var: a reference method

Among variational DA methods, 3D-Var and 4D-Var are generally considered as references. It is thus interesting to compare other methods with them. In the present study, we consider a problem with dynamical evolution towards a steady state. Consequently, 3D-Var and 4D-Var algorithms are equivalent in this case, and we chose to keep the name 3D-Var since we consider steady states. 3D-Var is based on the minimisation of a cost function defined in the control space (e.g., Daley, 1991). The forward operator $\mathcal{F}$ considered here transports the BCs into the observation space. It thus includes the dynamical model between BCs and steady state ($\mathcal{M}$), and the projection of the physical steady state thus obtained into observation space ($\mathcal{H}$):

$$\mathcal{F} = \mathcal{H} \circ \mathcal{M}, \tag{1}$$

where the operator '$\circ$' refers to function composition.

Here the control vector corresponds to the BCs and the goal is to find the optimal value of this control vector, according to all available information. 3D-Var seeks this optimum through the minimisation of a cost function that measures the departure from the background and the departure of the image of the control vector, by the forward operator, from the observations. Each distance is weighted by the trust placed in either the background or the observations, which corresponds to the inverse of the error covariance matrices. The 3D-Var cost function adapted to the BCs can be derived from Baye's rule, as given below.

2.1.1. Bayesian derivation of the cost function

The control vector $\mathbf{z}$ is formed of l values that define the constant BCs. The control vector is assumed to follow a Gaussian distribution with mean $\mathbf{z}^{\mathrm{b}}$, which is the background vector of BCs, and the background error covariance matrix is $\mathbf{B}$ of size $l \times l$. The probability density function (pdf) of $\mathbf{z}$ is of the form

$$p(\mathbf{z}) = n\left(\mathbf{z}|\mathbf{z}^{\mathrm{b}}, \mathbf{B}\right), \tag{2}$$

where $n(\mathbf{a}|\overline{\mathbf{a}}, \mathbf{A})$ represents the normal distribution of mean $\overline{\mathbf{a}}$ and with covariance matrix $\mathbf{A}$. The background error

$$\epsilon^{\mathrm{b}} = \mathbf{z}^{\mathrm{b}} - \mathbf{z}^{\mathrm{t}}, \tag{3}$$

where the superscript 't' refers to the unknown truth, is thus assumed unbiased and only the random component of the error is considered here.

The steady state obtained when these BCs are prescribed, is referred to as $\mathbf{x}_s$. It is a vector of $\mathbb{R}^m$ and depends only on $\mathbf{z}$. The size of the state vector is substantially larger than the vector defining the BCs: $l \ll m$. The model which relates the steady state to the BCs is assumed perfect:

$$p(\mathbf{x}_s|\mathbf{z}) = \delta\left[\mathbf{x}_s - \mathcal{M}(\mathbf{z})\right], \tag{4}$$

where δ is the Dirac distribution defined here in $\mathbb{R}^m$.

Observations ($\mathbf{y} \in \mathbb{R}^p$) are available for this steady state. Given the state of the system, $\mathbf{x}_s$, the observations $\mathbf{y}$ are assumed to follow a Gaussian distribution:

$$p(\mathbf{y}|\mathbf{x}_s) = n\left(\mathbf{y} - \mathcal{H}(\mathbf{x}_s)|\mathbf{0}, \mathbf{R}\right), \tag{5}$$

where $\mathbf{R}$ is the observation error covariance matrix of size $p \times p$. Both $\mathcal{M}$ and $\mathcal{H}$ may be nonlinear.

Using Bayes' rule and equation (4), the filtering pdf of $\mathbf{z}$ reads

$$\begin{aligned} p(\mathbf{z}|\mathbf{y}) &\propto p(\mathbf{y}|\mathbf{z})p(\mathbf{z}) \\ &\propto p\left(\mathbf{y}|\mathbf{x}_s = \mathcal{M}(\mathbf{z})\right)p(\mathbf{z}). \end{aligned} \tag{6}$$

A cost function can be derived from this pdf, satisfying

$$p(\mathbf{z}|\mathbf{y}) \propto \exp\left(-\mathcal{J}(\mathbf{z})\right). \tag{7}$$

The assumption of Gaussian distributions (equations 2 and 5) combined with equation (6) gives the expression of the cost function, expressed in the control space:

$$\mathcal{J}(\mathbf{z}) = \frac{1}{2}\|\mathbf{z} - \mathbf{z}^{\mathrm{b}}\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\|\mathbf{y} - \mathcal{H} \circ \mathcal{M}(\mathbf{z})\|^2_{\mathbf{R}^{-1}} \tag{8}$$

$$= \frac{1}{2}\|\mathbf{z} - \mathbf{z}^{\mathrm{b}}\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\|\mathbf{y} - \mathcal{F}(\mathbf{z})\|^2_{\mathbf{R}^{-1}}, \tag{9}$$

where we use the notation $\|\mathbf{x}\|^2_{\mathbf{A}} = \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}$.

2.1.2. Minimisation of the 3D-Var cost function

The minimisation of the cost function previously defined requires the estimation of its gradient, which can be performed using the tangent linear and the adjoint of the forward operator. If they are not available, the gradient of the cost function might also be estimated with finite differences. Even though this second method is quite inefficient, because finite difference approximation requires as many model integrations as control variables, it is easier to implement and does not depend on the dynamical model. Consequently, we use the ADAO module of the SALOME open-source platform (http://www.salome-platform.org) which offers the possibility to run 3D-Var with a finite difference approximation of the gradient. The L-BFGS-B nonlinear, constrained optimisation algorithm (Byrd et al., 1995) is then used to minimise the cost function. We have performed some tests to verify that the tangent linear is correctly estimated with finite differences when the differential increment is set to 10^{-2}. The results of these verification tests are shown in Appendix B.

The minimum of this cost function is reached for a given set of BCs: the analysis vector $\mathbf{z}^{\mathrm{a}}$.

2.2. The iterative ensemble Kalman smoother

The iterative ensemble Kalman smoother (Bocquet and Sakov, 2014) is a variational ensemble method: a cost function is minimised in order to find the best combination of ensemble members. Since we consider stationary BCs and we assume that the model is perfect, there is no difference between the filtering and smoothing formulations here. Anticipating the future applications with unsteady BCs, we will consider the smoothing method (IEnKS) in what follows, keeping in mind that it is here strictly equivalent to the filtering method (IEnKF). The IEnKS cost function is the same as the one obtained in the derivation of 3D-Var (equation 8), though considered in the ensemble space. We give below the derivation of the cost function in the ensemble space and its minimisation avoiding the use of the adjoint model.

2.2.1. Derivation of the cost function in the ensemble space

Similarly to the derivation in Sakov et al. (2012) and Bocquet and Sakov (2014), we consider a background ensemble of N members: $\mathbf{E}^{\mathrm{b}} = \left\{\mathbf{z}_{[i]}\right\}_{i=1..N}$, where index $[i]$ refers to the member index in the ensemble. Each member corresponds to a vector of BCs.

The background ensemble is centred on the background vector of BCs:

$$\mathbf{z}^{\mathrm{b}} = \overline{\mathbf{z}} = \frac{1}{N}\mathbf{E}^{\mathrm{b}}\mathbf{1}, \tag{10}$$

where $\mathbf{1}$ is a vector of size N with all components equal to one. The (normalised) anomaly matrix for $\mathbf{z}$ is defined in $\mathbb{R}^{l\times N}$ as the departure of each member from the ensemble mean ($\mathbf{z}^{\mathrm{b}}$):

$$\mathbf{A} = \frac{1}{\sqrt{N-1}}\left[\mathbf{z}_{[1]} - \mathbf{z}^{\mathrm{b}}, \mathbf{z}_{[2]} - \mathbf{z}^{\mathrm{b}}, ..., \mathbf{z}_{[N]} - \mathbf{z}^{\mathrm{b}}\right]. \tag{11}$$

The background error covariance matrix is then estimated from the ensemble using the previously defined anomaly matrix:

$$\mathbf{B} = \mathbf{A}\mathbf{A}^{\mathrm{T}}. \tag{12}$$

In fact, the spread of the background ensemble around its mean represents the uncertainty on the background.

The ensemble space is smaller than the state space such that it is generally interesting and more efficient to seek a solution to the DA problem in the ensemble space rather than in the state space. Consequently, the IEnKS seeks the linear combination of the ensemble members that best describes the analysis BCs. In practice, for a given value of the weight vector, the corresponding vector of BCs is

$$\mathbf{z} = \mathbf{z}^{\mathrm{b}} + \mathbf{A}\mathbf{w}. \tag{13}$$

Replacing (12) and (13) in (8) and using the notation $\|\mathbf{x}\|^2 = \mathbf{x}^{\mathrm{T}}\mathbf{x}$, the cost function in the ensemble space reads

$$\widetilde{\mathcal{J}}(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \mathcal{H}\circ\mathcal{M}(\mathbf{z}^{\mathrm{b}} + \mathbf{A}\mathbf{w})\|^2_{\mathbf{R}^{-1}} + \frac{1}{2}\|\mathbf{w}\|^2. \tag{14}$$

The cost function derived here is the same as that obtained in previous DA schemes (e.g., Bocquet and Sakov, 2014), albeit replacing the initial conditions by the BCs.

2.2.2. Minimisation of the cost function in the ensemble space

In the following, the cost function is minimized in the ensemble space following the Gauss-Newton algorithm. In theory, it would be equivalent to use the L-BFGS-B scheme, since the non-linearities are not too strong here. A great advantage of working in the ensemble space is that the calculation of the gradient of the cost function does not require the adjoint nor the tangent linear of the forward operator $\mathcal{H}\circ\mathcal{M}$. It is replaced by the use of an ensemble, thanks to the pre-conditioning by $\mathbf{A}$ as in Liu et al. (2008) and Gu and Oliver (2007). In fact, one only requires the tangent linear of the operator transporting from the ensemble space to the observation space:

$$\mathbf{Y}^{(j)} = \left[\mathcal{H}\circ\mathcal{M}\right]'_{|\mathbf{z}^{(j)}}\mathbf{A}, \tag{15}$$

where $\mathbf{z}^{(j)} = \mathbf{z}^{\mathrm{b}} + \mathbf{A}\mathbf{w}^{(j)}$ and the index (j) refers to the iteration index in the Gauss-Newton algorithm. The matrix $\mathbf{Y}^{(j)}$ is of size $p \times N$ and since p and N are generally small (especially in the cases we are interested in), the computation of $\mathbf{Y}^{(j)}$ and $\left(\mathbf{Y}^{(j)}\right)^{\mathrm{T}}$ is not too expensive.

Using previously defined notation, the Gauss-Newton algorithm reads

$$\mathbf{w}^{(j+1)} = \mathbf{w}^{(j)} - \mathbb{H}_{(j)}^{-1} \nabla \widetilde{\mathcal{J}}_{(j)} \left(\mathbf{w}^{(j)}\right), \tag{16}$$

where $\nabla \widetilde{\mathcal{J}}$ is the gradient and $\mathbb{H}$ is the Hessian of the cost function.

At each step of this iterative algorithm, the ensemble anomalies must be rescaled such that the propagated ensemble is representative of the current estimate of the state error. There are two methods to do so (Sakov et al., 2012; Bocquet and Sakov, 2012). In the *bundle* variant, the ensemble is shrunk by a small factor ϵ then propagated and eventually inflated by $1/\epsilon$. In the *transform* variant, the ensemble is transformed before its propagation using the ensemble transform matrix

$$\mathbf{T}_{(j)} = \left(\mathbf{I}_N + \left(\mathbf{Y}^{(j-1)}\right)^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{Y}^{(j-1)}\right)^{-\frac{1}{2}} \tag{17}$$

obtained at the previous iteration, where $\mathbf{I}_N$ is the identity matrix of size $N \times N$. The propagated ensemble is then rescaled using the transform inverse to (17).

