Inflow and initial conditions for direct numerical simulation based on adjoint data assimilation

Alejandro Gronskis, Dominique Heitz, Etienne Mémin

Inflow and initial conditions for direct numerical simulation based on adjoint data assimilation

A. Gronskis[a,1], D. Heitz[b,c], E. Mémin[a]

[a]INRIA Rennes - Bretagne Atlantique, Campus universitaire de Beaulieu, F-35042 Rennes, France.
[b]Irstea, UR TERE, F-35044 Rennes, France.
[c]Université européenne de Bretagne, Rennes, France.

Abstract

A method for generating inflow conditions for direct numerical simulations (DNS) of spatially-developing flows is presented. The proposed method is based on variational data assimilation and adjoint-based optimization. The estimation is conducted through an iterative process involving a forward integration of a given dynamical model followed by a backward integration of an adjoint system defined by the adjoint of the discrete scheme associated to the dynamical system. The approach's robustness is evaluated on two synthetic velocity field sequences provided by numerical simulation of a mixing layer and a wake flow behind a cylinder. The performance of the technique is also illustrated in a real world application by using noisy large scale PIV measurements. This method denoises experimental velocity fields and reconstructs a continuous trajectory of motion fields from discrete and unstable measurements.

Keywords: Data driven simulation, Variational assimilation, Automatic differentiation, Adjoint equations, Inlet condition specification, PIV, Wake flow

1. Introduction

In spite of significant developments in computational methods over the past few decades, a number of real flows with moderate Reynolds number

---

[1]*Present address:* LFD, Facultad de Ingeniería, Universidad de Buenos Aires, Argentina.

are still very difficult to simulate accurately due to complex and unknown boundary conditions. Such boundary conditions arising for instance in real world turbulent boundary layer flows require to be properly formalized and taken into account in order to faithfully reproduce the main flow features. Besides, such flows require a high computational effort as they are characterized by unsteady mixing due to eddies at many scales. To overcome these issues and limit the computational cost, new strategies that aim at simulating only a region of interest in the flow have recently been developed. A critical issue that arises as a direct consequence is the imperative need for the correct specification on this simulation region of all the boundary conditions, which become unsteady. Generating proper inlet conditions for unsteady simulations of spatially developing flows requires the generation of vector fields evolving in time in agreement with the spatiotemporal dynamics of the flow.

Usually, turbulent mean velocity profiles can be used and some random noise superimposed in order to start some artificial perturbation supposed to mimic the real instantaneous behaviour of the turbulent flow. In this case, a lack of realistic turbulent structures induces a transient region near the inlet. This development region has no physical significance in general and is of no practical interest. Another solution consists in using an auxiliary (temporal or spatial) simulation where velocity data are stored at a given section corresponding to the inlet boundary of the main simulation (Lund et al., 1998). Such a technique provides satisfactory results for specific studies, but its application remains difficult for complex flow configurations. Moreover, such calculations are expensive in time and/or storage capacities. Procedures based on low order dynamical models and proper orthogonal decomposition (POD) (Druault et al., 2004; Perret et al., 2008) are probably less expensive for the generation of inflow data than the above method. However, such a procedure requires that direct numerical simulation (DNS), large-eddy simulation (LES) or experimental datasets are available to compute the most energetic modes, with an appropriate set of time dependent projection coefficients providing phase information. The reconstructed inflow data are more realistic, however the experimental databases usually suffer from either low spatial resolution, common with hot-wire anemometry measurements or limited temporal resolution, when using Particle Image Velocimetry (PIV) measurements. Special treatment must then be applied to alleviate the low resolution issues. Alternative methods rely on synthetic turbulence generation (Keating et al., 2004). This is of particular interest when only limited

turbulence statistics data are available for the procedure. The usual approach consists in generating a velocity signal that exibits some statistical profiles learned from experimental data or empirical correlations.

So far, the proposed stategies proceed hence in two separate stages. In a first step, the inflow condition is built from numerical or experimental data. Then the simulation of the downstream development of the flow is conducted from this information. As a consequence, this second step highly depends on the first one. Noisy or incomplete inlet conditions (with no or erroneous small scales information) may lead to numerical catastrophe or to unrealistic over-smoothed velocity fields. To alleviate such problematic issue, an atractive solution consists in gradually modifying the inflow condition in such a way to enforce the simulation to remain the closest as possible to the data with respect to a given criterion. This is in essence an optimal control strategy to learn the inlet conditions from measurements.

With this state of mind, we propose here to explore an optimal control approach -referred in the litterature as variational data assimilation (VDA)- allowing generating simultaneous transitional initial and inflow boundary conditions and reproducing the spatiotemporal dynamics of an experimentally observed flow. VDA (Le-Dimet and Talagrand, 1986) is a technique derived from optimal control theory (Lions, 1971). It is expressed as the minimization with respect to a control variable of an objective function that measures a discrepancy between a state variable and noisy measurements, subject to a constraint given by the state variable dynamics. The control variable may be for instance a parameter of the dynamics or the initial condition. Assuming that both the model and the objective function are differentiable, VDA proposes to solve this inverse problem looking for a control that cancels out the gradient of this cost function through the use of adjoint minimization techniques. Such techniques enable to compute the functional gradient by means of the adjoint of the tangent linear dynamics. The tangent linear dynamics and its adjoint are provided by automatic differentiation (AD) tools. In the present study, the gradient descent minimization is coupled with a limited memory BFGS deterministic gradient based optimization algorithm (Liu and Nocedal, 1989).

Coupled with this dynamics we consider available noisy measurements of the velocity at discrete instants separated by a given latency (much larger than the DNS time step). By modifying the initial and inflow condition of the system, the proposed method provides the state of an unknown function on the basis of a DNS model and noisy measurements.

It should be emphasized that such an experiment/simulation association can be considered as a procedure for generating realistic inflow conditions for a numerical simulation, but also as a procedure for a dynamical data postprocessing, where the DNS is used to improve experimental data by restoring the part of the information that has been missed or deteriorated during the measurement step.

It is important to outline, that this study aims at constituing only a first proof of concept of the methodology capabilities. We will hence consider only the case of 2D flows. The extension to 3D though computationally much more intensive could be devised on the same basis.

2. Data Assimilation

Variational data assimilation allows estimating over time state variables trajectory of a system of interest. It can be seen as a procedure in which noisy and eventually incomplete data are filtered out by a dynamical system with hidden parameters. This framework allows to handle in a natural way high-dimensional state spaces and is thus intensively used in environmental sciences (Le-Dimet and Talagrand, 1986) for the analysis of atmospheric or oceanic flows in view of their forecast. More precisely, the problem we are dealing with consists in recovering a system's state $X(\mathbf{x}, t)$ obeying a dynamical law, given some noisy and possibly incomplete measurements $\mathcal{Y}$ of the state. The measurements, in this context also called observations, are assumed to be available only at discrete points in time t^* separated by a given latency Δt_{obs}. This is formalized, for any location, $\mathbf{x}$, at time $t \in [t_0, t_f]$, by the system

$$\partial_t X(\mathbf{x}, t) + \mathbb{M}(X(\mathbf{x}, t), \eta(t)) = 0 \tag{1}$$
$$X(\mathbf{x}, t_0) = X_0(\mathbf{x}) + \epsilon(\mathbf{x}), \tag{2}$$

where $\mathbb{M}$ is a nonlinear dynamical operator depending on a control parameter η. The term X_0 is the initial vector at time t_0, and ϵ is an (unknown) additive control variable on the initial condition.

2.1. Dynamical Model

In this study the dynamics of interest consists of the *pressure-velocity* formulation of *Navier-Stokes* equations. Those equations and the numerical implementation we consider in this study are briefly described hereafter.

4

2.1.1. Governing Equations

Mass and momentum conservation principles are represented by the Navier-Stokes equations, which have the following form for an incompressible fluid

$$\nabla \cdot \mathbf{u} \;=\; 0 \tag{3}$$

$$\frac{\partial \mathbf{u}}{\partial t} \;=\; -\nabla p - \omega \times \mathbf{u} + \nu \Delta \mathbf{u}, \tag{4}$$

where ν is the kinematic viscosity, $p(\mathbf{x}, t)$ the dynamic pressure field ($P + \frac{1}{2}\rho|\mathbf{u}|^2$), $\mathbf{u}(\mathbf{x}, t)$ the velocity field, and $\omega(\mathbf{x}, t)$ the vorticity field ($\nabla \times \mathbf{u}$).

2.1.2. Numerical Method

In this study we will rely on the numerical code *Incompact3d*, based on sixth-order compact finite difference schemes and a Cartesian grid to solve the incompressible Navier-Stokes equations (Laizet and Li, 2011). The incompressibility condition is ensured via a fractional step method introducing a Poisson equation for the pressure. An original characteristic of *Incompact3d* is that this equation is directly solved in the framework of the modified spectral formalism. More precisely, the Poisson solver is only based on Fast Fourier Transforms (FFT) despite the use of inflow/outflow boundary conditions (Laizet and Lamballais, 2009). The time advancement is performed using a second-order Adams-Bashforth scheme. Free-slip boundary conditions are applied at $y = \pm L_y/2$, whereas outflow boundary conditions at $x = L_x$ are determined through the resolution of a simplified convection equation

$$\frac{\partial \mathbf{u}}{\partial t} + U_{conv} \frac{\partial \mathbf{u}}{\partial x} = 0, \tag{5}$$

where U_{conv} is a mean convection velocity of the main structures in the outflow region calculated at each time step.

2.2. Cost Functional

Basically, data assimilation is formulated as a goodness of fit problem under a particular additional constraint: the state dynamics (1-2), which expresses the dependence of the system's state variable $X \equiv \mathbf{u}(\gamma)$ on the control variable $\gamma = \{\epsilon(\mathbf{x}), \eta(t)\} \equiv \{\mathbf{u}(\mathbf{x}, t_0) - \mathbf{u}_0(\mathbf{x}), \mathbf{u}(\mathbf{x}_{in}, t) - \mathbf{u}_{in}(t)\}$. These control variables correspond to deviations between, on the one hand, the initial condition $\mathbf{u}(\mathbf{x}, t_0)$ and a reference initial state $\mathbf{u}_0$ and, on the other

hand, between the flow in the inlet section, $\mathbf{x}_{in}$, and a reference time varying inlet condition $\mathbf{u}_{in}$. The initial and inlet references will be further precised later on. The optimal control objective consists then to find a control (or error) of lower magnitude (with respect to an appropriate norm) that leads to the lowest discrepancy between the measurements and the flow velocity. Formally, this is expressed as the minimization of the following cost function

$$
\begin{aligned}
J(\gamma) \;=\; & \int_{t_1}^{t_N} \int_{\Omega_A} \|\overline{\omega}(\mathbf{x}, t) - \omega^{obs}(\mathbf{x}, t)\|_R^2 \, d\mathbf{x} \, \Delta_{t^*} dt \\
& + \int_{\Omega_{C_{ob}}} \|\overline{\mathbf{u}}(\mathbf{x}, t_0) - \mathbf{u}_b(\mathbf{x}, t_0)\|_{Q_{ob}}^2 \, d\mathbf{x} \\
& + \int_{t_0}^{t_N} \int_{\Omega_{C_{in}}} \|\mathbf{u}(\mathbf{x}_{in}, t) - \overline{\mathbf{u}}(\mathbf{x}_{in}, t)\|_{Q_{in}}^2 \, d\mathbf{x} \, dt, \qquad (6)
\end{aligned}
$$

where $t_1 \in [t_0, t_N]$ denotes the initial time of the assimilation window, $\overline{\omega}$ indicates a spatial average of the vorticity field generated from the model output, ω^{obs} stands for the vorticity field associated to the PIV observations, $\mathbf{u}_b(\mathbf{x}, t_0)$ designates the observed initial velocity field (referred as the background state) and $\overline{\gamma} = \{\overline{\mathbf{u}}(\mathbf{x}, t_0), \overline{\mathbf{u}}(\mathbf{x}_{in}, t)\}$ is a smoothed version of the control variable. Here, $\overline{\mathbf{u}}(\mathbf{x}, t_0)$ denotes a spatial average of the initial condition whereas $\overline{\mathbf{u}}(\mathbf{x}_{in}, t)$ corresponds to a temporal average of the inflow condition. The times t^* denote the set of measurements instants and Δ_{t^*} represents a Dirac comb: $\Delta_{t^*} = \sum_k \delta(t - k\Delta t_{obs})$, indicating formally that observation are available with a periodicity Δt_{obs}. The norms $\| \cdot \|_R$, $\| \cdot \|_{Q_{ob}}$ and $\| \cdot \|_{Q_{in}}$ are induced norms of the inner products $< R^{-1} \cdot, \cdot >$, $< Q_{ob}^{-1} \cdot, \cdot >$ and $< Q_{in}^{-1} \cdot, \cdot >$; R, Q_{ob} and Q_{in} are covariance matrices of the assimilation space (spatial domain Ω_A), initial control space (spatial domain $\Omega_{C_{ob}}$) and inlet control space (spatial domain $\Omega_{C_{in}}$). Few remarks can be done here. First of all, we choose in this study to express the data discrepancy with respect to vorticity in order to strengthen the noise measurement. Concerning this, previous experimental data assimilation tests were performed by using a control law based on flow measurements relying on velocity, showing that this strategy, in spite of its straightforward implementation, has the following disadvantages:

(i) It increases dramatically the assimilation domain size to take into account the velocity components and as such, we reduce the performance of the minimization algorithm regarding its convergence properties, i.e. by enhancing

the interval of uncertainty for a minimizer of the functional.

(ii) It introduces ambiguity in the criterion used to identify the location of the spatial assimilation errors, due to strong differences observed between both velocity components.

Otherwise, for numerical stability reasons, the spatial resolution (dx) used in our simulation code is generally much higher than that of the observations (dx_g) supplied by large scale Particle Image Velocimetry (PIV) measurements. The spatial averaging filter applied to the DNS vorticity field allows comparing the flow velocity with the coarse grid vorticity field associated to the observations. The spatial averaging is here defined through a discrete Gaussian filter with support: $\Lambda = \{(x, y) \in \Omega_C \mid (x^2 + y^2) \leq (dx_g/dx)^2\}$. This filtering is reminiscent of the usual weighting function defined over the span of the interrogation window in conventional correlation PIV implementations.

The observation model relating the measurements and the state variables can be formally written as:

$$\underbrace{g_{\sigma_b} * \nabla \times \mathbf{u}(\mathbf{x}, t^*)}_{\mathbb{H}} = \omega^{obs}(\mathbf{x}, t^*) + \zeta, \tag{7}$$

where g_{σ_b} is a Gaussian kernel with $3\sigma_b = dx_g/dx$. Here the operator '$*$' denotes the convolution product and the variable ζ is a Gaussian noise.

Previous assimilation experiments show that as a consequence of neglecting the third term of the functional, the optimization process generates peak shape patterns on velocity profiles of the inflow estimated condition. Those peaks occur at instants t^* corresponding to the observations, revealing that the cause of this pattern is based on the response of the optimization algorithm to the first term of the functional, which only takes into account the model solution $\bar{\mathbf{u}}(\mathbf{x}, t^*)$ evaluated at discrete points in time t^* but ignores what happens during the intermediate temporal states. In order to reduce the magnitude of those peaks, the strategy involved in the third term of the functional consists in comparing the current inflow condition, $\mathbf{u}(\mathbf{x}_{in}, t)$, with a weighted temporal average $\bar{\mathbf{u}}(\mathbf{x}_{in}, t)$ obtained by means of a Gaussian function over the interval $[t^* - \Delta t_{obs}, t^* + \Delta t_{obs}]$. This averaging time is chosen to preserve the global accuracy of temporal discretization and to affect only a time interval between the nearest observations of the flow. We consider that the present smoothing procedure applied to the expected value of the inflow condition is very soft in the sense that when a coarse temporal resolution

7

is used, no significant reduction of spurious patterns can be obtained. In contrast, when the inflow solution is directly smoothed at every iteration, we have observed that a strong reduction of those peaks is often possible, even at very low resolution, but with a simultaneous artificial smoothing of the overall flow not fully compatible with the purpose of a VDA.

This approach allows the inflow condition to change gradually over the whole sequence time range $[t_0, t_f]$. The role of the third term of the functional is thus to enforce a temporal continuous trajectory of the solution but also to impose a base flow corresponding to the flow harmonic component absent from the vorticity observation. The control variables are assumed to be related to a filtered version of the velocity component up to a Gaussian noise. This is formalized through relations:

$$\mathbf{u}(\mathbf{x}, t_0) = g_{\sigma_b}(\cdot, t_0) * \mathbf{u}_b(\mathbf{x}, t_0) + \epsilon \qquad \forall \mathbf{x} \in \Omega_{C_{ob}} \quad (8)$$

$$\mathbf{u}(\mathbf{x}_{in}, t) = g_{\sigma_{in}}(\mathbf{x}_{in}, \cdot) * \mathbf{u}(\mathbf{x}_{in}, t) + \eta \qquad \forall \mathbf{x} \in \Omega_{C_{in}}, \forall t \in [t_0, t_N], \quad (9)$$

where ϵ and η are Gaussian variables encoding respectively the noise on the initial condition and on the inlet condition. The value of the considered standard deviation for the inlet condition have been set in practice in order to allow a smoothing on n consecutive image frames ($n\sigma_{in} = \Delta t_{obs}/dt$).

2.3. Adjoint Model

Regarding the minimization of the objective function, a direct numerical evaluation of the functional gradient is computationally infeasible, because this would require to compute perturbations of the state variables along all the components of the control variables ($\delta\epsilon, \delta\eta$) - i.e. to integrate the dynamical model for all perturbed components of the control variables, which is obviously not possible in practice. As described in Appendix A a solution to this problem consists in relying on an adjoint formulation (Le-Dimet and Talagrand, 1986). Within this formalism, the gradient functional is obtained by a forward integration of the dynamical system (1-2) followed by a backward integration of an adjoint variable, λ, which is driven by a dynamics defined from the adjoint of the tangent linear dynamical operator, $\partial_X \mathbb{M}$, and the tangent linear observation operator, $\partial_X \mathbb{H}$. This reverse dynamics, referred as the adjoint dynamics, is defined as:

$$- \partial_t \lambda(\mathbf{x}, t) + (\partial_X \mathbb{M})^* \lambda(\mathbf{x}, t) = (\partial_X \mathbb{H})^* R^{-1}(\mathcal{Y} - \mathbb{H}(X(\mathbf{x}, t))) \quad (10)$$

$$\lambda(\mathbf{x}, t_f) = 0, \quad (11)$$

allows expressing the functional gradient with respect to the control variables. This gradient whose expressions are derived in Appendix A reads:

$$\frac{\partial \mathcal{J}}{\partial \epsilon} = -\lambda(t_0) + I_c^{-1}(X(t_0) - \mathbb{C}_b X_b),$$
$$\frac{\partial \mathcal{J}}{\partial u} = (\partial_u \mathbb{C}_u)^* F^{-1}(\mathbb{C}_u u - u_0) + (\partial_u \mathbb{M})^* \lambda. \tag{12}$$

These cost functional derivatives, involve the linear tangent expression of several operators. Although only their numerical expression will be needed, we detail further their analytic expression in the following. As can be noted from (7) the observation operator $\mathbb{H}$ is linear. Its linear tangent expression is itself, so we have in 2D (as the smoothing filter is symmetric):

$$\left\langle \omega^{obs}, g * \nabla \times \mathbf{u} \right\rangle = \left\langle g_\sigma * \omega^{obs}, \partial_x v - \partial_y u \right\rangle$$
$$= \left\langle \nabla^\perp g_\sigma * \omega^{obs}, \mathbf{u} \right\rangle, \tag{13}$$

with the orthogonal gradient defined as $\nabla^\perp = (\partial_y, -\partial_x)$. We have hence immediately:

$$(\partial_X \mathbb{H})^* = g_\sigma * \nabla^\perp. \tag{14}$$

Let us also remark that in 3D, this expression would further simplifies to $\mathbb{H}$ as the curl operator is auto-adjoint.

As for the control variables, they involve general linear relations (8-9) of the form

$$\mathbf{u}(\mathbf{x}, t_0) = g_\sigma * \mathbf{u}_b(\mathbf{x}, t_0) + \boldsymbol{\epsilon} \tag{15}$$
$$\mathbb{C}_u \mathbf{u}(\mathbf{x}_{in}, t_0) = \boldsymbol{\eta}, \tag{16}$$

with $\mathbb{C}_u = (\delta - g_\sigma)^*$ and where δ is the Dirac mass and g_σ a spatial or temporal filtering. The adjoint linear tangent operator of $\mathbb{C}_u$ is hence itself.

Concerning the dynamics, the exact adjoint of the discrete scheme associated to the dynamical system is needed for an accurate implementation of the adjoint dynamics. It is thus necessary to construct a numerical procedure that corresponds to the adjoint of the tangent linear expression of the discrete scheme used to implement the dynamics. To this end, we will rely on an automatic differentiation tool, called *Tapenade* (Hascoet, 2003). This systematic approach leads however to massive use of storage, requiring code transformation by hand to reduce memory usage as explained in the next section.

2.4. Building Adjoint Algorithms through AD

Automatic Differentiation (AD) is a technique to evaluate derivatives of a function $F : X \in \mathbb{R}^m \mapsto Y \in \mathbb{R}^n$ defined by a computer program P. In AD, the original program is automatically transformed or extended to a new program P' that computes the derivatives analytically (Griewank, 2000). It can be used to build a program encoding the tangent linear numerical expression of the discrete implementation of a given operator or its adjoint. Such automatic derivation, guaranties thus to compute the exact numerical adjoint. In our case, we set the input $X \equiv \gamma$ and function F has a single real output (the cost).

2.4.1. Application of AD

An AD tool uses the source of the program that computes the state dynamics (1-2), and identifies this program with a composition of mathematical functions, one per run-time instruction. Denoting $\{I_k\}_{k=1\to p}$ the sequence of instructions executed at run-time, each of them implementing an elementary function f_k, the function F computed by P is:

$$F = f_p \circ f_{p-1} \circ \cdots \circ f_2 \circ f_1. \tag{17}$$

Setting $W_0 = X$ and $W_k = f_k(W_{k-1})$, the chain rule gives us the Jacobian of F:

$$F'(X) = f'_p(W_{p-1}) \cdot f'_{p-1}(W_{p-2}) \cdot \cdots \cdot f'_1(W_0). \tag{18}$$

The so-called *adjoint* (or *reverse*) mode of AD aims at computing the product of the transposition of F' by a given weight vector $\overline{Y}$ to get:

$$\overline{X} = F'^t(X) \cdot \overline{Y} = f_1'^t(W_0) \cdot f_2'^t(W_1) \cdot \cdots \cdot f_p'^t(W_{p-1}) \cdot \overline{Y}. \tag{19}$$

In this way, the adjoint mode of AD builds a new code $\overline{\text{P}}$ that computes $\{\overline{W}_{k-1} = f_k'^t(W_{k-1}) \cdot \overline{W}_k\}_{k=p\to 1}$, by using values from P in the reverse of their computation order. The stack needed to store W_k is the bottleneck of reverse AD. In order to keep it small enough, we applied a storage/recomputation strategy described as follows:
(i) During the computation of the cost function J, we store in memory only

10

W_k at instants t^*.