Here we use the *transform* method, which gives

$$\mathbf{Y}^{(j)} \approx \mathcal{H} \circ \mathcal{M} \left(\mathbf{z}^{(j)} \mathbf{1}^{\mathrm{T}} + \sqrt{N-1} \mathbf{A} \mathbf{T}_{(j)}\right) \left(\mathbf{I}_N - \frac{\mathbf{1}\mathbf{1}^{\mathrm{T}}}{N}\right) \frac{\mathbf{T}_{(j)}^{-1}}{\sqrt{N-1}}, \tag{18}$$

with $\mathbf{T}_{(j)}$ is the ensemble transform matrix previously defined (equation 17).

The gradient and the approximate Hessian of the cost function are given by

$$\nabla \widetilde{\mathcal{J}}_{(j)} = \mathbf{w}^{(j)} - \left(\mathbf{Y}^{(j)}\right)^{\mathrm{T}} \mathbf{R}^{-1} \left[\mathbf{y} - \mathcal{H} \circ \mathcal{M}(\mathbf{z}^{(j)})\right] \text{ and} \tag{19}$$

$$\mathbb{H}_{(j)} = \mathbf{I}_N + \left(\mathbf{Y}^{(j)}\right)^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{Y}^{(j)}. \tag{20}$$

In the computation of the gradient $\nabla \widetilde{\mathcal{J}}$, we assume that averaging the ensemble of simulated observations is equivalent to applying the forward operator to the mean BCs:

$$\mathcal{H} \circ \mathcal{M} \left(\mathbf{z}^{(j)} \mathbf{1}^{\mathrm{T}} + \sqrt{N-1} \mathbf{A} \mathbf{T}_{(j)}\right) \frac{\mathbf{1}\mathbf{1}^{\mathrm{T}}}{N} \approx \mathcal{H} \circ \mathcal{M}(\mathbf{z}^{(j)}). \tag{21}$$

The convergence criterion for the minimisation algorithm is defined here as the iteration (j) when the change in the cost function relative to the initial value of the cost function $\frac{|\mathcal{J}^{(j)} - \mathcal{J}^{(j-1)}|}{\mathcal{J}^{(0)}}$ is smaller than a predefined precision level e_J. More details about this choice are given in Appendix A. The solution of the cost function minimisation is referred to as $\mathbf{w}^{\mathrm{a}}$. The 'a' superscript is used to refer to all the variables obtained at the end of the minimisation. The optimal vector of BCs, according to the previously defined DA problem, is then given by

$$\mathbf{z}^{\mathrm{a}} = \mathbf{z}^{\mathrm{b}} + \mathbf{A}\mathbf{w}^{\mathrm{a}}. \tag{22}$$

A posterior ensemble is obtained at the end of the analysis and its spread is informative about the uncertainty of the analysis:

$$\mathbf{E}^{\mathrm{a}} = \mathbf{z}^{\mathrm{a}} \mathbf{1}^{\mathrm{T}} + \sqrt{N-1} \mathbf{A} \left(\mathbb{H}^{\mathrm{a}}\right)^{-\frac{1}{2}} \mathbf{U}, \tag{23}$$

where $\mathbf{U}$ is an orthogonal matrix of size $N \times N$ satisfying $\mathbf{U}\mathbf{1} = \mathbf{1}$, used to keep the ensemble centred on the analysis (here we set $\mathbf{U} = \mathbf{I}_N$).

In an unsteady context, the results obtained at the end of the minimisation ($\mathbf{z}^{\mathrm{a}}$, $\mathbf{E}^{\mathrm{a}}$, etc.) could be used as background information for a subsequent analysis cycle. In the present study, we only consider one analysis cycle, such that we stop the algorithm at this point. The pseudo-code of Algorithm 1 details one analysis cycle of the IEnKS using the *transform* variant and the Gauss-Newton minimisation algorithm.

Algorithm 1 A cycle of transform/Gauss-Newton IEnKS

Require: Transition model from BCs ($\mathbf{z}$) to steady state ($\mathbf{x}_s$): $\mathcal{M}$, observation operator: $\mathcal{H}$, observation vector: $\mathbf{y}$, background ensemble: $\mathbf{E}^{\mathrm{b}}$, and algorithm parameters: e_J, j_{max}. $\mathbf{U}$ is an orthogonal matrix of size $N \times N$ satisfying $\mathbf{U1} = \mathbf{1}$.

1. $j = 0$, $\mathbf{w} = \mathbf{0}$
   $\mathbf{T} = \mathbf{I}$
2. $\mathbf{z}^{\mathrm{b}} = \overline{\mathbf{z}} = \mathbf{E}^{\mathrm{b}}\mathbf{1}/N$
3. $\mathbf{A} = \frac{1}{\sqrt{N-1}}\left(\mathbf{E}^{\mathrm{b}} - \mathbf{z}^{\mathrm{b}}\mathbf{1}^{\mathrm{T}}\right)$
4. **repeat**
   5. $\mathbf{z} = \mathbf{z}^{\mathrm{b}} + \mathbf{A}\mathbf{w}$
   6. $\mathbf{E}_0 = \mathbf{z}\mathbf{1}^{\mathrm{T}} + \sqrt{N-1}\mathbf{A}\mathbf{T}$
   7. $\mathbf{E}_s = \mathcal{M}\left(\mathbf{E}_0\right)$
   8. $\bar{\mathbf{y}}_s = \mathcal{H}\left(\mathbf{E}_0\right)\mathbf{1}/N$
   9. $\mathbf{Y} = \left(\mathcal{H}\left(\mathbf{E}_s\right) - \bar{\mathbf{y}}_s\mathbf{1}^{\mathrm{T}}\right)\frac{\mathbf{T}^{-1}}{\sqrt{N-1}}$
   10. $\nabla\widetilde{\mathcal{J}} = \mathbf{w} - \mathbf{Y}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y} - \bar{\mathbf{y}}_s)$
   11. $\mathbb{H} = \mathbf{I}_N + \mathbf{Y}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{Y}$
   12. Solve $\mathbb{H}\Delta\mathbf{w} = \nabla\widetilde{\mathcal{J}}$
   13. $\mathbf{w} := \mathbf{w} - \Delta\mathbf{w}$
   14. $j := j + 1$
   15. $\mathbf{T} := \mathbb{H}^{-1/2}$
   16. $\mathcal{J} = \mathbf{w}^{\mathrm{T}}\mathbf{w} + (\mathbf{y} - \bar{\mathbf{y}}_s)^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y} - \bar{\mathbf{y}}_s)$

   until $\frac{|\Delta\mathcal{J}|}{\mathcal{J}^{(0)}} < e_J$ **or** $j \geq j_{\max}$
17. $\mathbf{z}^{\mathrm{a}} = \mathbf{z}^{\mathrm{b}} + \mathbf{A}\mathbf{w}^{\mathrm{a}}$
18. $\mathbf{E}^{\mathrm{a}} = \mathbf{z}^{\mathrm{a}}\mathbf{1}^{\mathrm{T}} + \sqrt{N-1}\mathbf{A}\left(\mathbb{H}^{\mathrm{a}}\right)^{-\frac{1}{2}}\mathbf{U}$

2.3. Estimation of the background error covariance matrix

One of the major issues raised by DA methods is that they require the estimation of the background and observation error covariances. For ensemble-based methods, such as the IEnKS, the background error is represented by a limited number of vectors – the ensemble members – and the problem is to chose the background ensemble for the first analysis cycle. In fact, for subsequent analysis cycles, the analysis ensemble obtained at the end of a cycle can be used as the background ensemble for the next analysis cycle. In the present study, we only consider one analysis cycle such that it is crucial to appropriately choose the initial background ensemble.

In the present study, we first estimate the background error covariance matrix associated with the BCs of the local scale domain in order to construct the background ensemble. In fact, if the background error covariance matrix is known, the background ensemble might either be randomly sampled, according to this matrix, or it might consist of the eigenvectors associated with the largest eigenvalues of this matrix. The estimation of the background error covariance matrix is here based on the statistical analysis of a time series of the control vector associated with the BCs. Such a time series is estimated using the results of three years of simulation with the mesoscale model WRF (Skamarock et al., 2005) at the location of the CFD domain. We thus quantify the uncertainty on inflow BCs based on results from a mesoscale model, similar to what has been done in García-Sánchez and Gorlé (2018). In what follows, we define the control vectors used in the 2D and 3D experiments, then we explain how we use three years of WRF simulations to obtain a series of realisations of these control vectors. Eventually, we present the method used to construct the background error covariance matrix from these time series.

2.3.1. Definition of the 2D and 3D control vectors

In this study, we consider stationary BCs and steady-state wind fields simulated with the CFD model *Code_Saturne*. At first, we consider a 2D domain in an XZ plane for which the vector of BCs corresponds to the vertical profile of wind speed in the plane of the domain, prescribed at the western boundary. We assume that, at the boundary, the wind is only horizontal, such that we only analyse the west-east component of the velocity, referred to as u. The control vector of BCs corresponds to the values of u defined in 21 vertical levels.

For 3D studies, several vertical profiles along the four lateral boundaries – West, South, East, and North - are required. Each border of the 3D *Code_Saturne* domain is 4km long, which corresponds to 6 grid points in the WRF domain (green squares on Fig. 1a). In order not to overestimate the horizontal correlations between the vertical profiles, we only consider 6 vertical profiles of BCs for each border, equally distributed (red circles on Fig. 1a). In total, 20 vertical profiles are included in the control vector and analysed. In 3D, both the west-east (u) and the south-north (v) components of the wind velocity are considered and each profile is defined in 21 vertical levels, such that the control vector in 3D is of size: $l = 20 \times 21 \times 2 = 840$.

2.3.2. From WRF results to time series of control vector estimates

The WRF simulations cover a mesoscale domain of 54km $\times$ 54km, with a horizontal resolution of 5km. Vertically, the domain extends up to 19km above the ground with variable vertical resolution from 15.1m near the ground to 1.7km at the top of the domain. The WRF simulations are available every hour and cover a three-year period. The local scale domains used for *Code_Saturne* simulations are smaller. The 2D domain extends over 8km along x with a 25m horizontal resolution. In 3D the domain extends over 4km $\times$ 4km with a horizontal resolution of 100m. The vertical resolution is finer in *Code_Saturne* than in WRF, especially near the ground: the first 100 meters are described by only 4 vertical levels in WRF and 7 levels in *Code_Saturne*.

To estimate the value of the control vector for each WRF result, we proceed in two steps. First, we perform a horizontal bilinear interpolation from the four closest WRF profiles to the location of each BC profile. Second, we perform specific processing of the lowest vertical levels of the profile obtained after the first step. Indeed, the topography is less precise in the WRF domain than in the *Code_Saturne* domain, such that the wind profiles do not necessarily cover the same vertical region in WRF as in *Code_Saturne*; the *Code_Saturne* topography, which is more accurate, might be locally higher or lower. In the 2D case studied

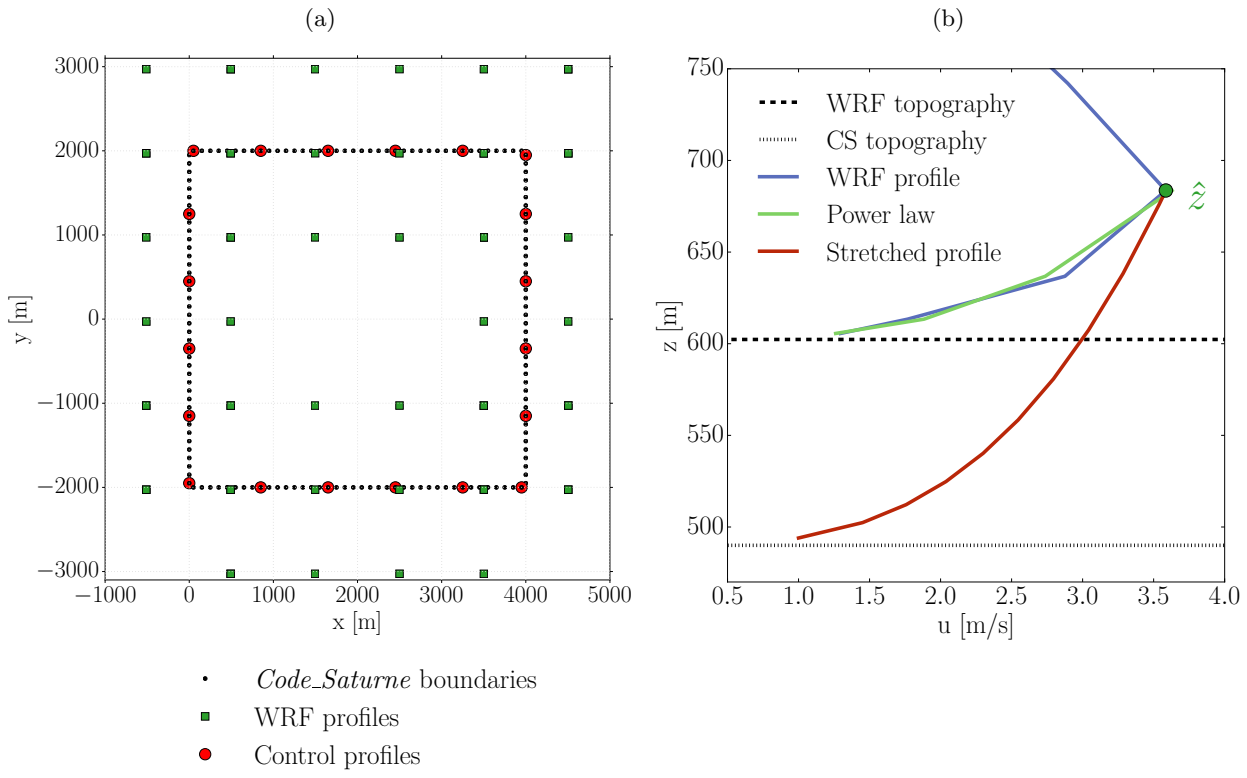


Figure 1: Methodology of the adaptation of WRF simulation results to estimate the BCs prescribed for *Code_Saturne* simulations. a) View from above of the *Code_Saturne* boundaries of the 3D domain. They correspond to 160 vertical profiles shown as black dots. The control vector only includes 6 vertical profiles per border, equally distributed, shown as red disks on the figure. The wind speed values at each red disk is obtained from a bilinear interpolation of the four closest WRF profiles (green squares). b) Example of profile stretching near the ground, required due to the differences of topography resolution between WRF and *Code_Saturne* domains. The WRF profile (blue) is kept above $\hat{z}$ and replaced by the stretched power law (red) below.

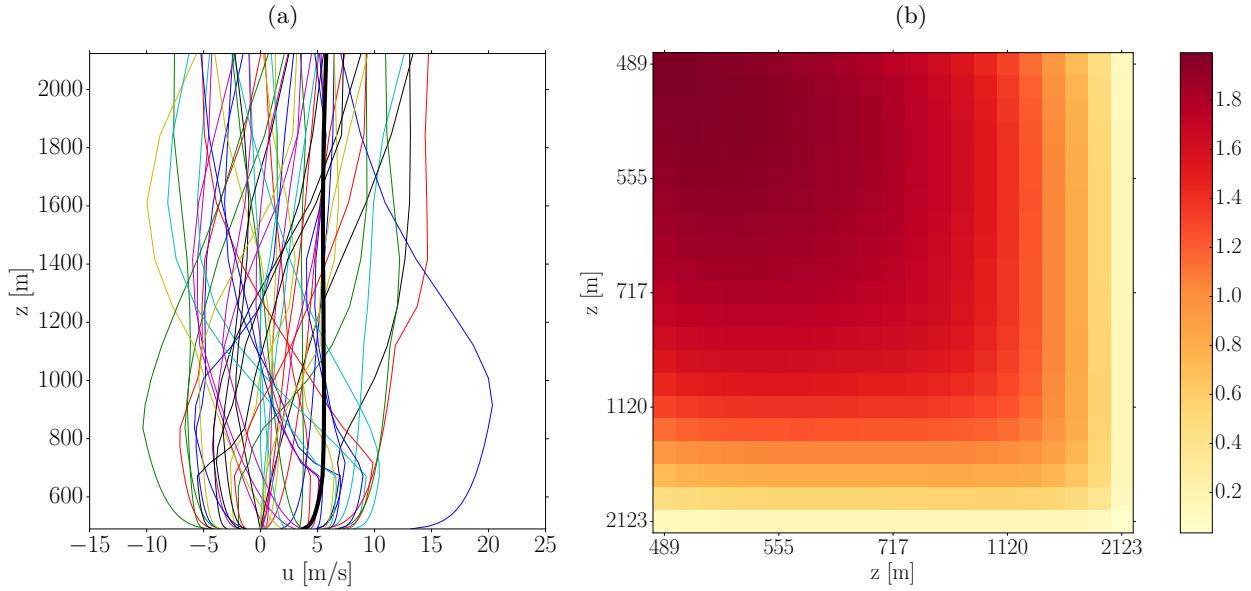


Figure 2: a) Example of 50 vertical profiles of wind velocity obtained from WRF results interpolated to the location of the western boundary of the 2D *Code_Saturne* domain and stretched near the ground using a power law. The thicker, black line shows the vertical profile of mean absolute velocity, averaged over the 2.10^4 profiles obtained from 3 years of WRF simulations. b) Background error covariance matrix for the vector of BCs prescribed for the 2D *Code_Saturne* simulations. The correlations are estimated from the climatological variability and the variances correspond to the mean WRF simulation error.

here, the interpolated WRF profile starts at an absolute height of 605m while the *Code_Saturne* profile starts at 490m. The first 8 levels of *Code_Saturne* (out of 21 levels in total) are not covered by the WRF domain. The horizontally interpolated WRF profiles have to be vertically stretched to fit the *Code_Saturne* grid near the ground. One should note that the reverse situation where the *Code_Saturne* topography is higher than the WRF topography might also occur, in which case the WRF profiles should be contracted.

For each BC profile obtained by horizontal interpolation, we find the power law that best fits the profile near the ground. This law is then used to extrapolate the profile down to the *Code_Saturne* topography, as shown on Fig. 1b. More details on this step of streching (or contraction) is given in Appendix C. We apply the two steps explained above to all the WRF results to obtain a series of more than 2×10^4 realisations of the control vector. Fig. 2a shows an example of 50 values of the 2D control vector. This time series is used to construct the background error covariance matrix, as explained in the next section.

2.3.3. Construction of the background error covariance matrix

We hypothesise that the background error covariance matrix, related to the vector of BCs, is a combination of the climatological variability in WRF simulations and of the mean error made by WRF at a given height. We thus estimate the correlations of the background error covariance matrix $\mathbf{B}$ from a climatological covariance matrix ($\mathbf{V}$) and the variances of $\mathbf{B}$ correspond to the estimated WRF errors ($\boldsymbol{\lambda}$). Each coefficient of the background error covariance matrix $\mathbf{B}$ is thus given by:

$$B_{i,j} = \frac{V_{ij}}{\sqrt{V_{ii}V_{jj}}}\sqrt{\lambda_i \lambda_j}. \tag{24}$$

The climatological covariance matrix is estimated from the series of control vector estimates, obtained from WRF simulations as explained in the previous section (Fig. 2a). The matrix $\mathbf{V}$ thus represents the climatological variability of velocity, in time, simulated by WRF and is very informative with respect to the patterns of covariances.

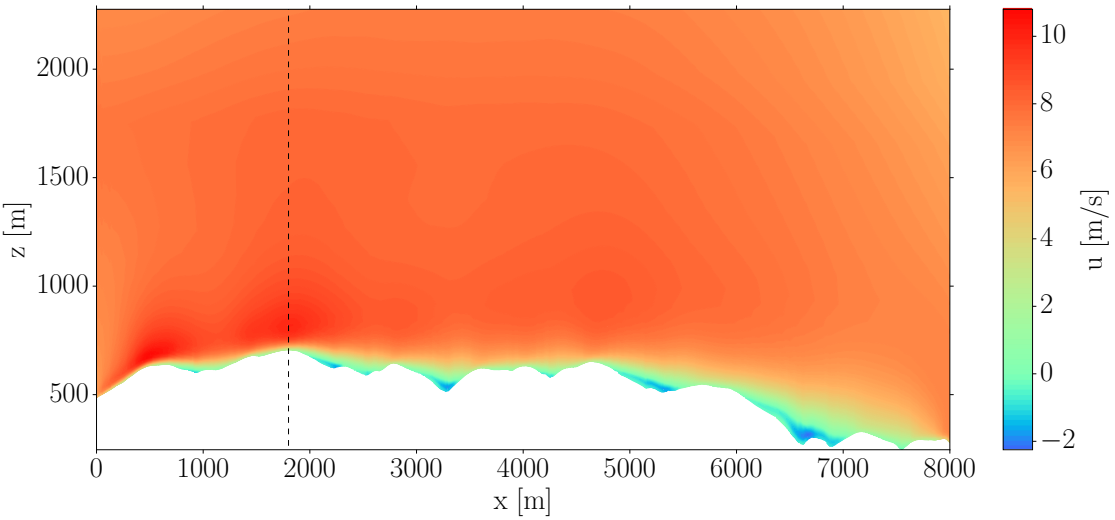


Figure 3: Wind field simulated with *Code_Saturne* over the 2D vertical domain. The vertical dashed line shows the location of the virtual instrumented mast which provides synthetic observations.

In order to get a background error covariance matrix that represents well the confidence given in WRF results, we set the variances of $\mathbf{B}$ equal to the mean error made by WRF. This error mainly depends on the distance to the ground (Shimada et al., 2011, Fig. 3a) and increases with height. In fact, the influence of topography is smaller at higher levels such that the WRF simulations are generally more precise there. From data shown in Shimada et al. (2011), we can deduce an approximate expression of the mean absolute WRF error as a function of height:

$$\epsilon_{\mathrm{w}}(h) = \begin{cases} |2 - 3 \times \frac{h}{2500}| & \text{if } h < 2500\text{m}, \\ 1 & \text{if } h > 2500\text{m}. \end{cases} \tag{25}$$

For each of the l control variables, defined at the vertical level h_i, we compute the coefficient $\lambda_i = \epsilon_{\mathrm{w}}(h_i)$.

Fig. 2b shows the background error covariance matrix associated with the error in the background vector of BCs, estimated using equation 24.

3. Results of twin experiments in 2D

3.1. Experimental set up

At first we consider a 2D vertical section of the domain over realistic topography (Fig. 3). The wind field is simulated over this 2D domain with the open source CFD model *Code_Saturne*. In all the atmospheric simulations performed with *Code_Saturne*the flow is considered steady, the density is assumed constant, and turbulence is modelled by a constant mixing length equal to 10m. The RANS closure model is used such that the simulated variables, and therefore the velocity components, correspond to ensemble means. Symmetric BCs are set for lateral faces, the top boundary is an inlet and the eastern face is a free outlet. The inlet vertical wind profile is prescribed at the western boundary. It is this vertical profile defined in 21 vertical levels that we try to retrieve with DA. The BCs are constant in time and the simulations are run over a sufficiently long time for steady state to be reached.

A reference simulation is performed with a prescribed boundary profile, considered as the truth, shown on Fig. 3. We can see that horizontal wind accelerates over crests, which therefore represent a preferred location to install wind turbines. Wind speed values are observed at $x = 1800$m, corresponding to a local crest (dashed line on Fig. 3), and at five different heights above the ground: 10m, 25m, 50m, 75m, and 100m. These wind speed values constitute pseudo-observations used in the DA process. To account for

instrumental errors, which are generally of the order of magnitude of a few centimetres per second, Gaussian white noise with a variance of $10^{-3}\mathrm{m}^2/\mathrm{s}^2$ is added to these pseudo-observations. Observation error includes the instrument error and the representativeness error which we represent by a diagonal observation error covariance matrix

$$\mathbf{R} = r\mathbf{I} \tag{26}$$

with $r = 0.1\mathrm{m}^2/\mathrm{s}^2$.