(ii) During the computation of the adjoint variable $\overline{X} \equiv \lambda$, we restart the program on snapshot W_k until W_{k+1} in the forward sweep of P, storing in memory all the intermediate values W_{itime} at each DNS time step. Later, in the backward sweep, each W_{itime} is restored from the stack to be used by $\overline{P}$.

In our case, by setting the weight vector $\overline{Y} \equiv Y$, the code $\overline{P}$ enables to get the gradient functional $\overline{X} = F'^t(X) \cdot Y \equiv \nabla Y$. Concerning this issue, one can think of $\overline{Y}$ as a weighting vector on Y, the results of a function $F : X \in \mathbb{R}^m \mapsto Y \in \mathbb{R}^n$, that defines a scalar composite result, of which we compute the gradient. Alternatively, given a program P that discretizes and computes the function F, AD in the reverse mode creates a new program $\overline{P}$ that computes the transposed Jacobian of F multiplied by a given vector. In our optimization context, F has a single output (the cost) and therefore the program $\overline{P}$ computes exactly the gradient of F by considering $\overline{Y} \equiv Y$. Thus, the reverse mode of AD takes as input a single vector $\overline{Y}$ that defines the composite optimization criterion for which the gradient must be computed.

2.4.2. Validation checks

The usual process to validate the AD generated codes consists in validating the tangent derivatives with respect to finite differences, and to validate the reverse derivatives with respect to the tangent derivatives using the *dot-product* test. More precisely, choosing an arbitrary state X (initial and inflow condition in our case) and an arbitrary direction $\dot{X}$, we compute the finite difference $FD = [F(X+\epsilon\dot{X})-F(X)]/\epsilon$. Using the tangent differentiated program, we compute $\dot{Y} = F'(X) \cdot \dot{X}$. Using the adjoint differentiated program, we compute $\overline{X} = F'^t(X) \cdot \dot{Y}$. The dot-product test just checks that

$$\langle \overline{X}, \dot{X} \rangle = \langle F'^t(X) \cdot \dot{Y}, \dot{X} \rangle = \langle \dot{Y}, F'(X) \cdot \dot{X} \rangle = \langle \dot{Y}, \dot{Y} \rangle, \qquad (20)$$

up to an admissible error.

2.5. Optimization with Gradient Descent

In our optimization problem, Hessian matrix of the cost function is too dense if we take into account the large number of variables involved, i.e. $(2 \cdot \mathtt{n_x} \cdot \mathtt{n_y} + 2 \cdot \mathtt{n_y} \cdot \Delta\mathtt{itime})$, where $\mathtt{n_x}$, $\mathtt{n_y}$ is the number of grid points in control space domain Ω_C, and $\Delta\mathtt{itime}$ the number of DNS time steps in $[t_0, t_f]$. So, in order to reduce the cost of storing and manipulating it, we have chosen a limited-memory quasi-Newton method which maintains simple

and compact approximations of Hessian matrices (Liu and Nocedal, 1989). We have used an algorithm known as L-BFGS, which is based on the BFGS updating formula

$$\gamma_{k+1} = \gamma_k + \alpha_k p_k, \tag{21}$$

where α_k is the step length, $p_k = -H_k \nabla Y_k$ is the search direction, and H_k denotes a first order approximation of the Hessian in the direction of the previous increment. This approximation is updated at every iteration by storing the vector pairs

$$s_k = \gamma_{k+1} - \gamma_k, \qquad y_k = \nabla Y_{k+1} - \nabla Y_k. \tag{22}$$

The step length is computed from a line search procedure to satisfy the Wolfe conditions.
We have chosen a stop criterion for the L-BFGS algorithm based on gradient reduction from its initial value, i.e. $\|\nabla F(X_k)\|/\|\nabla F(X_0)\| < 1 \cdot 10^{-5}$.

3. Results

First, the correctness and the computational cost of the AD is checked. Subsequently, the values we chose for the different parameters of the method are described. Then, validations of the control on initial and inflow conditions are carried out using three different setups: a synthetic image setup based on a mixing layer and synthetic/experimental setups based on a circular cylinder wake.

3.1. Correctness of AD codes

In order to assess the validity of the differentiated codes, we performed validation experiments (dot-product test) by running the generated tangent (forward mode AD) and adjoint (reverse mode AD) codes many times, with the same value of X (independent state introduced by the initial and inflow condition) but with different values of $\dot{X}$ (direction in the input space along which the derivatives are computed), where the directional variables are assumed to be composed by the application of a Gaussian noise (with zero mean and unit variance) to the independent ones. This can be formally written as:

$$\dot{\mathbf{u}}(\mathbf{x}, t_0) = \mathbf{u}(\mathbf{x}, t_0) + \tau \cdot \epsilon \qquad \forall \mathbf{x} \in \Omega_{C_{ob}} \tag{23}$$

$$\dot{\mathbf{u}}(\mathbf{x}_{in}, t) = \mathbf{u}(\mathbf{x}_{in}, t) + \tau \cdot \eta \qquad \forall \mathbf{x} \in \Omega_{C_{in}}, \forall t \in [t_0, t_N], \tag{24}$$

12

τ	0.01	0.1
$\langle \dot{FD}, \dot{FD} \rangle$	5.94345959284251984E-006	6.05599345995614287E-006
$\langle \dot{Y}, \dot{Y} \rangle$	5.94345528277551301E-006	6.05598865931465216E-006
$\langle \overline{X}, \dot{X} \rangle$	5.94345528277551301E-006	6.05598865931465470E-006
τ	1.0	10.0
$\langle \dot{FD}, \dot{FD} \rangle$	7.23940570717684538E-006	2.48837235816609840E-005
$\langle \dot{Y}, \dot{Y} \rangle$	7.23936862789542849E-006	2.48777886326398566E-005
$\langle \overline{X}, \dot{X} \rangle$	7.23936862789542849E-006	2.48777886326398194E-005

Table 1: Validity test of the AD technique as the noise on the directional variables is increased ($\epsilon = 10^{-7}$).

where ϵ and η are gaussian variable encoding respectively the noise on the initial condition and on the inlet condition, and τ is a scalar noise amplitude parameter. Results shown in Table 1 for each arbitrary direction obtained by varying τ with random draws of (ϵ, η), indicate that the tangent norm and the adjoint norm match very well, up to the last few digits. This shows that the tangent and adjoint codes really compute the same derivatives independently of the arbitrary direction of choice. The norm obtained with Finite Differences (FD) matches only to half the machine precision, because of the inaccuracy of FD approximation. This validity test enables us to ascertain that the program doing $F'^{t}(X)$ is indeed adjoint to the program doing $F(X)$, and thus the proposed code for the adjoint operator is precisely consistent with the operator itself.

3.2. Computational cost

To assess the computational cost of the method, the AD generated code was run on a desktop machine with a 3.40 GHz Intel Core i7-2600 CPU with 8 GB of RAM. Performance results are shown in Table 2 corresponding to VDA Test A for a DNS mesh size of 109×339 grid points in control space domain. Table 2 shows that the forward and backward implementations are cost competitive. Further, this gives a ratio of 2.7 between the run-times of the gradient and function evaluations, indicating that the overall cost of the adjoint method is reasonable. The generated code required 33.5 MB of tape size and 15 hours to reach the assimilated trajectory, showing that:
(i) The proposed storage/recomputation strategy, based on storing an appropiate number of intermediate states during the forward path of differentiation and then recalling them in the reverse computation, is effective in

terms of memory consumption.

(ii) While this is not a trivial computational cost for a 2D test case, it does bring the optimization of unsteady problems into the realm of possibility.

Computation of the cost function	1.08
Computation of the adjoint variable	2.98
Forward sweep	1.38
Backward sweep	1.31

Table 2: Computational cost of the AD generated code. Times are normalized to the total computational cost of the flow solution (i.e. the time needed to complete a trajectory of the dynamic model), which takes 91.3 s in the case of VDA cylinder wake experimental test A.

3.3. Parameter estimation

The only parameters of the method are constituted by the covariance matrices associated to the observations R, the initialization Q_{ob} and the dynamical model Q_{in}. For the observation, we systematically imposed $R=1$. The initialization covariance matrices are defined with respect to the initialization model. In the case of synthetic model (given through velocity fields provided by a combination of DNS and filtering techniques), we fixed the covariance matrix to $Q_{ob}=10$. When initializing from the first PIV field of the sequence, the value $Q_{ob}=100$ is introduced. Concerning the dynamical model covariance matrix, we fixed it to $Q_{in}=1$ for the synthetic sequences, as in this case the dynamic is quasi respected and to $Q_{in}=10$ for real world applications with larger dynamical uncertainty.

3.4. Control on the Initial Condition

In order to assess the benefits of our technique, we carried out a VDA experiment applied to the identification of the initial condition from a spatially evolving 2D mixing layer flow by using numerical data. The velocity ratio between low and high speed is 0.66 and the Reynolds number based on the velocity difference of two streams and the inflow vorticity thickness $\delta_{\omega 0}$ is $Re = 400$. The governing equations are directly solved using 401x257 grid points in x and y directions respectively. The assimilation domain size was $L_x=200\delta_{\omega 0}$, $L_y=512\delta_{\omega 0}$, and the grid resolution in x direction was $\Delta x=0.5\delta_{\omega 0}$. The stretching of the grid in y direction leads to a minimal mesh size of $\Delta y_{min}=0.15\delta_{\omega 0}$.

As a first step, a precursor simulation was run to provide artificial inflow conditions for the assimilation experiment. The spanwise section retained and fixed subsequently to the inlet section of the assimilation domain is located at the beginning of the convection region of vortical structures. This VDA experiment was done in a so-called twin experiment framework whereby the direct model trajectory is used to generate synthetic observations. The initial velocity field has been then perturbed using zero mean Gaussian noise and an additional spatial filtering is applied to this vector velocity field in order to provide the noisy field at the start of the assimilation process. Synthetic observations are given by a sequence of 20 velocity fields generated from the model's original outputs starting from the unperturbed field.

In this experiment the functional $J(\gamma)$ considered depends only on the initial condition, and comes to an initial value control problem. Figure 1(a) illustrates the performance of the optimization loop for an integration period of $2\delta_{\omega 0}/U_m$ (with a DNS time step of $0.01\delta_{\omega 0}/U_m$) and an observation frequency equal to $0.1\delta_{\omega 0}/U_m$, where U_m is the mean convective velocity at the inlet. After 44 iterations the cost function decreases by 99.8% from $8 \cdot 10^{-3}$ to $1.6 \cdot 10^{-5}$. Figure 1(b) shows the temporal evolution of the assimilation error with respect to the initial unperturbed observation (our reference or true state). The initial trajectory is generated from the model output starting from the perturbed observation at $t = t_0$ (initial estimate for L-BFGS algorithm), and the assimilated one is obtained from the best estimate for the initial state found by L-BFGS.

As can be seen, the minimization algorithm corrects the error of the initial guess and converges to a global minimum point that is close to the true state point. The initial condition of the model is retrieved, yielding a very good agreement with the unperturbed velocity fields. Figure 2 indicates that the true state at the start of the assimilation window (top panel) is recovered with a very good accuracy (bottom panel) from the perturbed one (middle panel), showing the quality of the derivatives obtained.