The departure of the background from the true profile of inlet wind velocity is consistent with the background error covariance matrix $\mathbf{B}$ defined in Section 2.3. Note that the largest eigenvalue of $\mathbf{B}$ is equal to $29.5\mathrm{m}^2/\mathrm{s}^2$.

For 3D-Var, the increment used in the finite difference approximation is set to 10^{-2}. The control space, in which the cost function is minimised, corresponds to the vector of BCs, defined in 21 vertical levels.

For the IEnKS and 3D-Var, the same criterion of convergence is defined for the cost function: the minimisation algorithm stops when the change in the cost function, relative to the initial value of $\mathcal{J}$, is smaller than $e_{\mathrm{J}} = 1\%$.

3.2. Results with the IEnKS in 2D

In order to analyse the results obtained with DA methods, we consider two statistical indicators. First, the *error* refers to the departure from the truth as defined in equation 3, and corresponds to the ensemble mean. Second, the *uncertainty* is related to the ensemble spread and refers to the confidence interval of the BCs' estimate, and of the resulting wind field simulations. We also compute the root mean square error (RMSE) of the simulated wind field with a given set of BCs, compared to the wind field obtained when the reference BCs are prescribed.

With the 2D configuration presented above, we perform a sensitivity analysis to the number of members and to the method used to generate the initial ensemble (not shown). We consider ensembles of 2, 3, 5, 8, 10, and 20 members. For each ensemble size, we consider two methods of ensemble generation. In one case, the members correspond to the eigenvectors associated with the largest eigenvalues of the singular value decomposition (SVD) of the background error covariance matrix. The second method consists in a random sampling of the members, according to the background error covariance matrix. With this second method, we repeat the experiment 15 times for each ensemble size. In both cases, the ensembles are recentred to ensure that the anomaly matrix has zero mean.

With the first method of ensemble generation based on SVD, the error in the analysis BCs and the global RMSE of the simulated wind field depend very little on the number of members. Even with very small ensembles the IEnKS is able to correct the BCs and to reduce the error in the simulated wind field. This is consistent with the fact that the control variables are highly correlated in the vertical.

With random samplings of the ensemble members, on average, the error in the BCs and the RMSE of the wind field do not depend much on the number of members. However, the variability of results obtained with the 15 ensembles of same size is quite large, especially for small ensembles (2 or 3 members). With both methods of ensemble generation, the IEnKS always converges in 2 or 3 iterations. The observed sensitivity to random sampling for a given ensemble size motivates us to chose the method based on SVD for the following experiments. Since the results are not sensitive to the number of members with this method, we decide to keep 3 members for what follows.

We also verify that in the lower atmosphere ($z < 800\mathrm{m}$), the analysis is nearly insensitive to the background. We have randomly drawn 15 values for background error, according to the same background error covariance matrix, and we assimilate the same observations. The analysis profiles of BCs are quite similar to each other below 800m, whereas in the upper layers of the atmosphere, the analysis depends more on the background. These results are consistent with the fact that observations are only available below 800m and that a large weight is given to the observations ($\mathbf{R}$ is much smaller than $\mathbf{B}$). In all the cases, the RMSE computed over the whole domain is smaller for the analysis than for the background and the analysis RMSE is smaller than 0.5m/s for the 15 experiments.

Finally, we perform a sensitivity analysis to verify that the results do not depend on the parameters of the CFD simulations. In twin experiments, the results are almost insensitive to the mixing length or to mesh resolution.

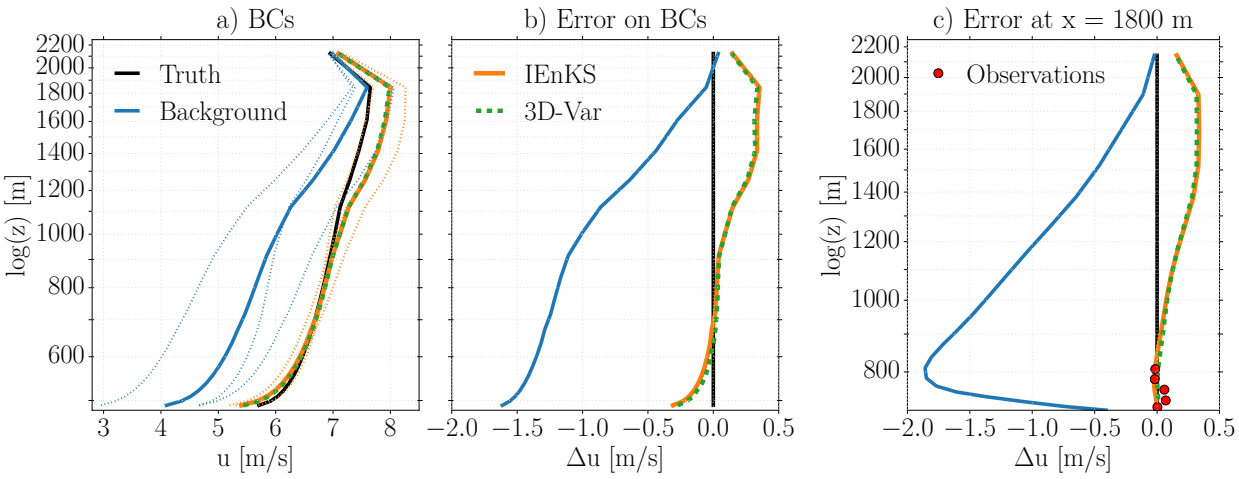


Figure 4: Results in 2D with the IEnKS (ensemble of 3 members) and 3D-Var. a) Vertical profile of velocity used as BCs in 2D: truth (black), background (blue), and analysis (IEnKS: orange and 3D-Var: dashed, green) profiles. The background (dotted blue) and analysis (dotted orange) ensembles of the IEnKS are also shown. b) Departure from the truth of the background and analysis vertical profiles of velocity used as BCs in 2D. c) Departure from the true vertical profile of velocity simulated at the location of observations, $x = 1800$m, for the background and analysis profiles. The noisy observations (red dots) are also shown.

3.3. Comparison between the IEnKS and 3D-Var in 2D

The IEnKS and 3D-Var are both used to solve the DA problem defined in Section 3.1. The control vector contains the 21 values of horizontal velocity, constituting the vertical wind profile prescribed as western BC.

Fig. 4 shows the results obtained with the IEnKS and 3D-Var in this 2D case. The first panel (Fig. 4a) shows the vertical profiles of BCs for the reference simulation (black), the background (blue), and the analyses obtained with the IEnKS (orange) and 3D-Var (green). The background (dotted blue) and analysis (dotted orange) ensembles are also shown. To better show the impact of the DA methods, the true values of velocity are subtracted to present the background and analysis errors in the BCs (Fig. 4b). Similar errors are shown at the location of observations ($x = 1800$m) at the end of the simulation (Fig. 4c). It can be seen that with both methods the error is largely reduced for the BCs, and thus within the domain, especially where observations are available. The two methods give very similar results, which is consistent with the fact that they minimize the same cost function.

Table 1 presents the error in the BCs and the subsequent error in the simulated velocity field. The maximum and the mean absolute error (MAE) is given for the BCs. Regarding the error of the velocity field, we give the local maximum and the root mean square error computed over the whole 2D domain (RMSE). These error statistics are given for the background and for the analyses obtained with the IEnKS and 3D-Var. We can see from this table the large sensitivity of the velocity field to BCs. The error in the background BCs is of 1.07m/s on average and smaller than 1.61m/s for all the control variables. However, it triggers errors in the domain that locally reach 2.48m/s. The results presented in Table 1 show that the two DA methods have the same capacity to greatly reduce the error in the BCs and thus in the whole velocity field.

In the table, we also give the number of integrations needed for the methods to converge. The inner algorithm of the IEnKS converges in 2 iterations and we consider an ensemble of $N = 3$ members, such that the IEnKS requires only 6 integrations of *Code_Saturne*. The minimisation algorithm used for 3D-Var requires 5 iterations which correspond to more than 100 integrations. In fact, for each iteration, $l + 1$ integrations are required to estimate the gradient of the cost function by finite differences, where $l = 21$ is the size of the control vector. In this case, the IEnKS is thus much more efficient regarding computational cost.

The spread of the background ensemble used in the IEnKS characterizes the estimated uncertainty in

	Error in BCs (m/s)		Error in simulated velocity field (m/s)		STD of simulated ensemble (m/s)		Nb of integrations
	max	MAE	max	RMSE	max	mean	
Background	1.61	1.07	2.48	1.06	1.39	0.59	
IEnKS	0.35	0.14	0.35	0.15	0.28	0.13	6
3D-Var	0.34	0.13	0.34	0.15			110

Table 1: Results in 2D with the IEnKS (ensemble of 3 members) and 3D-Var on the same case (same background and same observations). Error in the BCs (maximum and mean absolute error - MAE), error in the simulated velocity field (local maximum and root mean square error - RMSE), standard deviation (STD) of the ensemble of simulated velocity fields (local maximum and average over the whole 2D domain), and the number of model integrations necessary to compute the analysis.

the background BCs. Similarly, the spread of the updated ensemble, at the end of the IEnKS analysis, provides a diagnostic of the posterior error for the BCs. For each ensemble member – which corresponds to a profile of BCs – we run the CFD model *Code_Saturne* forced with these BCs. We thus obtain background and analysis ensembles of simulated wind fields. The spread of these ensembles characterise the accuracy of the simulated wind field. Fig. 5a shows the standard deviation (STD) of the ensemble of simulated wind fields. The larger the standard deviation is, the less precise the velocity field, simulated with the ensemble mean (here the background) prescribed as BCs. We can see that the standard deviation is particularly large between 50m and 100m above the ground, especially over local crests, where it is the most crucial for wind power assessment. As an example, we analyse more precisely the vertical profile of wind accuracy above a local crest ($x = 3600$m, shown by the green triangle on Fig. 5a). The vertical profiles of STD of the background and analysis ensembles of simulated wind fields are shown on Fig. 5b. We can observe that thanks to the IEnKS, the uncertainty is largely reduced near the ground – where observations are available – and that the influence of observations vanishes with height. Even at high altitudes the IEnKS helps to reduce the uncertainty in the simulations. This vertical pattern of analysis STD is observed in all the 2D domain. It is observed that for 90% of the grid points, the standard deviation is divided by more than 1.6 between the background and the analysis. The points where the standard deviation is not reduced are points where the background standard deviation is already small. Table 1 gives the local maximum of STD and the mean computed over the whole 2D domain. The large decrease in these values of STD proves that, besides decreasing the mean error in the BCs, the IEnKS helps to increase the accuracy of the control vector estimate and consequently of the simulated wind field.

The twin experiments performed in 2D have allowed to compare the performances of the IEnKS with the reference method 3D-Var. In order to gradually increase the complexity of the test cases, we first perform twin experiments with the 3D domain (Section 4) and then we assimilate field measurements (Section 5).

4. Results of twin experiments in 3D

4.1. Experimental set up

We consider in this section a 3D domain which extends from $x = 0$m to $x = 4$km, from $y = -2$km to $y = 2$km, and from $z = 288$m to $z = 2318m$. As explained in Section 2.3.1, the control vector in 3D is of size 840, which corresponds to 20 vertical profiles for u and v, defined in 21 vertical levels (Fig. 1a). This control vector is too large to use 3D-Var with finite differences, consequently we only use the IEnKS in this 3D case.

The true value of the control vector corresponds to one of the WRF results projected on the finer, local grid, as explained in Section 2.3.2. The background is obtained as a perturbation of this true vector of BCs. The departure of the background from the truth is randomly sampled from the background error covariance matrix $\mathbf{B}$, defined in Section 2.3.