3.5. Control on the Initial and Inflow Condition

In the previous assimilation experiment, an artificial inlet condition obtained from the reference DNS trajectory was used as a fixed parameter in our optimization system, and then $\gamma = \{\mathbf{u}(\mathbf{x}, t_0)\}$. From now on, in addition to the flow initial condition we will incorporate inflow condition as a supplementary control parameter. In order to make it possible, we need to provide an initial estimate for the inflow condition, i.e. $\mathbf{u}_{k=0}(\mathbf{x}_{in}, t)$ and the initial
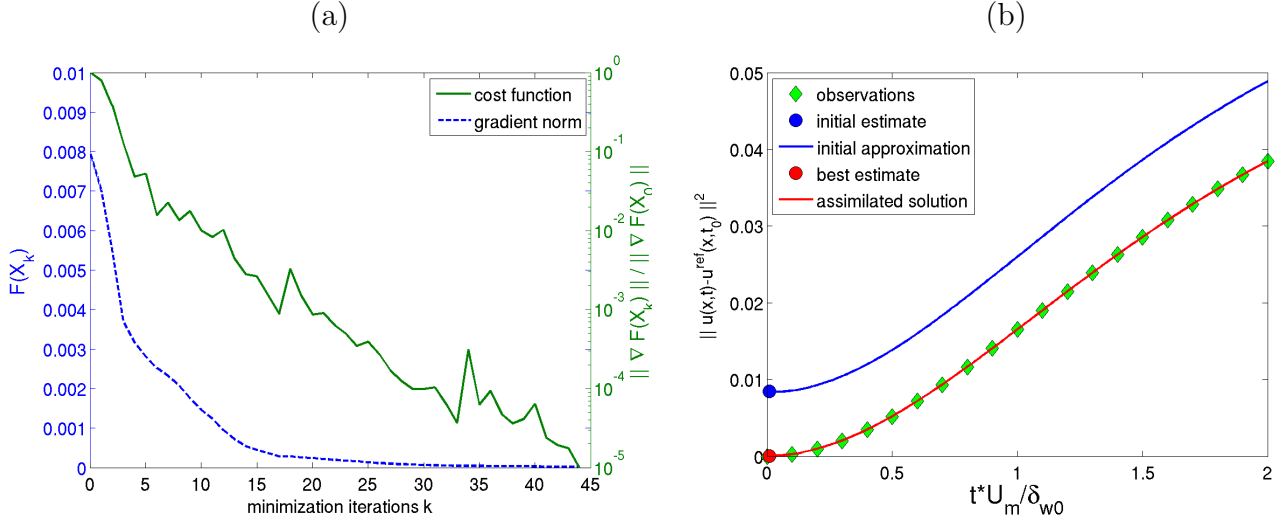
15

Figure 1: (a) Reduction of the cost function and the gradient norm *versus* the number of minimization iterations with the L-BFGS algorithm. (b) Temporal evolution of the squared norm of the discrepancy between each velocity field generated from the model output $\mathbf{u}(\mathbf{x}, t)$ and the reference value $\mathbf{u}^{ref}(\mathbf{x}, t)$.

condition. Similarly to the previous experiment, we initialized the initial condition to the observed data at the initial time:

$$\mathbf{u}_{k=0}(\mathbf{x}, t_0) = \mathbf{u}^{obs}(\mathbf{x}, t_0). \tag{25}$$

A natural choice to provide an initial value of the inflow condition consists in using the complete sequence of observations together with a "frozen" turbulence Taylor's hypothesis. From Taylor's hypothesis, the spatial changes caused by advection between two contiguous observations are set to $\Delta x_{obs} = U_m \Delta t_{obs}$, where U_m is the mean convection velocity at the inlet section. One solution for the initial estimate of the inflow condition emerges immediately as

$$\mathbf{u}_{k=0}(\mathbf{x}_{in}, t_\sigma) = \mathbf{u}^{obs}(\mathbf{x}_\sigma, t^*), \tag{26}$$

where $\mathbf{x}_\sigma = \mathbf{x}_\sigma(\Delta x_{obs}, dx), \quad t_\sigma = t_\sigma(t^*, \Delta t_{obs}, dt)$. Despite this solution is strongly based on the Taylor's assumption of frozen turbulence, we will see
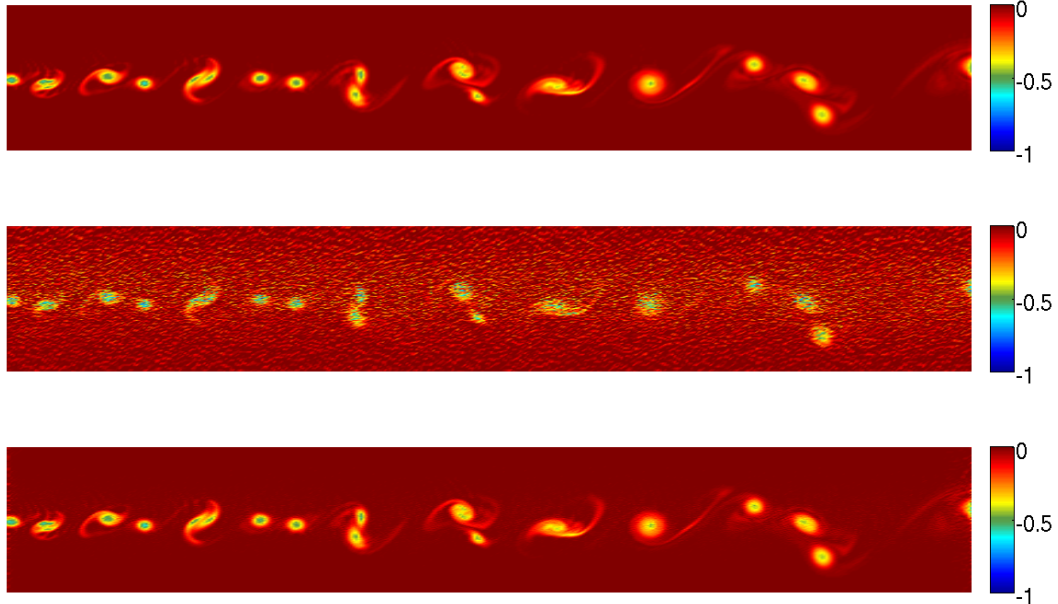
16

Figure 2: Vorticity fields at the start of the assimilation window. True field (top), initial perturbed field (middle) and retrieved optimal initial condition (bottom).

that such hypothesis imposed only on the initial run allows us to face situations for which this hypothesis is not valid. This ability will be nevertheless paid by an increase of the number of iterations to reach convergence.

In order to assess the performance of the VDA method for the specification of inflow condition we have constituted a benchmark composed of DNS results and experimental PIV data. The numerical simulation and the experimental data concerns both a wake behind a circular cylinder at Reynolds 125 and 170 respectively (Reynolds number based on the free stream velocity U_∞, the kinematic viscosity ν and the diameter of the circular cylinder D). For these Reynolds numbers, the transition to turbulence takes place in the wake. This regime is identified with the Bénard-von Karman vortex street and the largest scales remain bidimensional (Zdravkovich, 1997).

3.5.1. Numerical Data Assimilation

The VDA approach is now validated through a twin experiment in which the observation data are built from a numerical simulation of reference, a
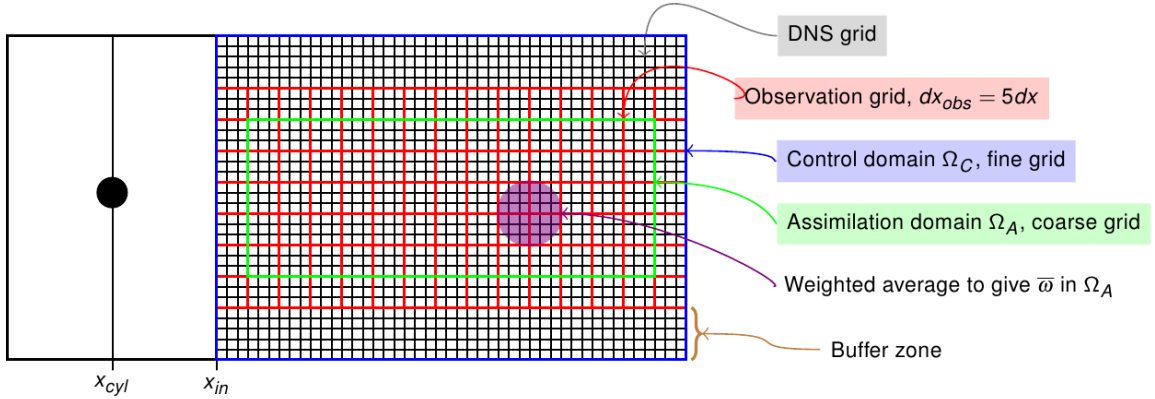
17

Figure 3: Computational flow configuration of VDA cylinder wake tests.

circular cylinder wake at Reynolds 125. Our computational domain size is $L_x \times L_y$ =20$D \times$20D and the corresponding number of points is $n_x \times n_y$ =1801$\times$721. A constant flow is imposed at the entrance of this reference (ground truth state) domain, and the center of the cylinder is located at x_{cyl} =8D downstream of the inflow. The simulation was carried out with a time step Δt =0.012D/U_∞. A sequence of 50 velocity fields with 10Δt time steps between them has been kept to build a discrete sequence of flow motion snapshots. The spatial domain size has been reduced from the computational domain size to 9$D \times$9D. The inlet section of the assimilation domain has been chosen at 22D from the center of the cylinder, i.e. $(x_{in} - x_{cyl})$ =14D, in a way to satisfy Taylor's hypothesis by verifying $\langle u'_y u'_y \rangle_{max}/U_m$ $\approx$0.1 at the inlet. To mimic a typical experimental situation, the spatial resolution has been reduced by a factor 5, yielding 65$\times$64 points for the assimilation domain in the streamwise and normal directions. In Figure 3 is depicted the geometry of the two-dimensional problem considered. In order to properly adjust the observation data to the numerical grid, a buffer zone has been created to verify the specific lateral boundary conditions required by DNS by means of a ramp interpolation function introduced only at the beginning of each optimization iteration.

Figure 4 presents the observation and model dynamics trajectories at two different stages of the assimilation process. The initial trajectory is generated from the model output by starting from the spatially interpolated observation
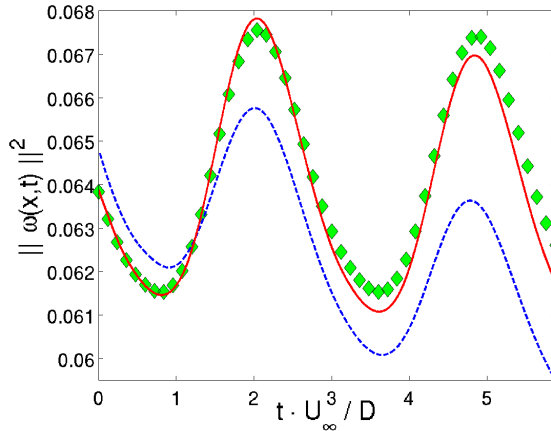
18

Figure 4: VDA cylinder wake twin test. Temporal evolution of the squared norm of the vorticity field; diamonds: observations; dashed line: initial approximation; solid line: assimilated solution.

at $t = t_0$, and using a temporally smoothed inflow condition constructed from the observations by applying Taylor's hypothesis.

The assimilated trajectory is obtained from the best estimate for the initial and inflow condition found by L-BFGS algorithm. As it can be observed from these results, this trajectory fits almost perfectly the observation data. The inflow control variable enables to complete the missing elements of the initial DNS dynamics trajectory to explain the data. The good results obtained in this case were expected since the approach assimilate enough observations to describe the time evolution of physical phenomenon. Indeed, Δt_{obs} was equal to $10\Delta t$ thus providing more than 25 observations during one vortex shedding. However in real world applications the constitution of a noise-free velocity field PIV sequence at high temporal resolution is completely unrealistic. We show in the following section results obtained with real world PIV measurements.