We consider synthetic observations coming from 3 virtual instrumented masts located at:

- $x = 2000$m and $y = 0$m (mast M80),
- $x = 1750$m and $y = -1290$m (mast M),

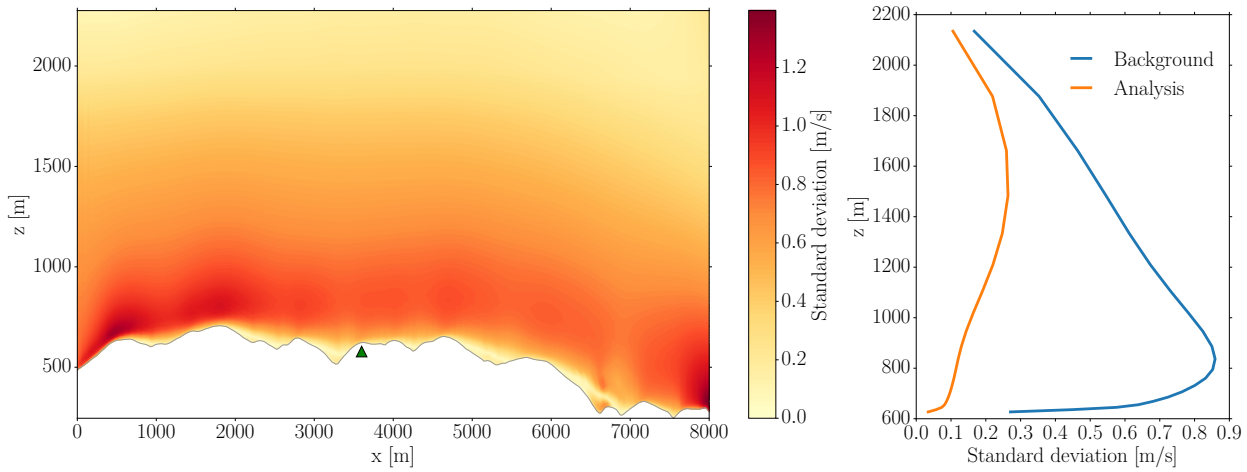


Figure 5: a) Standard deviation of the ensemble of 2D velocity fields simulated with the background ensemble members prescribed as BCs. b) Vertical profile of background (blue) and analysis (orange) uncertainty on u at a local crest ($x = 3600$m, green triangle on panel a). The uncertainty corresponds to the standard deviation of the ensemble of simulated velocity fields.

	Error in BCs (m/s)		**Error in simulated velocity field (m/s)**		**STD of simulated ensemble (m/s)**			
					u		**v**	
	max	**MAE**	**max**	**RMSE**	**max**	**mean**	**max**	**mean**
Background	2.96	0.96	3.40	0.89	1.28	0.41	1.25	0.41
IEnKS	2.00	0.40	1.90	0.20	0.68	0.10	0.38	0.11

Table 2: Results in 3D with the IEnKS (ensemble of 5 members). Error in the BCns (maximum and mean absolute error - MAE), error in the simulated velocity field (local maximum and root mean square error - RMSE), and standard deviation (STD) of the ensemble of simulated velocity fields for u and v (local maximum and average over the whole 3D domain).

- $x = 3670$m and $y = 605$m (mast P).

For each mast, pseudo-observations of u and v are extracted at 10, 25, 50, 75, and 100m above the ground. Consequently, 30 observations are extracted from the reference simulation and Gaussian white noise is added to them. The observation error covariance matrix is $\mathbf{R} = 0.1\mathbf{I}\text{m}^2/\text{s}^2$.

The precision required for the IEnKS algorithm to stop is set to $e_\text{J} = 1\%$.

4.2. Results of twin experiments in 3D with the IEnKS

The IEnKS has been tested on the 3D case with ensembles of 2, 5, 10, 15, 20, 30, and 50 members, all other things remaining equal – such as the background and the noisy observations. We compare the analysis BCs and the steady wind field, obtained when these BCs are prescribed, to the true BCs and wind field. In particular, we compute the MAE of the BCs and the RMSE of the wind field over the whole domain. The sensitivity of the results to the number of members is rather small. The MAE and the RMSE slightly decrease from 2 to 10 members and then increase again. This could be explained by the fact that increasing the ensemble size increases the number of degrees of freedom which is beneficial between 2 and 10 ensemble members. The slight deterioration of the results with ensembles larger than 10 could be explained by overfitting. In fact, the background error covariance matrix is not perfectly known and increasing the ensemble size tends to represent more precisely this incorrect background error statistic. Consequently the analysis is negatively affected. For the sake of numerical efficiency we thus adopt an ensemble of 5 members in what follows.

Table 2 presents some statistics for the background and analysis errors. We can see that the IEnKS helps to divide by more than 2 the MAE of the BCs and thus to divide by more than 4 the RMSE of the

velocity field simulated in the whole domain. In this case the IEnKS has converged in 3 iterations such that it has required 15 integrations of *Code_Saturne*.

Fig. 6 shows the results obtained with an ensemble of 5 members, in the surface defined by $h = 80$m, where h is the relative height above the ground. This height is chosen because it corresponds to a typical height of wind turbine. The background colors represent the topography of the domain, above which the arrows represent the horizontal velocity field (only a subset is shown). Fig. 6a shows the reference wind field obtained with the true BCs and Fig. 6b and c show the departure from this wind field for the background and the analysis. We can see on Fig. 6b that the background error is larger over higher topography. Fig. 6c shows that the IEnKS allows to largely reduce the error in the wind field, especially near the available observations.

Fig. 7a shows the true, background, and analysis values of the vertical profile of u prescribed as BC at $x = 0$m and $y = -350$m. Fig. 7b shows the background and analysis errors for this same profile of BCs and Fig. 7c shows the vertical profile of u simulated at the location of the mast M80 ($x = 2150$m and $y = 0$m). We can see that the analysis error at this location is greatly reduced thanks to the IEnKS. Note that Fig. 7 only shows the u velocity component for one of the 20 vertical profiles included in the control vector. However, the conclusions drawn on this particular case also hold for the v velocity component and for the other vertical profiles.

Fig. 7a and b also show the background and analysis ensemble members. The analysis ensemble is narrower than the background ensemble, meaning that the confidence in the analysis has improved over that in the background. The standard deviation of the velocity fields obtained when the different members of the background ensemble are prescribed as BCs is indicative of the uncertainty in the background wind field. Fig. 8a shows the values of this background STD, for the u-component of velocity, for all the points in the plane of constant height above the ground $h = 80$m (colors). The topography is also shown (contours) and we can observe that the STD is generally larger above the crests. This pattern of uncertainty is largely smoothed by the IEnKS, and the values of standard deviation for u and v are divided by more than 2 for 90% of the grid points between the STD of the background and the analysis ensemble. The local maxima of STD for u and v and the mean calculated over the whole domain are given in Table 2. Globally, the STD of the simulations with the analysis ensemble is smaller than 0.68m/s for u and than 0.38m/s for v. Since the operational objective of DA in the present study is mainly to improve the wind estimate above local crests – where wind turbines might be installed – we present in Fig. 8b the vertical profile of standard deviation over a crest at $x = 800$m and $y = -1400$m (green triangle on Fig. 8a). Fig. 8b shows the vertical profile of STD for u (solid line) and v (dashed line), simulated with the background (blue) and analysis (orange) BCs. For both velocity components, the IEnKS helps to largely reduce the uncertainty in the simulated velocities, and especially near the ground.

5. Results with field measurements in 3D

5.1. Experimental set up

A field campaign has been performed over the 3D domain as described in Section 4.1, between August and February 2007. The observations are available every 10 minutes at 3 masts (M80, M, and P) defined in Section 4.1 and shown in Fig. 6c. We use here these observations in a cross-validation procedure: we assimilate the observations provided by the masts M and P (classical cup anemometers and vanes usually available for wind farms), and we compare the results to the observations provided by the mast M80 (sonic anemometer).

The available observations at M and P are:

- wind speed at 30, 39, and 49m above the ground,
- wind direction at 30 and 47m above the ground.

Assimilating observations of wind direction is quite complex, especially because the wind direction is not well defined for small wind speeds. Since wind direction does not change much with altitude, we consider

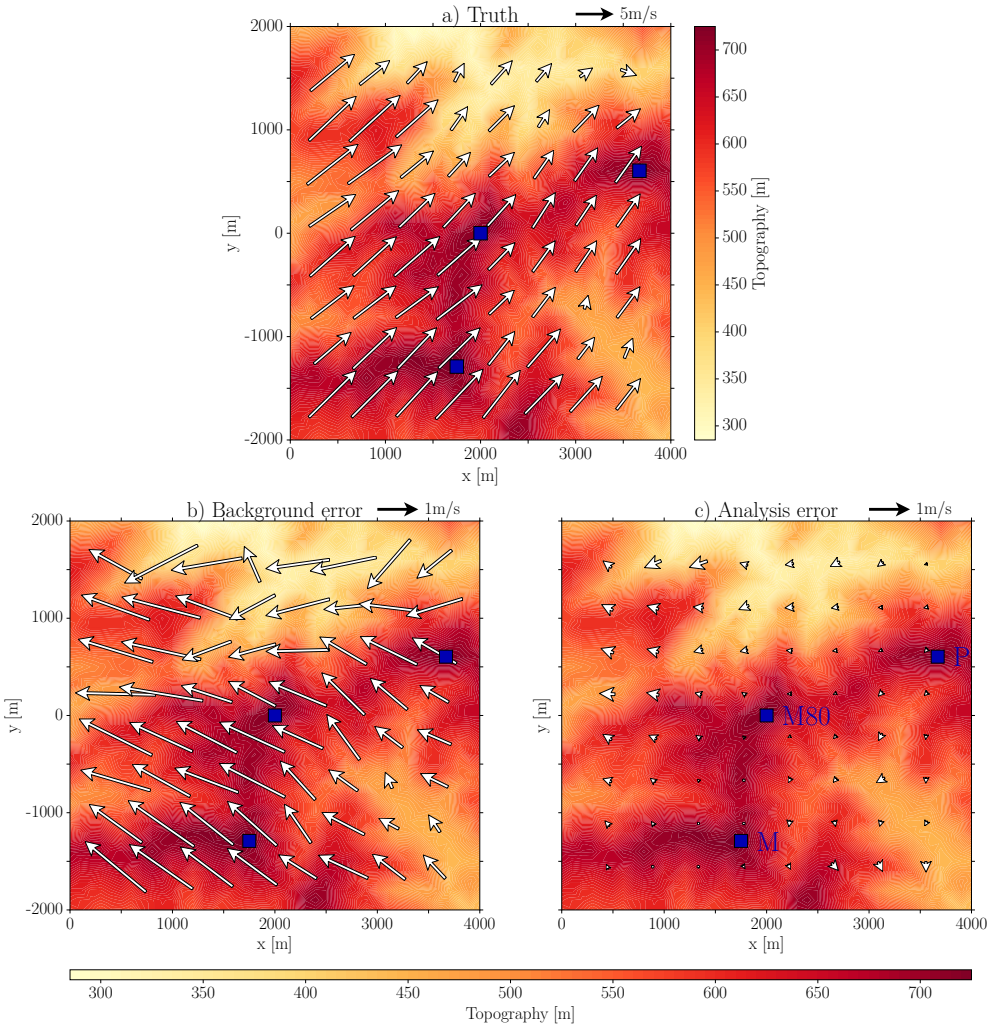


Figure 6: Results of the IEnKS in 3D with 5 members. Relief (shaded) and horizontal velocity field (arrows) in the plane of constant height above the ground $h = 80$m. The blue squares represent the location of the three virtual masts providing pseudo-observations. For the sake of visibility, only 3% of the horizontal velocity vectors are shown on the figures. a) Reference velocity field obtained with true BCs. b) Background error in the velocity field. c) Analysis error in the velocity field.

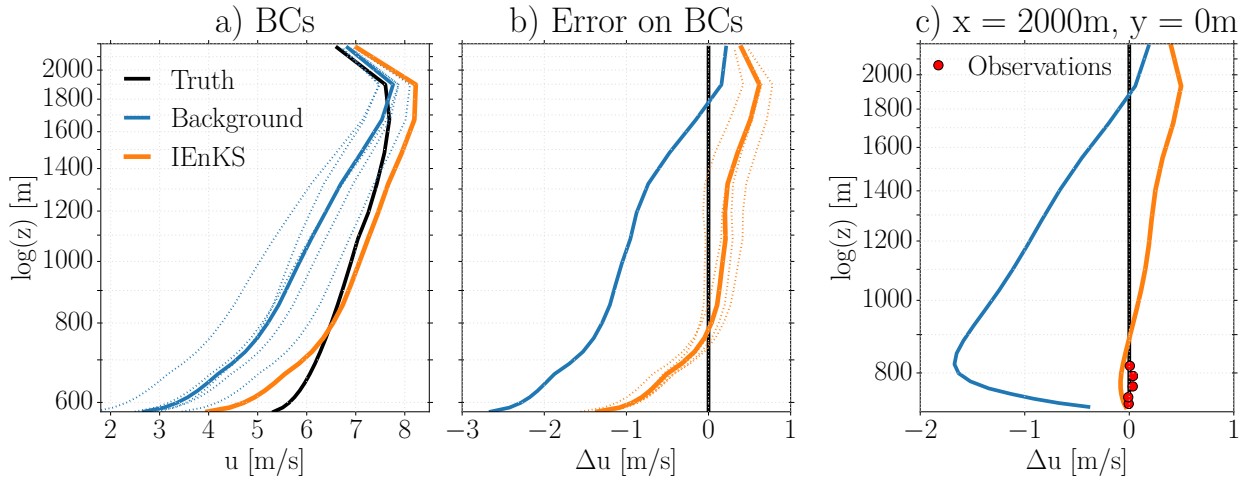


Figure 7: Same as Fig. 4, for the 3D twin experiment. A logarithmic scale is used for the vertical axis. a) Vertical profile of the BC on u located at $x = 0$m and $y = -350$m, in the middle of the western boundary. The dotted, blue curves correspond to the background ensemble members. b) Departure from the true profile of BC for u. The dotted, orange curves correspond to the analysis ensemble members. c) Error in the vertical profile of velocity simulated at the location of the mast M80 ($x = 2000$m and $y = 0$m).

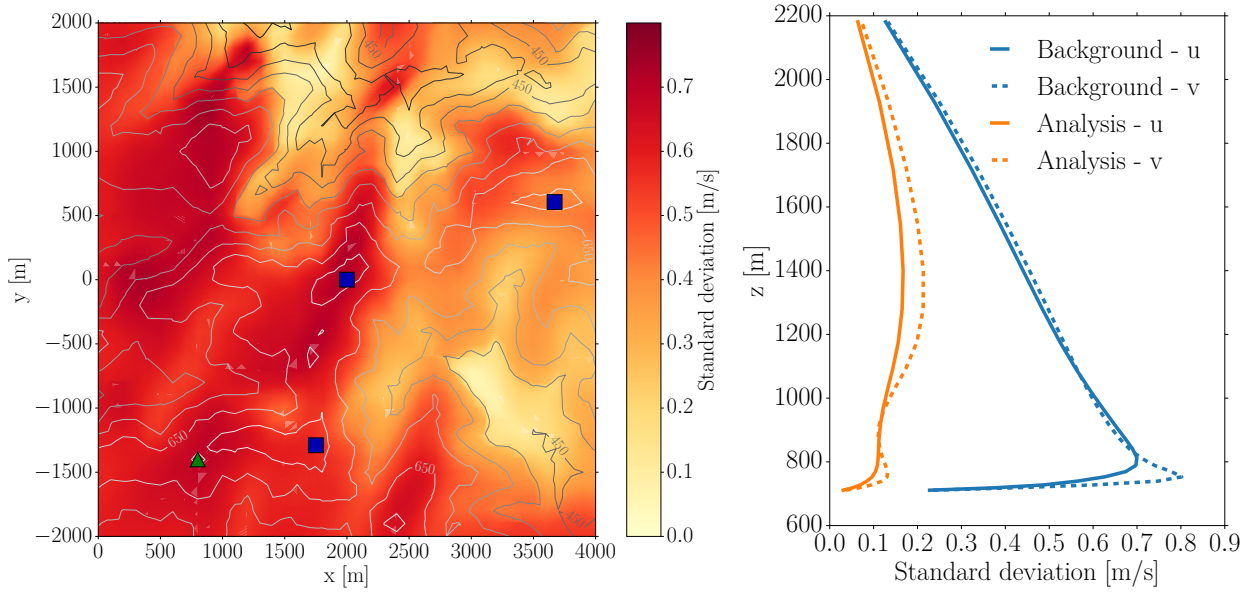


Figure 8: a) Same as Fig. 5a but for the 3D domain. Here we show only the standard deviation for the background values of u (color) in the plane of constant height above the ground $h = 80$m. The contours show topography. The blue squares represent the locations of the three masts. b) Same as Fig. 5b but for the results in 3D for u and v at $x = 800$m and $y = -1400$m, shown by the green triangle on panel a.

that the observation of wind direction at 47m is still valid at 49m. Consequently we convert the observations of wind speed and direction to u and v values at 30 and 49m. At 39m we assimilate directly observations of wind speed, as nonlinear observation operator is well handled by the IEnKS. In the following experiments we thus assimilate 10 observations: the 2 wind components (u and v) at 2 heights (30 and 49m) and the wind speed at 1 height (39m) for the 2 masts (M and P).

To validate the results we use the sonic measurements available at the location of the mast M80 and at 4 different heights above the ground (10, 25, 45, and 78m). The sonics have a sample rate of 10 Hz and here we use the values averaged over 10 minutes.

To test the IEnKS in different meteorological situations and with different background errors, the results of WRF simulations are sorted according to the wind speed, the wind direction, and the departure of the WRF results from the sonic observations. Using a k-means clustering method, 18221 hourly WRF results are clustered in 50 classes and one representative date and time is selected for each class (hereafter referred to as 'representative time'). We thus perform the cross validation process for each of these 50 representative times. For each representative time, the background corresponds to the result of the WRF simulation, projected on the *Code_Saturne* grid as explained in Section 2.3.2.

All the other parameters used in the IEnKS experiments are the same as for the twin experiments (Section 4.1).

5.2. Results of cross validation

We have performed an analysis cycle of the IEnKS for each of the 50 representative times. The IEnKS generally converges in 4 to 6 iterations. The wind field simulated with the analysis BCs thus obtained are compared with the observations from the sonic anemometers. Fig. 9a shows the analysis error – computed by comparing the simulations to the observations at M80 – against the background error for the 50 class representatives. The size of the circles is proportional to the size of the class. We also show the linear regression of the points (dashed orange). We can see that the IEnKS helps reduce the error, except in some cases for which the background error is already small. The regression line indicates that the larger the background error is, the larger the error reduction – in average – by the IEnKS. This figure suggests that if the error between WRF simulations and the available observations in the domain is quite small (< 1.5m/s) it might not be beneficial to resort on DA.

Fig. 9b shows the histograms of the errors of WRF simulations at the location of the mast M80, when compared to sonic observations (green). The histograms are representative of the 18221 situations as the class representatives are duplicated according to the class size. Similar histograms are shown for the error made by *Code_Saturne* simulations when the WRF results are used as BCs (blue). These BCs – and the simulations – correspond to the background of the DA experiment. At the end of the IEnKS analysis, the simulations with *Code_Saturne* forced by the analysis BCs are compared to the sonic observations at M80 to obtain the orange histogram on Fig. 9b. The mean error made by WRF (green), *Code_Saturne* before DA (blue), and *Code_Saturne* after DA (orange) are shown by the stars at the top of the figure. We can see that the use of CFD model does not reduce much the error if the BCs are imprecise. However, after a cycle of the IEnKS, the distribution of the errors is largely shifted toward smaller errors and the mean error is divided by 1.7.

To estimate the impact of DA in a context of wind resource assessment, we assume that a unique wind turbine of 6MW is installed at the location of the mast M80. For each representative situation, we consider the wind speed, at 78 m above the ground, given by: the sonic anemometer, WRF simulations, *Code_Saturne* simulations with the BCs provided by WRF (referred to as CS^{b}), and *Code_Saturne* simulations with the BCs corrected by DA (referred to as CS^{a}). Using the power curve of a 6MW, we can estimate the corresponding power for each representative time. Eventually, we average the 50 power values by weighting them according to the size of the classes. We thus obtain a potential power for the wind turbine installed there. The wind power potential computed from the measurements is equal to 2993kW, WRF results lead to an estimation of 2098kW, CS^{b} gives 1737kW, and CS^{a} gives 2701kW. The wind potential estimated with WRF results is underestimated by 30%, with CS^{b} it is underestimated by 42%, and with CS^{a} by less than 10%. Moreover, the uncertainty on the wind resource estimation is reduced from 4.7% with CS^{b} to 1.3% with CS^{a}. Consequently,

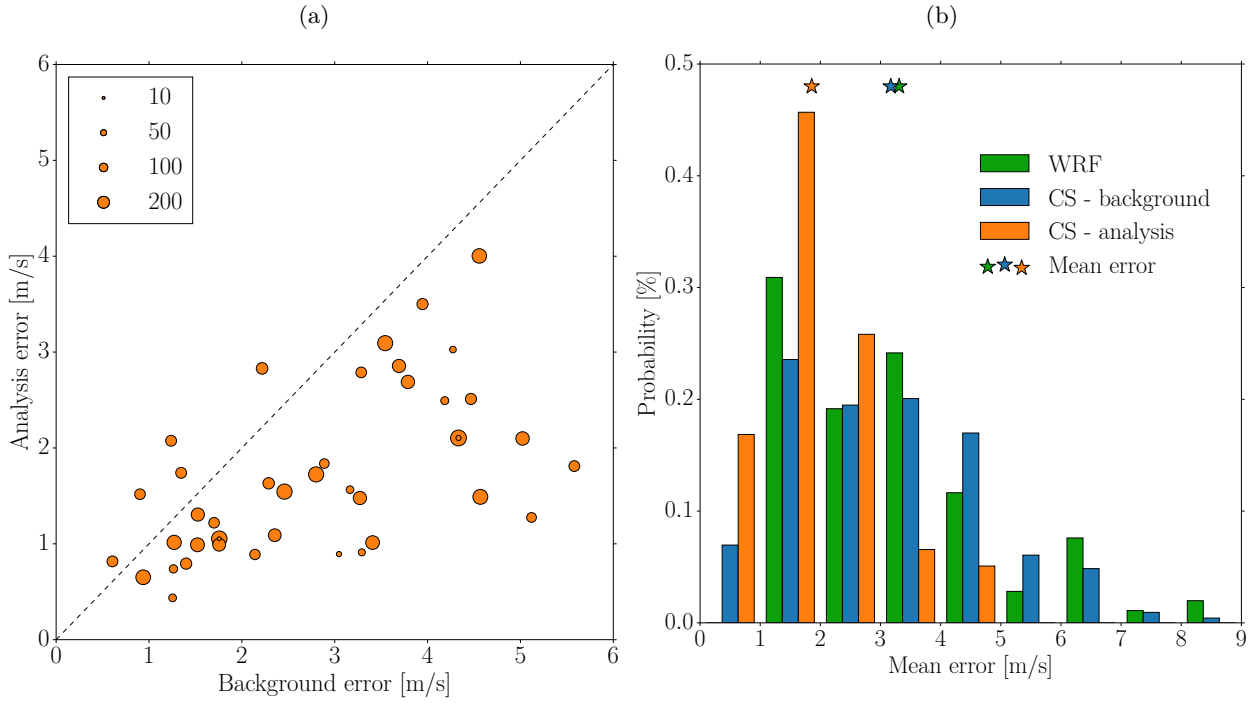


Figure 9: a) Analysis error versus background error for the different meteorological situations representative of the 50 classes. The size of the circles represent the class size. b) Histograms of the difference between simulation results and sonic observations.

the use of the IEnKS to correct the BCs of the 50 representative times allows to largely reduce the error of the wind potential estimation as well as the uncertainty in this estimate.