3.5.2. Experimental Data Assimilation

Time-resolved 2D PIV measurements, in the wake of circular cylinder at Reynolds 170, were carried out in one of the wind tunnels at Rennes research centre of Irstea. A sequence of 10 velocity fields with a temporal resolution of $\Delta t_{obs} = 1.2 D/U_\infty$ and with a temporal length of approximately 4 vortex shedding has been kept for building the snapshots sequence of the
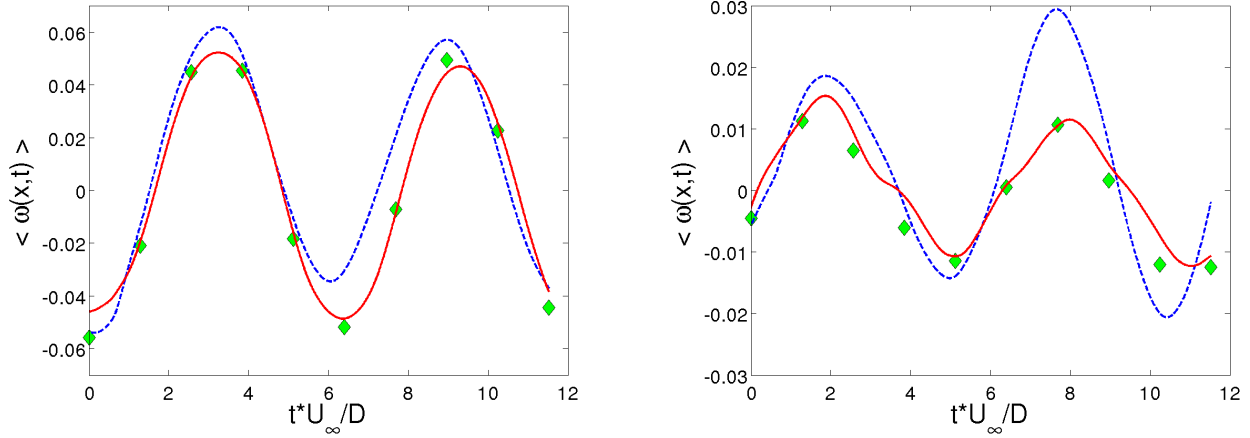
19

Figure 5: Temporal evolution of the spatial mean of the vorticity field for (left) VDA cylinder wake experimental test A and (right) test B; diamonds: observations; dashed line: initial approximation; solid line: assimilated solution.

flow motion. Two assimilation tests were performed by changing the inlet location of the assimilation domain. During these tests the simulations were carried out with a time step equal to $\Delta t = 0.01 D/U_\infty$. Table 3 summarizes the main spatial domain characteristics of these velocity fields. Figure 5 presents the observation and DNS trajectories for both VDA tests at two different stages of the optimization cycle.

Test	$(x_{in} - x_{cyl})$	$(L_x \times L_y)_{\Omega_C}$	$(n_x \times n_y)_{\Omega_C}$	$(L_x \times L_y)_{\Omega_A}$	$(n_x \times n_y)_{\Omega_A}$
A	$10D$	$7.7D \times 24.2D$	109×339	$6D \times 8D$	29×39
B	$3D$	$14.2D \times 24.2D$	199×339	$12D \times 8D$	59×39

Table 3: VDA cylinder wake experimental tests from PIV velocity fields. Characteristics of the spatial domain. Ω_C denotes the control space and Ω_A the assimilation space.

As it can be observed from these curves, the assimilation technique enables to modify inflow condition to recover with quite a good accuracy the trajectory corresponding to the observations. When the inlet of the assimilation domain is located downstream of the vortex formation region (VDA test A), initial guess stay close to the observation data and the optimization algorithm requires 150 iterations to reach the assimilated trajectory. On the other hand, if the inlet belongs to the vortex formation region (VDA test B), Taylor's
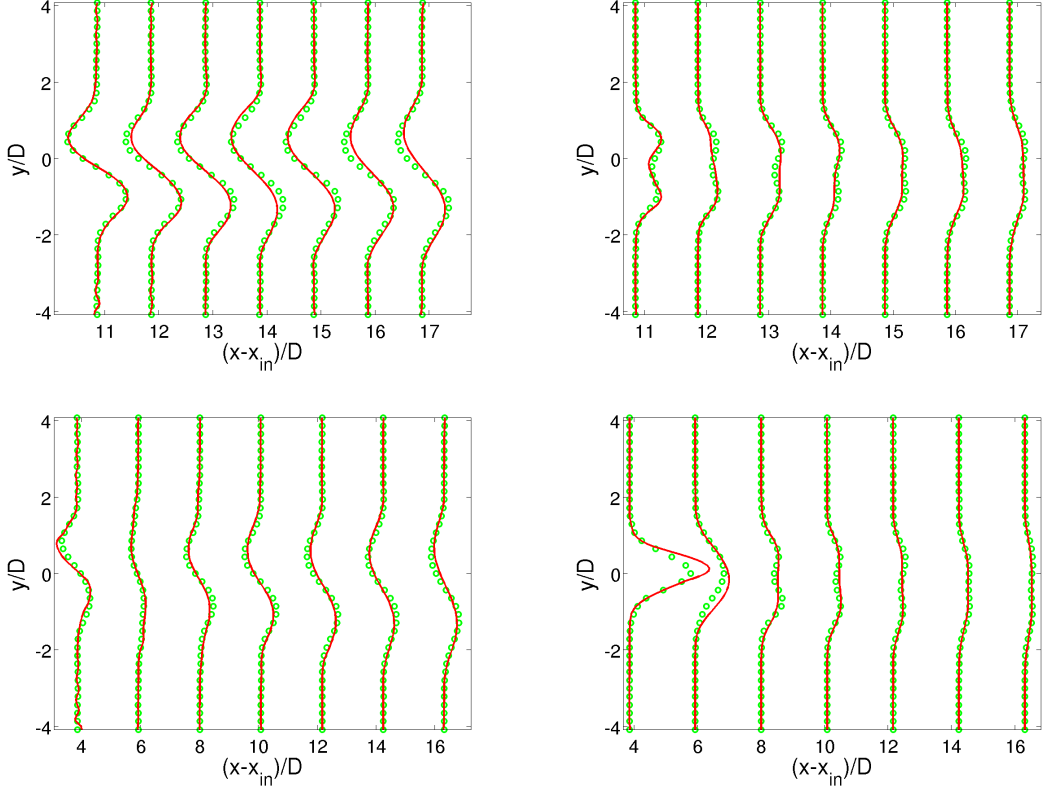
20

Figure 6: Longitudinal evolution of (left) the mean vorticity and (right) the vorticity fluctuation $\langle \omega' \omega' \rangle$ for (top) VDA cylinder wake experimental test A and (bottom) test B; solid lines: assimilated solution; symbols: observations.

hypothesis does not hold anymore (the relative turbulence intensity is high) and the initial approximation is far from the data. This leads to an increase of the number of iterations required (400) to get the assimilated solution. To characterize the assimilation results further and more quantitatively, mean flow characteristics are compared. Figure 6 presents the downstream evolution of the profiles of both the mean and fluctuating vorticity for VDA tests A and B.

It reveals that temporal mean properties of assimilated vorticity are in good agreement with experimental ones. To illustrate the vorticity fields estimated through the assimilation procedure, we plot in Figure 7 two pairs of consecutive snapshots of the vorticity corresponding to the DNS and the PIV

sequence for VDA test B.

Results indicate that despite observations with low spatial and temporal resolutions, the assimilated state exhibits fine scale details revealing vortex filaments. Furthermore, it should be noted that the proposed method provides a means to simulate a wake flow without simulating the flow around the obstacle.

To investigate the spatial structure of the flow simulated by DNS, mean and fluctuating velocity field contours are computed and compared to experimental results in Figures 10 and 11. In spite of the low statistical convergence (only 1152 time steps are retained for the statistics computation in the case of the DNS, which corresponds to the time required for the flow to shed 4 vortices), a good agreement is obtained for velocity field level sets as well as velocity field shapes. Thus, even in the vortex formation region of the assimilation domain, characteristics of the spatial structure of wake flows, such as wake length and location of recirculation, are accurately reproduced by the simulation.

Temporal evolution of the initial (based on the Taylor's assumption) and optimal estimated velocity fields at the inlet section (Figures 8 and 9) exhibits well organized regions both in the longitudinal and vertical components. The noisy character and low temporal resolution of the experimental data leads to high gradients in the generated velocity fields at the inlet section. Taking account of the fact that the covariance parameters involved in the functional (6) ensue from the assumption of an inexact dynamical law together with noisy measurements and inaccurate initial conditions, and considering the large uncertainty in the inital estimate for the inflow condition built from the PIV sequence, one finds that the increase in the model covariance associated to the inlet control space produces a solution closer to the expected value, by giving a smoothed representation of the inflow condition.

4. Conclusions

In this work, a new method for generating inflow boundary conditions for DNS has been introduced. This approach relies on variational data assimilation principles and adjoint-based optimization. By modifying the initial and inflow condition of the system, the proposed method allows us to recover the state of an unknown function on the basis of a DNS model and noisy measurements.

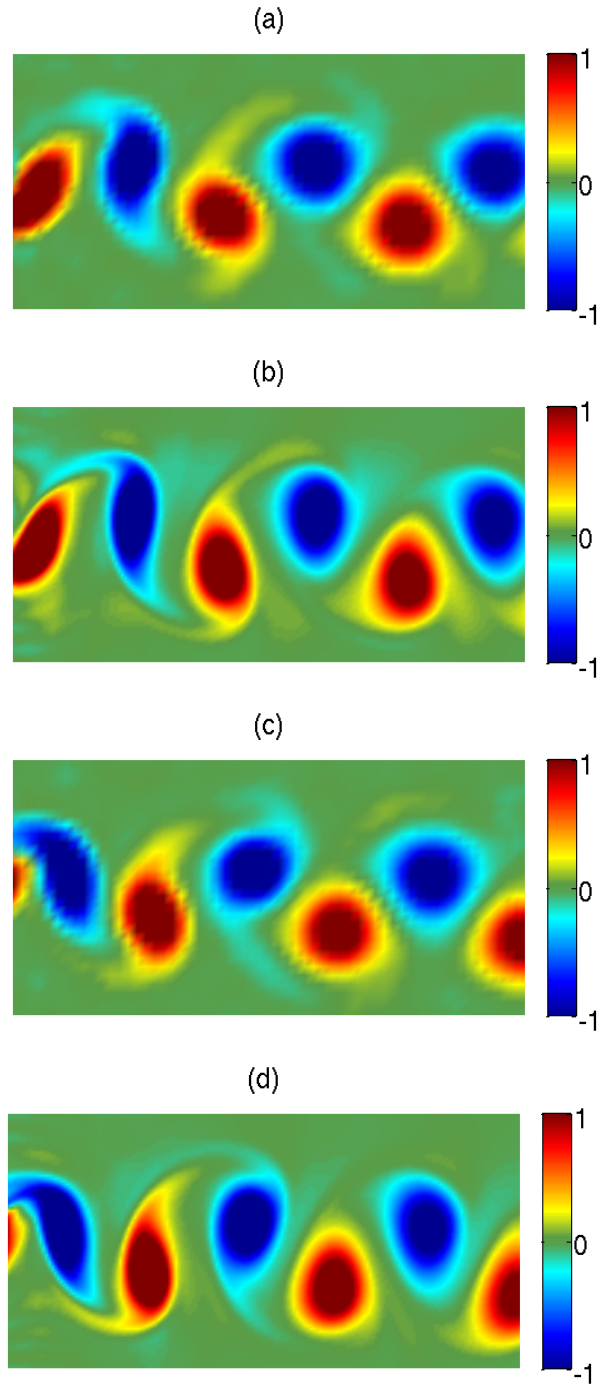In the proposed method, attention has been paid not only to the correct

Figure 7: VDA cylinder wake experimental test B. Vorticity fields at two discrete instants in the assimilation window. (a) Experimental observation at $t \cdot U_\infty/D = 6.4$; (b) Assimilated state at $t \cdot U_\infty/D = 6.4$; (c) Experimental observation at $t \cdot U_\infty/D = 10.2$; (d) Assimilated state at $t \cdot U_\infty/D = 10.2$.

modeling of the spatiotemporal dynamics but also to the proper spatial adjustment of the experimental data to the numerical grid. In particular, a combination of interpolation and domain reconstruction techniques has been employed to deal with the specific lateral boundary conditions required by DNS. We described a number of improvements to reduce the memory needed by reverse-differentiated programs. In order to test this new approach, DNS of a 2D mixing layer flow and a wake flow behind a circular cylinder have been performed to provide a synthetic database. Both twin experiments allowed the validation of the solution methodology in a controlled scenario and demonstrated the feasibility and the reliability of the proposed method. The potential of the assimilation technique was also illustrated in a real world application. For this purpose, the database consisted of a sequence of noisy large scale PIV vorticity measurements. This first proof of concept shows that the approach might be of particular interest to perform computation of highly complex external flows, by considering inlet boundary conditions taken far downstream from the body in order to avoid the expensive computation of the near wall region.