One can note that CS^b gives worse results than WRF, whereas the mean error made by WRF is larger than the one made by CS^b. This might be explained by the nonlinearity of the power curve and especially the effect of cut in (the wind turbine is cut if the wind speed is smaller than 3m/s). Another source of explanation could be that WRF errors are especially large in situations of strong wind, but the power curve reaches a plateau after 13m/s such that these errors do not impact the wind potential computation.

Note that the observations, and thus all this study, are not available during a full year but only for 8 months. Consequently these estimations are not perfectly representative of the potential of this site. Moreover, a mast equipped with cup anemometers is located just next to the sonic anemometer. If the observations from the cup anemometer is used to estimate the wind power potential, we obtain a value of 2895kW. The difference with the value given previously, obtained with the sonic anemometer measurements, indicates that the observations are not perfect and that the confidence interval is approximately equal to 100kW.

6. Concluding remarks

Local scale atmospheric simulations are mainly driven by boundary conditions. In order to optimally use available observations through data assimilation, the control variables of the DA algorithm must therefore include BCs. In the present study, we show that the iterative ensemble Kalman smoother (IEnKS) can easily be adapted to take BCs into account, instead of initial conditions, as in its original derivation.

This new version of the IEnKS is tested here with twin experiments in two and three dimensions. In 2D (vertical plane), the BCs correspond to the vertical profile of the east-west component of velocity prescribed on one side of the domain. The control vector corresponds to the values of velocity defined in the 21 vertical

levels. The IEnKS is tested in a configuration with 5 pseudo-observations and an ensemble of 3 members. In this configuration, it allows to divide by more than 7 the mean absolute error of the BCs which leads to a large reduction of the global error of the simulated wind field. Moreover, the IEnKS has the great advantage to provide an error estimate of the analysis, thanks to the ensemble. We thus show that the IEnKS helps to increase the accuracy of the BCs and consequently of the wind field simulated over the whole domain. The reference method 3D-Var is also used on this 2D case to provide a comparative basis. In fact, the IEnKS and 3D-Var theoretically minimize the same cost function, albeit for 3D-Var it is minimised in the control space whereas for the IEnKS the cost function is defined and minimised in the ensemble space. Both methods give very similar results, though the IEnKS is more efficient in this case, from a computing-cost perspective.

Twin experiments are also performed with a complete 3D domain of $4\text{km} \times 4\text{km} \times 2030\text{m}$. In this case the control vector corresponds to the BCs on wind velocity (both u and v components) defined on the four sides of the domain. The control vector is here of size 840 and we consider 30 synthetic observations extracted from 3 virtual masts in the domain. With a quite small ensemble (5 members), the IEnKS helps to improve the estimate of BCs and thus to reduce the global error inside the domain, by decreasing both the absolute error and the uncertainty in the estimate.

Finally, observations from a field campaign are used in a cross validation process: the observations from 2 masts are assimilated and the results are compared to the observations provided by a third mast. We have shown that the use of the IEnKS with 5 members helps to reduce the error in most cases. The mean error is then largely reduced which triggers a better estimation of the wind potential of this site. Though, the results have shown that in cases with a small background error (i.e. when the CFD simulations with BCs provided by mesoscale outputs are already close to observations), it might not be beneficial to perform DA. Consequently a good practice could be to use DA only for situations for which the background error is quite large (here larger than 1.5m/s).

In the three cases studied here, the IEnKS adapted to BCs is proved to enhance the accuracy of local scale atmospheric simulations in operationally affordable conditions. In fact, thanks to the IEnKS, the absolute error of the wind field is reduced as well as the uncertainty associated with this error. As a consequence, the wind potential is significantly better estimated thanks to the use of the adapted IEnKS.

In the present study, we keep the same DA pattern (number and location of the assimilated observations). It would be interesting to assess the sensitivity of the results to the number of observations as well as their location, both horizontally and vertically. Such analyses could be used to study best practices in terms of experimental design.

Acknowledgement

We would like to thank J.P. Argaud who shared his expertise on the ADAO module of the Salome platform and helped us to use 3D-Var with *Code_Saturne*. We also want to show our gratitude to G. Angot for providing us with the clustering method and the wind power curve.

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.

Blocken, B., 2014: 50 years of Computational Wind Engineering: Past, present and future. *J. Wind. Eng. Ind. Aerodyn.*, **129**, 69–102.

Blocken, B., A. van der Hout, J. Dekker, and O. Weiler, 2015: CFD simulation of wind flow over natural complex terrain: Case study with validation by field measurements for Ria de Ferrol, Galicia, Spain. *J. Wind. Eng. Ind. Aerodyn.*, **147**, 43–57.

Bocquet, M., and P. Sakov, 2012: Combining inflation-free and iterative ensemble Kalman filters for strongly nonlinear systems. *Nonlin. Process. Geophys.*, **19**, 383–399.

Bocquet, M., and P. Sakov, 2014: An iterative ensemble Kalman smoother. *Quart. J. Royal Meteor. Soc.*, **140**, 1521–1535.

Byrd, R. H., P. Lu, J. Nocedal, and C. Zhu, 1995: A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM J. on Sci. Comput.*, **16**, 1190–1208.

Cattin, R., B. Schaffner, and S. Kunz, 2006: Validation of CFD wind resource modeling in highly complex terrain. *2006 European Wind Energy Association Conference*.

Courtier, P., and O. Talagrand, 1990: Variational assimilation of meteorological observations with the direct and adjoint shallow-water equations. *Tellus A: Dyn. Meteor. Oceanogr.*, **42**, 531–549.

Daley, R., 1991: *Atmospheric data analysis*. Cambridge University Press, 457 pp.

Defforge, C. L., B. Carissimo, M. Bocquet, P. Armand, and R. Bresson, 2018: Data assimilation at local scale to improve CFD simulations of atmospheric dispersion: application to 1D shallow-water equations and method comparisons. *Int. J. Environ. Pollut.*.

Duraisamy, V. J., E. Dupont, and B. Carissimo, 2014: Downscaling wind energy resource from mesoscale to microscale model and data assimilating field measurements. *J. Physics: Conf. Ser.*, **555**, 012 031.

Evensen, G., 1994: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.*, **99**, 10 143.

Franke, J., C. Hirsch, A. Jensen, H. K. C. action C, and U. 2004, 2004: Recommendations on the use of CFD in wind engineering. *Cost action C*, **14**, C1.

García-Sánchez, C., and C. Gorlé, 2018: Uncertainty quantification for microscale CFD simulations based on input from mesoscale codes. *J. Wind. Eng. Ind. Aerodyn.*, **176**, 87–97.

Gu, Y., and D. S. Oliver, 2007: An Iterative Ensemble Kalman Filter for Multiphase Fluid Flow Data Assimilation. *SPE J.*, **12**, 438–446.

Kalmikov, A., G. Dupont, K. Dykes, and C. Chan, 2010: Wind power resource assessment in complex urban environments: MIT campus case-study using CFD Analysis. *AWEA 2010 Windpower Conference*.

Kalnay, E., 2003: *Atmospheric modeling, data assimilation, and predictability*. Cambridge university press, 341 pp.

Kato, H., A. Yoshizawa, G. Ueno, and S. Obayashi, 2015: A data assimilation methodology for reconstructing turbulent flows around aircraft. *J. Comput. Phys.*, **283**, 559–581.

Koohkan, M., 2012: Multiscale data assimilation approaches and error characterisation applied to the inverse modelling of atmospheric constituent emission fields . Ph.D. thesis, Université Paris-Est. https://pastel.archives-ouvertes.fr/pastel-00807468.

Landberg, L., L. Myllerup, O. Rathmann, E. L. Petersen, B. H. Jørgensen, J. Badger, and N. G. Mortensen, 2003: Wind Resource Estimation-An Overview. *Wind. Energy*, **6**, 261–271.

Le Dimet, F.-X., and O. Talagrand, 1986: Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A: Dyn. Meteor. Oceanogr.*, **38**, 97–110.

Li, Y., I. M. Navon, P. Courtier, and P. Gauthier, 1993: Variational Data Assimilation with a Semi-Lagrangian Semi-implicit Global Shallow-Water Equation Model and Its Adjoint. *Mon. Wea. Rev.*, **121**, 1759–1769.

Liu, C., Q. Xiao, and B. Wang, 2008: An Ensemble-Based Four-Dimensional Variational Data Assimilation Scheme. Part I: Technical Formulation and Preliminary Test. *Mon. Wea. Rev.*, **136**, 3363–3373.

Misaka, T., T. Ogasawara, S. Obayashi, I. Yamada, and Y. Okuno, 2008: Assimilation Experiment of Lidar Measurements for Wake Turbulence. *J. Fluid Sci. Technol.*, **3**.

Mons, V., L. Margheri, J.-C. Chassaing, and P. Sagaut, 2017: Data assimilation-based reconstruction of urban pollutant release characteristics. *J. Wind. Eng. Ind. Aerodyn.*, **169**, 232–250.

Palma, J., F. Castro, L. Ribeiro, A. Rodrigues, and A. Pinto, 2008: Linear and nonlinear models in wind resource assessment and wind turbine micro-siting in complex terrain. *J. Wind. Eng. Ind. Aerodyn.*, **96**, 2308–2326.

Peterson, E. W., J. P. Hennessey, E. W. Peterson, and J. P. H. Jr., 1978: On the Use of Power Laws for Estimates of Wind Power Potential. *J. Appl. Meteor.*, **17**, 390–394.

Probst, O., D. Cárdenas, O. Probst, and D. Cárdenas, 2010: State of the Art and Trends in Wind Resource Assessment. *Energies*, **3**, 1087–1141.

Sakov, P., D. S. Oliver, and L. Bertino, 2012: An Iterative EnKF for Strongly Nonlinear Systems. *Mon. Wea. Rev.*, **140**, 1988–2004.

Shimada, S., T. Ohsawa, S. Chikaoka, and K. Kozai, 2011: Accuracy of the Wind Speed Profile in the Lower PBL as Simulated by the WRF Model. *Sola*, **7**, 109–112.

Simões, T., and A. Estanqueiro, 2016: A new methodology for urban wind resource assessment. *Renew. Energy*, **89**, 598–605.

Skamarock, W. C., J. B. . Klemp, J. Dudhia, D. O. Gill, D. M. Barker, M. G. M. Duda, W. Wang, and J. G. Powers, 2005: A description of the advanced research WRF version 2. NCAR Tech. Note NCAR/TN-468+STR, 88 pp.

Sousa, J., C. García-Sánchez, and C. Gorlé, 2018: Improving urban flow predictions through data assimilation. *Build. Environ.*, **132**, 282–290.

Wang, B., L. Cot, L. Adolphe, S. Geoffroy, and S. Sun, 2017: Cross indicator analysis between wind energy potential and urban morphology. *Renew. Energy*, **113**, 989–1006.

Yamada, T., 2004: Merging CFD and atmospheric modeling capabilities to simulate airflows and dispersion in urban areas. *Comput. Fluid Dyn. J.*, **13**, 329–341.

Yang, Y., M. Gu, S. Chen, and X. Jin, 2009: New inflow boundary conditions for modelling the neutral equilibrium atmospheric boundary layer in computational wind engineering. *J. Wind. Eng. Ind. Aerodyn.*, **97**, 88–95.

Zajaczkowski, F. J., S. E. Haupt, and K. J. Schmehl, 2011: A preliminary study of assimilating numerical weather prediction data into computational fluid dynamics models for wind prediction. *J. Wind. Eng. Ind. Aerodyn.*, **99**, 320–329.

Zou, X., F. Vandenberghe, M. Pondeca, and Y.-H. Kuo, 1997: Introduction to adjoint techniques and the MM5 adjoint modeling system. NCAR Tech. Note NCAR/TN-435-STR, 107 pp.

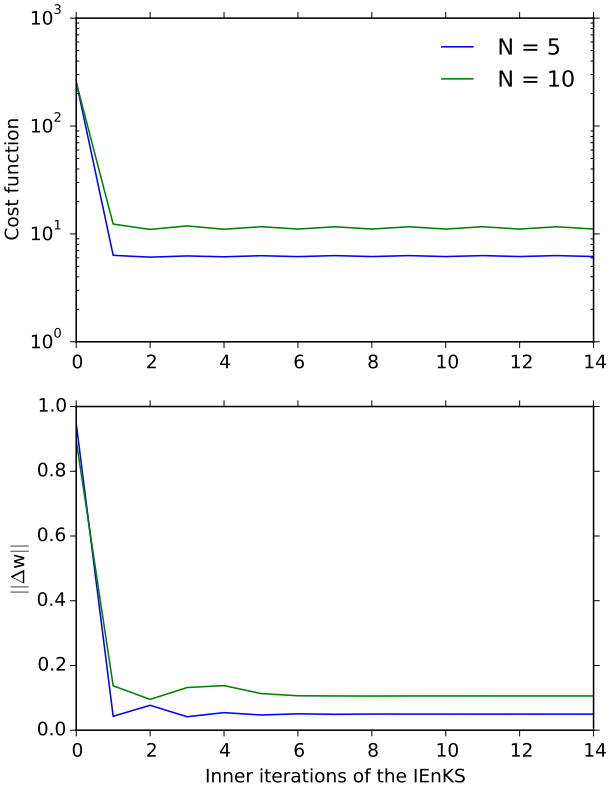


Figure A1: Evolution of (upper) the cost function ($\mathcal{J}$) and (lower) the norm of the weight vector increment $\|\Delta \mathbf{w}\|$ with the inner iterations of the IEnKS for the 2D twin experiment with 5 (blue) and 10 (green) members.

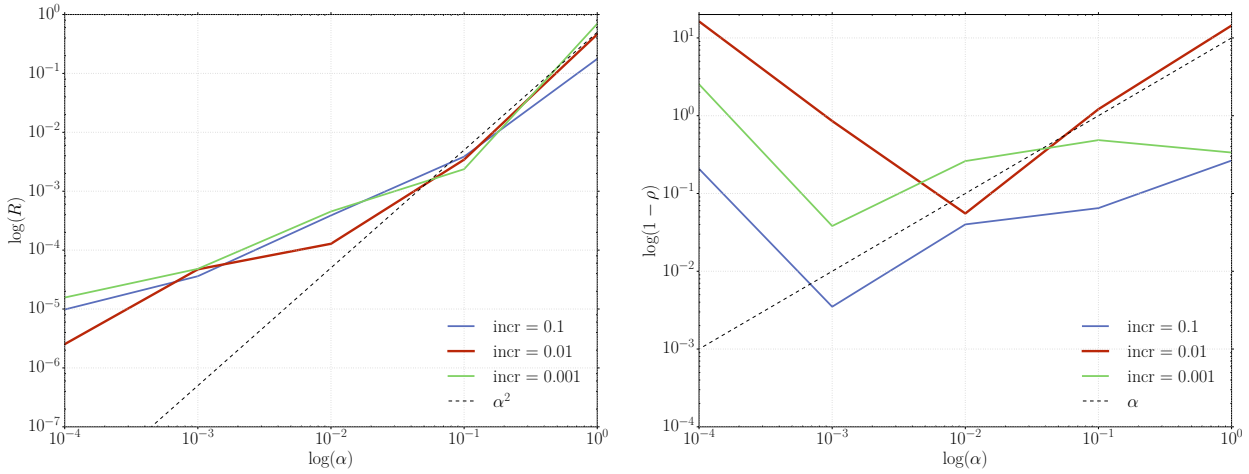
Appendix A. Criterion of convergence of the IEnKS inner minimisation algorithm

The IEnKS cost function is here minimised using a Gauss-Newton algorithm. The convergence criterion for this iterative algorithm is usually defined by the fact that the root mean square of $\mathbf{w}^{(j+1)} - \mathbf{w}^{(j)}$ is smaller than a predefined precision level e: $\|\Delta \mathbf{w}\| < e$. With the CFD model *Code_Saturne*, the cost function defined in 2.1.1 does not show a clear global minimum. In fact, due to some model imprecisions, there is a small region in the ensemble subspace where the cost function is minimal and nearly constant. When the IEnKS has reached this region, the vector $\mathbf{w}$ might change more than a typical value of $e = 10^{-3}$ while the cost function is nearly constant (Fig. A1). In this case the algorithm does not converge according to the criterion on $\|\Delta \mathbf{w}\|$, whereas the minimum of the cost function has been reached. Consequently, we have defined a new criterion of convergence related to the change in the value of the cost function, between two consecutive iterations, relative to the initial value of the cost function. The algorithm is here stopped when $\frac{|\mathcal{J}^{(j+1)} - \mathcal{J}^{(j)}|}{\mathcal{J}^{(0)}} < e_{\mathrm{J}}$. In all the experiments discussed in this paper, $e_J = 10^{-2}$.

Appendix B. Gradient tests for the tangent linear estimation

To assess the performance of the minimisation algorithm, we have verified that the tangent linear of the forward operator ($\mathcal{F}'$) is correctly estimated with finite differences:

$$\mathcal{F}'_{|_{(\mathbf{z})}} \mathrm{d}\mathbf{z} \approx \frac{\mathcal{F}\left(\mathbf{z} + \alpha \mathrm{d}\mathbf{z}\right) - \mathcal{F}(\mathbf{z})}{\alpha}. \tag{B.1}$$



(a) Evolution of the Taylor residue with respect to the multiplication factor α.

(b) Evolution of the difference between the ratio $\rho(\alpha)$ and 1, with respect to α.

Figure A2: Verification tests of the tangent linear estimation for three values of the differential increment, used in the finite difference approximation: 10^{-1}, 10^{-2} and 10^{-2}.

By construction of the cost function, if the tangent linear is well estimated, then the cost function is analytically well calculated. It is thus equivalent to perform the following test with the tangent linear of the forward operator or with the cost function.

At first, we consider the Taylor residue for the tangent linear:

$$R(\alpha) = \frac{\|\mathcal{F}(\mathbf{z} + \alpha \mathrm{d}\mathbf{z}) - \mathcal{F}(\mathbf{z}) - \alpha \mathcal{F}' \mathrm{d}\mathbf{z}\|}{\|\mathcal{F}(\mathbf{z})\|} = O(\alpha^2). \tag{B.2}$$

This residue should decrease as α^2 when α decreases, until the machine precision is reached.

The second test considers a residue ρ which is the ratio between a perturbation of the forward operator and the tangent linear estimate (Zou et al., 1997; Li et al., 1993; Koohkan, 2012):

$$\rho(\alpha) = \frac{\|\mathcal{F}(\mathbf{z} + \alpha \mathrm{d}\mathbf{z}) - \mathcal{F}(\mathbf{z})\|}{\|\alpha \mathcal{F}' \mathrm{d}\mathbf{z}\|} = 1 + O(\alpha). \tag{B.3}$$

When α tends to zero, $\rho(\alpha)$ should linearly converge towards unity.

Fig. A2a shows the decrease of $R(\alpha)$ with respect to α and Fig. A2b shows the difference between the residue $\rho(\alpha)$ and 1, with respect to α. In both figures, the results are presented for three different values of the differential increment ($\mathrm{d}\mathbf{z}$), which is used in the finite difference calculation. The point (in the space of dimension $\mathbb{R}^l$) around which the tangent linear is approximated, corresponds to the reference vertical profile of horizontal wind speed, presented in Section 3. In this case $l = 21$.

From Figs. A2a and A2b, one can see that a differential increment of 10^{-2} gives goods results: the decrease of $R(\alpha)$ is close to α^2 and the decrease of $1 - \rho(\alpha)$ is nearly linear until it reaches a constant value. Thus the differential increment has been set to $\mathrm{d}\mathbf{z} = 10^{-2}$ for all the tests with 3D-Var in the present study.

Appendix C. Stretching (or contraction) of WRF profiles near the ground

In order to stretch (or contract) the WRF profiles on the *Code_Saturne* grid, we assume that the wind velocity profiles near the ground follow a power law, which is a commonly used relationship between height above the ground and wind speed (Peterson et al., 1978):

$$u(h) = u_{\mathrm{R}} \left(\frac{h}{h_{\mathrm{R}}} \right)^{\alpha}, \tag{C.1}$$

where u_{R} is the wind speed at a reference height h_{R} and α is a coefficient which depends on the stability of the atmosphere. One can easily prove that u_{R} and h_{R} are linked such that we set $h_{\mathrm{R}} = 500$m in what follows and the power law is determined by two independent parameters: u_{R} and α. Note that we always refer to relative height above the ground as h whereas z represents absolute altitude.

For each WRF simulation, we find the power law that best fits the wind profile below $\hat{h}^{\mathrm{w}}$, where the superscripts 'w' and 's' refer to the variables in the WRF and *Code_Saturne* domains respectively and the variables with hat correspond to the values at the level where the profiles must match. The level $\hat{h}^{\mathrm{w}}$ is estimated for each profile as the height below which the WRF velocity increases with height. We thus obtain a couple of parameters $(u_{\mathrm{R}}^{\mathrm{w}}, \alpha)$ such that for any $h^{\mathrm{w}} < \hat{h}^{\mathrm{w}}$ the wind velocity simulated by WRF $(u(h^{\mathrm{w}}))$ approximately follows the power law:

$$u(h^{\mathrm{w}}) \approx u_{\mathrm{R}}^{\mathrm{w}} \left(\frac{h^{\mathrm{w}}}{h_{\mathrm{R}}^{\mathrm{w}}} \right)^{\alpha}. \tag{C.2}$$

Fig. 1b shows an example of WRF profile (blue) which is approximated by a power law (green) below $\hat{z} = \hat{h}^{\mathrm{w}} + z_0^{\mathrm{w}}$, where z_0^{w} refers to the elevation of WRF topography there.

To project the vertical wind profiles on the *Code_Saturne* grid, we keep the same power-law exponent α but we recalculate the reference velocity $u_{\mathrm{R}}^{\mathrm{s}}$ such that the new power-profile $u^{\mathrm{s}}(h^{\mathrm{s}})$ matches the WRF profile at $\hat{z}$, where the simulated wind speed is $\hat{u}$. Written in the frame relative to the *Code_Saturne* topography, this junction height is $\hat{h}^{\mathrm{s}} = \hat{h}^{\mathrm{w}} + \Delta z_0$ where $\Delta z_0 = (z_0^{\mathrm{w}} - z_0^{\mathrm{s}})$ is the difference in topography between WRF and *Code_Saturne*. Consequently, we must have:

$$u^{\mathrm{s}}(\hat{h}^{\mathrm{s}}) = u_{\mathrm{R}}^{\mathrm{s}} \left(\frac{\hat{h}^{\mathrm{w}} + \Delta z_0}{h_{\mathrm{R}}} \right)^{\alpha} \equiv \hat{u}, \tag{C.3}$$

$$\text{or: } u_{\mathrm{R}}^{\mathrm{s}} = \hat{u} \left(\frac{h_{\mathrm{R}}}{\hat{h}^{w} + \Delta z_0} \right)^{\alpha}. \tag{C.4}$$

Using the couple of parameters $(u_{\mathrm{R}}^{\mathrm{s}}, \alpha)$ thus obtained, we can compute the wind velocity profile for all the *Code_Saturne* vertical levels below $\hat{z}$, shown on red on Fig. 1b. We eventually obtain a full vertical profile by keeping the result of WRF simulation above $\hat{z}$ (blue curve on Fig. 1b) and completing it with the stretched power law below (red curve on Fig. 1b).