To go further, it could be interesting to introduce dynamical laws related to the observed phenomenon at higher Reynolds numbers; in this sense, an attempt at combining an experimental database to a LES code is now being considered.

Even if free-slip lateral and convective outflow boundary conditions were considered in all the tests in this work, the proposed methodology could be extended to periodic, no-slip or open conditions depending on the flow configuration considered. Periodic lateral boundary conditions can be imposed directly via the spatial differentiation (derivative and interpolation) without specific care in the time advancement. In contrary, the use of Dirichlet conditions on the velocity (for no-slip or open conditions) needs to be defined according to the time advancement procedure. In practice, the explicit nature of the time discretization does not lead to particular problems for the adaptation to the adjoint equations generated by automatic differentiation in reverse mode. Furthermore, considering the outflow condition, the actual configuration using a purely convective flow assumption could likely be well improved by introducing the outflow boundary condition as a control parameter of the optimization problem.

Future works will also include extending the adjoint code capabilities to include three-dimensional effects in data assimilation system. As far as this is concerned, the DNS code (called Incompact3d) selected in this work has

been recently adapted to massive parallel processing (Laizet and Li, 2011). The benefit of computing the adjoint of the new parallel code is that the parallelisation strategy adopted maintains the original structure of the code since no changes are made in the computation of the spatial differentiations and in the Poisson solver. On the contrary, we note two important issues in the AD of message-passing parallel programs. The first issue is preserving the association between variables and their derivative vectors not only during memory allocation and floating-point operations, but also when data are sent via messages. If the variable and its associated derivative object are to be communicated using a single message, they must be packed together. If they are to be communicated via separate messages, an association between these messages must be maintained by using tags. The benefit of using packing is that it is simple to maintain the association, while one disadvantage is the overhead of packing and unpacking the data. Another disadvantage of packing is that it may be necessary to allocate space for the packed data, especially if the packing is explicit. The second issue is the differentiation of parallel reduction operations. A reduction operation is an operation that reduces N values residing on up to N processors to a single value using an associative operator. These reduction operations are elementary operations and as such, we cannot apply AD directly to the reduction operation, but instead must provide a rule for computing the partial derivatives and a mechanism for applying the chain rule.

The tuning and the balance of the corresponding covariance matrices are also intricate issues and we wished in this work to focus explicitly on the methodology.

Appendix A. Variational data assimilation

Variational data assimilation aims at recovering the values of control parameters leading to the lowest discrepancy between the measurements and the system's state variable. This objective can be formalized as the minimization of a cost functional, $\mathcal{J} : U \times \mathcal{V} \to \mathbb{R}$, defined as:

$$\mathcal{J}(u, \epsilon) = \frac{1}{2} \int_{t_0}^{t_f} \|\mathcal{Y}(t) - \mathbb{H}(X(u(t), \epsilon, t))\|_R^2 dt +$$

$$\frac{1}{2}\|\epsilon\|_{I_c}^2 + \frac{1}{2} \int_{t_0}^{t_f} \|\mathbb{C}_u(u(t)) - u_o(t)\|_F^2 dt, \quad \text{(A.1)}$$

where the control variables consists of an unknown pertubation of the initial condition around a known background state X_b:

$$X(x, 0) = \mathbb{C}_b X_b(x) + \epsilon(x),$$

and a dynamics parameter $u(t)$ with an assumed value $u_o(t)$. This functional which can be interpreted as the energy function associated to the a posteriori distribution $p(X|\mathcal{Y})$ in a Bayesian setup gathers three terms. The first term comes directly from the measurement equation:

$$\mathcal{Y}(x, t) = \mathbb{H}X(x, t) + \eta(x), \tag{A.2}$$

with η a zero mean Gaussian random field. It is a quadratic best fit term between the observation and the state variable provided by the dynamics integration:

$$\partial_t X(x, t) + \mathbb{M}(X(x, t), u(t)) = 0 \tag{A.3}$$
$$X(x, t_0) = \mathbb{C}_b X_b(x) + \epsilon(x). \tag{A.4}$$

The second term aims at specifying a low error on the initial condition whereas the third term enforces the control variable to be close to a given *a priori* value u_0 of the control parameter. It is least squares best fit term similar to the observation model. It involves eventualy a nonlinear/linear operator $\mathbb{C}_u$. This operator plays the same role as the background operator $\mathbb{C}_b$ involved in the initial condition equation (A.4). Both of them encodes eventually an incomplete observability situation of the initial condition and the dynamics parameter control variable. The problem consists then to seek deviations of the lowest magnitude both between the *a priori* dynamic parameter value, $\mathbf{u}_0$, and its current value and between the initial state and a given - or observed - initial condition. For a null a priori control value a control of lowest norm is sought. Formally it is assumed that $u(t) \in U$, $X(t) \in \mathcal{V}$ and $\mathcal{Y}(t) \in \mathcal{O}$ are square integrable functions in Hilbert spaces identified to their dual. The norms correspond to the Mahalanobis distance defined from the inner products $< R^{-1}., . >_{\mathcal{O}}$, $< I_c^{-1}., . >_{\mathcal{V}}$ and $< F^{-1}., . >_U$ of the measurements, the state variable and the control variable spaces respectively. They involve covariance tensors R, I_c and F related to the measurement error, the error on the initial condition and the deviation between the control and its *a priori* value. In our applications, these covariance tensors have been defined as diagonal tensors (*i.e.* the noise is assumed to be uncorrelated in time

and space). For example the observation covariance tensor has been set to a covariance tensor of the form:

$$R(x, t, x', t') = \sigma \delta(x - x')\delta(t - t'), \tag{A.5}$$

and similar expressions hold for F and I_c. In order to compute the gradient of this functional we assume that $X(u(t), \epsilon; t)$ depends continuously on $(u(t), \epsilon)$ and is differentiable with respect to the control variables $u(t)$ and ϵ, on the whole time range.

Appendix A.1. Differentiation

Noting first that $dX = (\partial X/\partial u)\delta u(t) + (\partial X/\partial \epsilon)\delta \epsilon$, the differentiation of equations (A.3–A.4) in the direction $(\delta u, \delta \epsilon)$ reads:

$$\partial_t dX + \partial_X \mathbb{M}(X, u(t))dX + \partial_u \mathbb{M}(X, u(t))\delta u(t) = 0, \tag{A.6}$$
$$dX(x, t_0) = \delta \epsilon(x), \tag{A.7}$$

where $\partial_X \mathbb{M}$ denotes the *linear tangent operators* defined by:

$$\lim_{\beta \to 0} \frac{\mathbb{M}(X + \beta X, u(t)) - \mathbb{M}(X, u(t))}{\beta} = \partial_X \mathbb{M}(X)dX. \tag{A.8}$$

We can check immediately that for a linear operator the linear tangent operator is itself. The differentiation of the cost function (A.1) in the direction $(\delta u, \delta \epsilon)$ (denoting U_T as the space of square integrable function on a spatio-temporal domain) reads then:

$$\left\langle \frac{\partial \mathcal{J}}{\partial u}, \delta u \right\rangle_{U_T} = \int_{t_0}^{t_f} \left\langle \mathbb{C}_u u(t) - u_0(t), (\partial_u \mathbb{C}_u)\delta u(t) \right\rangle_F dt -$$
$$\int_{t_0}^{t_f} \left\langle \mathcal{Y}(t) - \mathbb{H}(X(t)), (\partial_X \mathbb{H})\left(\frac{\partial X}{\partial u}\delta u(t)\right) \right\rangle_{\mathcal{O}} dt, \tag{A.9}$$

$$\left\langle \frac{\partial \mathcal{J}}{\partial \epsilon}, \delta \epsilon \right\rangle_{\mathcal{V}} = \left\langle (X(x, t_0) - \mathbb{C}_b X_b(x)), \delta \epsilon \right\rangle_{I_c} -$$
$$\int_{t_0}^{t_f} \left\langle \mathcal{Y}(t) - \mathbb{H}(X(t)), (\partial_X \mathbb{H})\left(\frac{\partial X}{\partial \epsilon}\delta \epsilon\right) \right\rangle_{\mathcal{O}} dt. \tag{A.10}$$

27

Introducing the adjoint of the linear tangent operator $(\partial_X \mathbb{H})^*$, defined as:

$$\forall (x, y) \in (\mathcal{V}, \mathcal{O}), \; < (\partial_X \mathbb{H}) \, x, y >_{\mathcal{O}} = < x, (\partial_X \mathbb{H})^* \, y >_{\mathcal{V}}, \qquad (A.11)$$

and similarly the adjoint $(\partial_u \mathbb{C}_u)^*$, these two relations can be reformulated as:

$$\left\langle \frac{\partial \mathcal{J}}{\partial u}, \delta u \right\rangle_{U_T} = \int_{t_0}^{t_f} \left\langle (\partial_u \mathbb{C}_u)^* \, F^{-1}(\mathbb{C}u(t) - u_0(t)), \delta u(t) \right\rangle_U dt -$$
$$\int_{t_0}^{t_f} \left\langle (\partial_X \mathbb{H})^* \, R^{-1}(\mathcal{Y}(t) - \mathbb{H}(X(t)), \frac{\partial X}{\partial u} \delta u(t) \right\rangle_{\mathcal{V}} dt, \quad (A.12)$$

and

$$\left\langle \frac{\partial \mathcal{J}}{\partial \epsilon}, \delta \epsilon \right\rangle_{\mathcal{V}} = \left\langle I_c^{-1}(X(x, t_0) - \mathbb{C}_b X_b(x)), \delta \epsilon \right\rangle_{\mathcal{V}} -$$
$$\int_{t_0}^{t_f} \left\langle (\partial_X \mathbb{H})^* R^{-1}(\mathcal{Y}(t) - \mathbb{H}(X(t)), \frac{\partial X}{\partial \epsilon} \delta \epsilon \right\rangle_{\mathcal{V}} dt. \quad (A.13)$$

Expressions (A.12-A.13) provide the functional gradients in the directions $(\delta u, \delta \epsilon_m)$. We can remark from these expressions that a direct numerical evaluation of these gradients is in practice completely unfeasible. As a matter of fact, such an evaluation would require to compute perturbations of the state variable along all the components of the control variables $(\delta u, \delta \epsilon) - i.e.$ integrate the dynamical model for all perturbed components of the control variables, which is computationally completely unrealistic.

Appendix A.2. Adjoint model

An elegant solution of this problem consists in relying on an adjoint formulation. To that end, the integration over the range $[t_0, t_f]$ of the inner product between an adjoint variable $\lambda \in \mathcal{V}_T$ and relation (A.6) is performed:

$$\int_{t_0}^{t_f} \left\langle \frac{\partial dX}{\partial t}(t), \lambda(t) \right\rangle_{\mathcal{V}} dt + \int_{t_0}^{t_f} \left\langle (\partial_X \mathbb{M}) \, dX(t), \lambda(t) \right\rangle_{\mathcal{V}} dt +$$
$$\int_{t_0}^{t_f} \left\langle (\partial_u \mathbb{M}) \, \delta u(t), \lambda(t) \right\rangle_{\mathcal{V}} dt = 0. \quad (A.14)$$

An integration by parts of the first term yields:

$$-\int_{t_0}^{t_f} \left\langle -\frac{\partial \lambda}{\partial t}(t)) + (\partial_X \mathbb{M})^* \lambda(t), dX(t) \right\rangle_{\mathcal{V}} dt =$$

$$\left\langle \lambda(t_f), dX(t_f) \right\rangle_{\mathcal{V}} - \left\langle \lambda(t_0), dX(t_0) \right\rangle_{\mathcal{V}} + \int_{t_0}^{t_f} \left\langle \delta u(t), (\partial_u \mathbb{M})^* \lambda(t) \right\rangle_U dt, \quad \text{(A.15)}$$

where the adjoint of the tangent linear operators $(\partial_X \mathbb{M})^* : \mathcal{V} \to \mathcal{V}$ and $(\partial_u \mathbb{M})^* : \mathcal{V} \to U$ have been introduced. At this point no particular assumptions nor constraints have been imposed on the adjoint variable. However, we are free to particularize the set of adjoint variables of interest in setting a particular evolution equation or a given set of boundary conditions allowing simplifying the computation of the functional gradient. As we will see it, imposing that the adjoint variable λ is solution of the system:

$$\begin{cases} -\partial_t \lambda(t) + (\partial_X \mathbb{M})^* \lambda(t) = (\partial_X \mathbb{H})^* R^{-1}(\mathcal{Y} - \mathbb{H}(X(t))) \\ \lambda(t_f) = 0, \end{cases} \quad \text{(A.16)}$$

will provide us a simple and accessible solution for the functional gradient. As a matter of fact, injecting this relation into equation (A.15) with $dX(t_0) = \delta\epsilon$ and $dX = (\partial X/\partial u)\delta u(t) + (\partial X/\partial \epsilon)\delta\epsilon$ allows identifying the right hand second terms of the functional gradients (A.12-A.13) and we get

$$\left\langle \frac{\partial \mathcal{J}}{\partial \epsilon}, \delta\epsilon \right\rangle_{\mathcal{V}} = -\left\langle \lambda(t_0), \delta\epsilon \right\rangle_{\mathcal{V}} + \left\langle I_c^{-1}(X(t_0) - \mathbb{C}_b X_b), \delta\epsilon \right\rangle_{\mathcal{V}},$$

$$\left\langle \frac{\partial \mathcal{J}}{\partial u}, \delta u \right\rangle_{U_T} = \int_{t_0}^{t_f} \left\langle \delta u(t), (\partial_u \mathbb{C}_u)^* F^{-1}(\mathbb{C}_u u(t) - u_0) + (\partial_u \mathbb{M})^* \lambda(t) \right\rangle_U dt$$

$$= \left\langle (\partial_u \mathbb{C}_u)^* F^{-1}(\mathbb{C}_u u - u_0)) + (\partial_u \mathbb{M})^* \lambda, \delta u \right\rangle_{U_T}.$$

From these relations, one can now readily identify the two components of the cost function derivatives with respect to the control variables:

$$\frac{\partial \mathcal{J}}{\partial \epsilon} = -\lambda(t_0) + I_c^{-1}(X(t_0) - \mathbb{C}_b X_b),$$

$$\frac{\partial \mathcal{J}}{\partial u} = (\partial_u \mathbb{C}_u)^* F^{-1}(\mathbb{C}_u u - u_0) + (\partial_u \mathbb{M})^* \lambda. \quad \text{(A.17)}$$

The partial derivatives of $\mathcal{J}$ are now simple to compute when the adjoint variable λ is available. The knowledge of the functional gradient enables then

to define updating rules for the control variables from iterative optimization procedures. A quasi-Newton minimization process consists for instance of:

$$X_{n+1}(t_0) = X_n(t_0) - \alpha_n \tilde{H}^{-1}_{X_n(t_0)}(I_c^{-1}(X_n(t_0) - \mathbb{C}_b X_b) - \lambda(t_0)),$$
$$u_{n+1} = u_n - \alpha_n \tilde{H}^{-1}_{u_n}((\partial_{u_n}\mathbb{M})^* \lambda + (\partial_u \mathbb{C}_u)^* F^{-1}(\mathbb{C}_u u_n - u_0)), \tag{A.18}$$

where $\tilde{H}^{-1}_{x_n}$ denotes an approximation of the Hessian inverse computed from the functional gradient with respect to variable x_n; the constant α_n is chosen so that to respect Wolfe conditions. The adjoint variable is accessible through a forward integration of the state dynamics (A.3-A.4) and a backward integration of the adjoint variable dynamics (A.16). Let us point out that considering a final condition for the state variable (through a similar cost function term as for the initial condition) would change the null initial condition of the adjoint dynamics into a term similar to the one involved in the derivative with respect to the initial condition control variable. The overall optimal control process is schematically summarized in Figure A.12.

Acknowledgements

References

[1] Druault, P., Lardeau, S., Bonnet, J. P., Coiffet, F., Delville, J., Lamballais, E., Largeau, J. F., and Perret, L., 2004, "Generation of three-dimensional turbulent inlet conditions for large-eddy simulation", *AIAA J.*, 42, 447.

[2] Griewank, A., 2000, "Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation", *Frontiers in Applied Mathematics, SIAM*.

[3] Hascoet, L., Greborio, R., and Pascual, V., 2003, "Computing adjoints by automatic differentiation with tapenade", *Research report, INRIA*.

[4] Keating, A., Piomelli, U., Balaras, E., and Kaltenbach, H. J., 2004, "A priori and a posteriori tests of inflow conditions for large eddy simulation", *Phys. Fluids*, 16, 4696.

[5] Laizet, S., and Li, N., 2011, "Incompact3d: a powerful tool to tackle turbulence problems with up to $O(10^5)$ computational cores", *Int. J. of Numerical Methods in Fluids*, 67(11), 1735-1757.

[6] Laizet, S., and Lamballais, E., 2009, "High-order compact schemes for incompressible flows: a simple and efficient method with the quasi-spectral accuracy", *J. Comp. Phys.*, 228(16), 5989-6015.

[7] Le-Dimet, F. X., and Talagrand, O., 1986, "Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects", *Tellus* 38 (A), 97-110.

[8] Lions, J.L., 1971, "Optimal control of systems governed by PDEs", Springer-Verlag, New York.

[9] Liu, D., and Nocedal, J., 1989, "On the limited memory BFGS method for large scale optimization", *Mathematical Programming, Series B*, 45(3), 503-528.

[10] Lund, T. S., Wu, X., and Squires, K. D., 1998, "Generation of inflow data for spatially-developing boundary layer simulations", *J. Comput. Phys.*, 140, 233.

[11] Papadakis, N., and Memin, E., 2008, "Variational assimilation of fluid motion from image sequences", *SIAM Journal on Imaging Science*, 1(4), 343-363.

[12] Perret, L., Delville, J., Manceau, R., and Bonnet, J. P., 2008, "Turbulent inflow conditions for large-eddy simulation based on low-order empirical model", *Phys. Fluids*, 20, 075107.

[13] Colburn, C. H., Cessna, J. B., and Bewley, T. R., 2011, "State estimation in wall-bounded flow systems. Part 3. The ensemble Kalman filter", *J. Fluid Mech.*, 682, 289-303.

[14] Zdravkovich, M., 1997, "Flow Around Circular Cylinder - Volume 1: Fundamentals", New York, United States: Oxford University Press Inc.
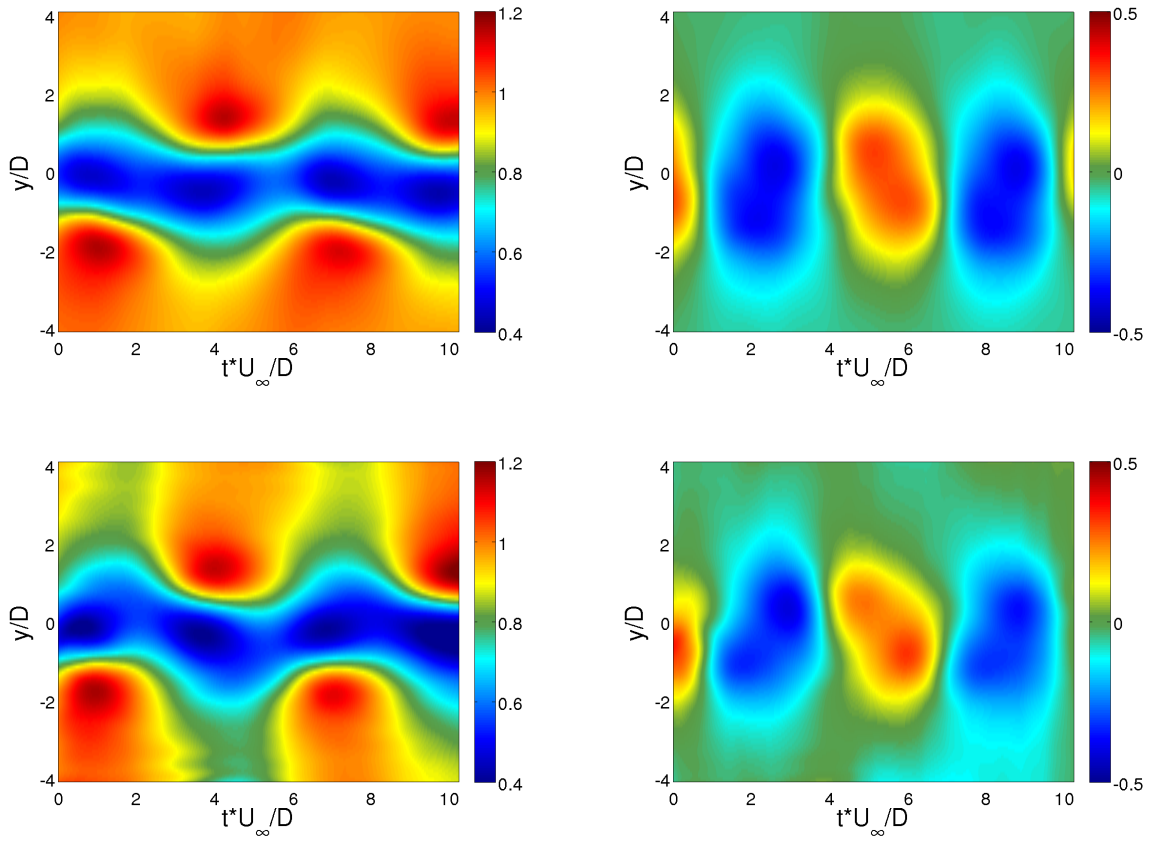
Figure 8: VDA cylinder wake experimental test *A*. Temporal evolution of the inflow condition in the assimilation domain. Longitudinal velocity component (*Left*); Vertical velocity component (*Right*); Initial estimate (*Top*); Best estimate (*Bottom*).
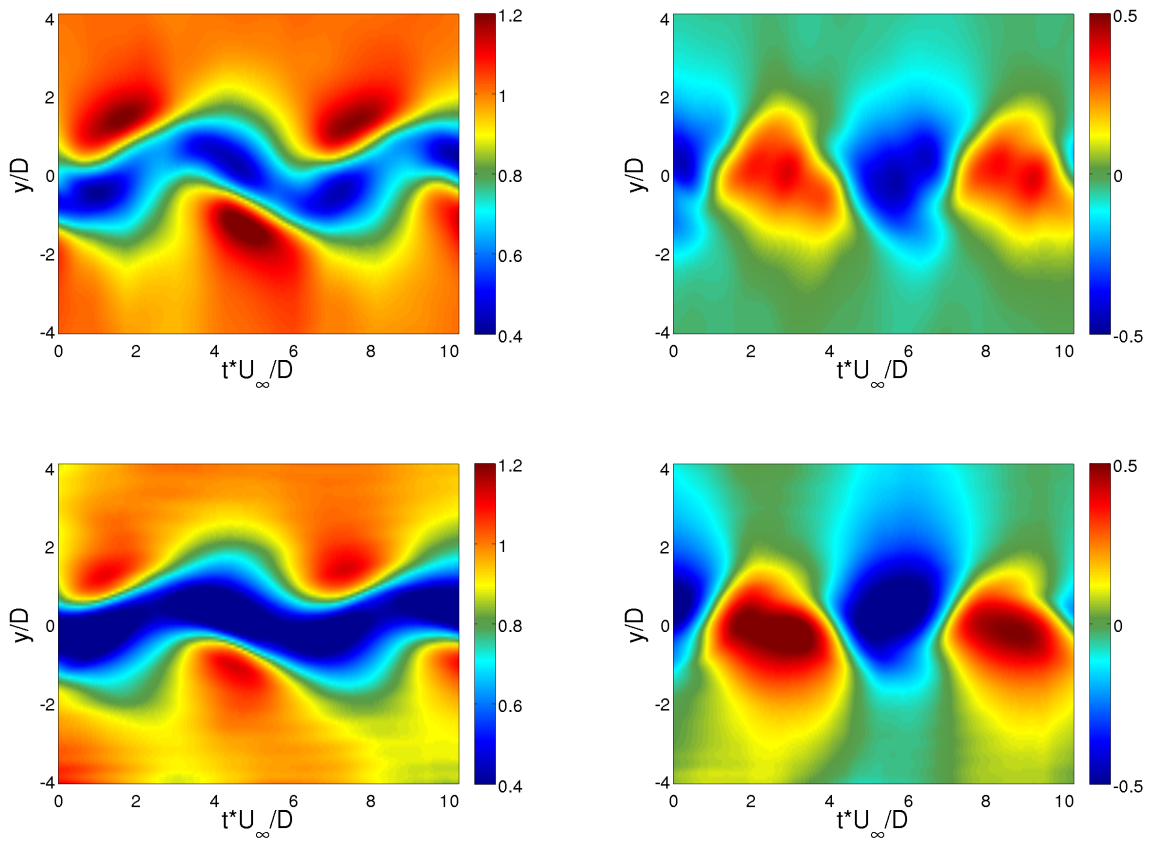
Figure 9: VDA cylinder wake experimental test *B*. Temporal evolution of the inflow condition in the assimilation domain. Longitudinal velocity component (*Left*); Vertical velocity component (*Right*); Initial estimate (*Top*); Best estimate (*Bottom*).
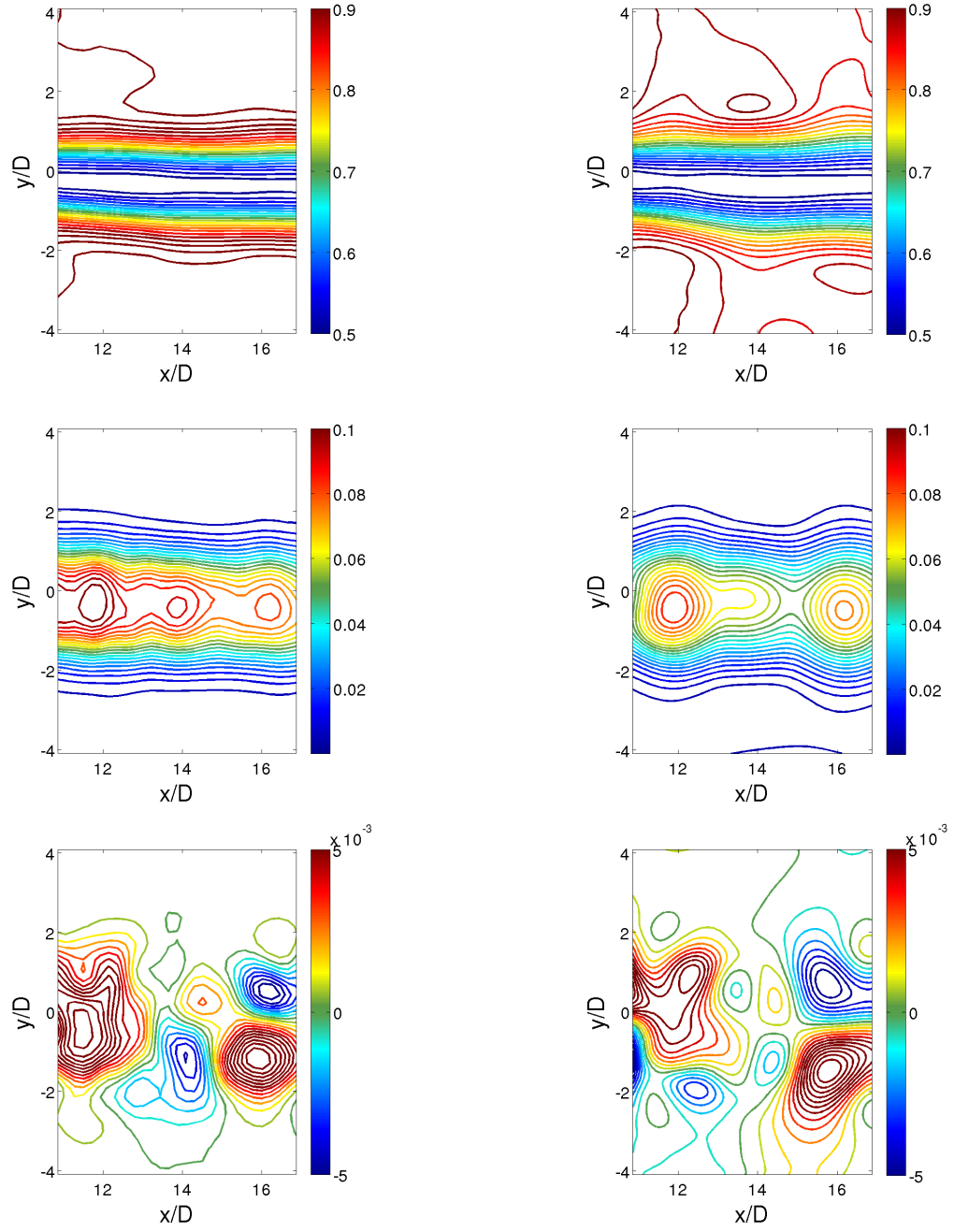
Figure 10: VDA cylinder wake experimental test A. Mean and fluctuating velocity field contours in the assimilation domain. Experimental observations (*Left*); Assimilated state (*Right*); Velocity modulus $\langle u \rangle / U_\infty$ (*Top*); Transverse Reynolds normal stress $\langle u'_y u'_y \rangle / U_\infty^2$ (*Middle*); Reynolds shear stress $\langle u'_x u'_y \rangle / U_\infty^2$ (*Bottom*).
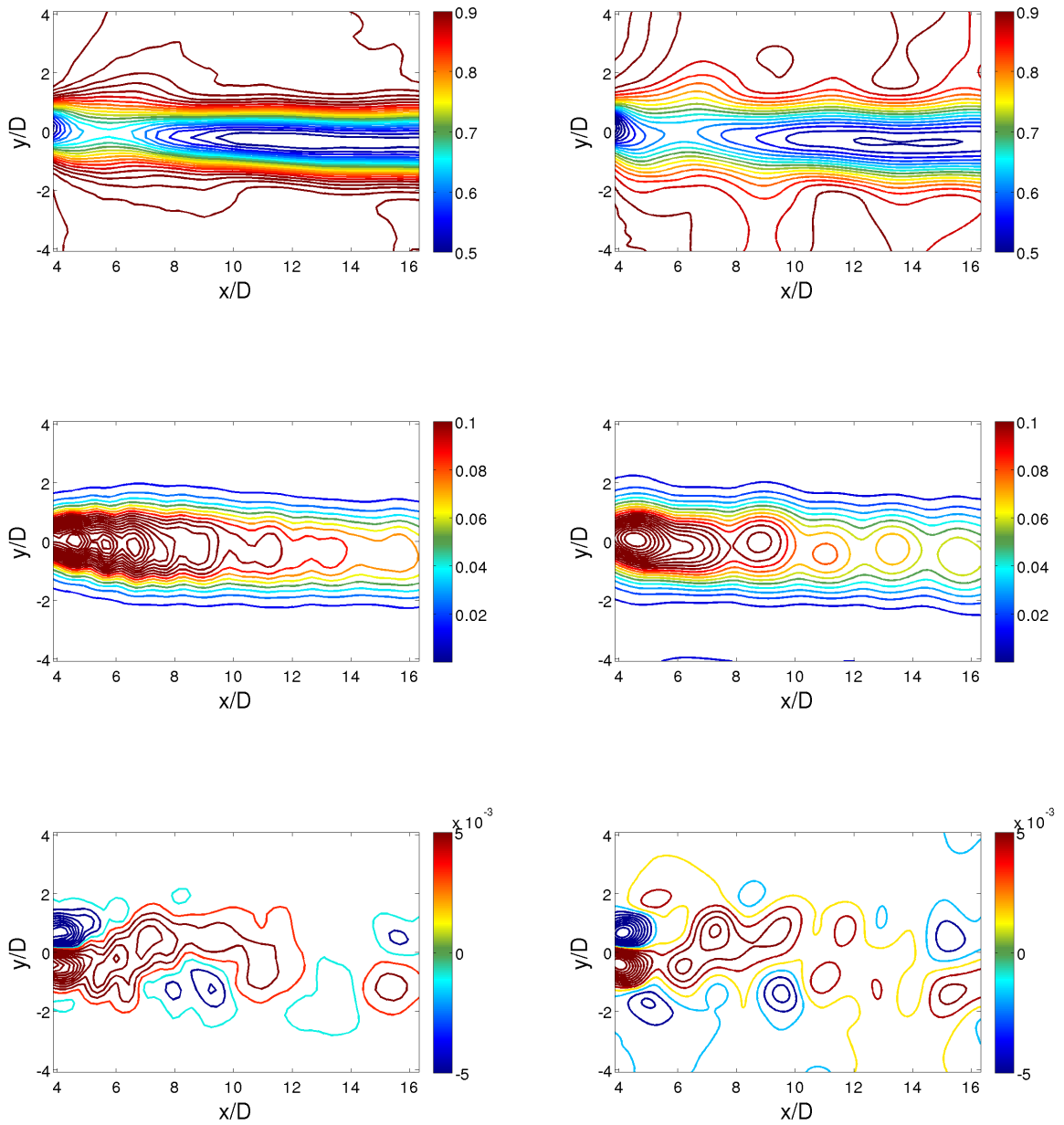
34

Figure 11: VDA cylinder wake experimental test B. Mean and fluctuating velocity field contours in the assimilation domain. Experimental observations (*Left*); Assimilated state (*Right*); Velocity modulus $\langle u \rangle / U_\infty$ (*Top*); Transverse Reynolds normal stress $\langle u'_y u'_y \rangle / U_\infty^2$ (*Middle*); Reynolds shear stress $\langle u'_x u'_y \rangle / U_\infty^2$ (*Bottom*).

1. Set an initial condition: $X(t_0) = X_0$
2. From $X(t_0)$, compute $X(t)$ with the forward integration of relation (A.3)
3. Compute the adjoint variable $\lambda(t)$ with the backward integration of relation (A.16)
4. Update the initial value $X(t_0)$ and the parameter model u with (A.18)
5. Loop to step 2 until convergence

Figure A.12: Schematic representation of the variational data-assimilation algorithm