FLUID Image analysis and Description (FLUID)
Project No. FP6-513663
FET - Open Domain, Priority IST

Deliverable 3.4
Demonstrator on turbulent structure characterization
Nicolas Papadakis, Juan D’adamo, Etienne Mémin and Guillermo Artana
VISTA group, INRIA

Due date of deliverable: 1st September 2007
Actual submission date: 13th September 2007
Start date of project: 1 December 2004
Duration: 3 years

Project cofunded by the European Comission
within the Sixth Framework Programme (2002-2006).

Dissemination level: PU (Public)  Revision: 1
Demonstrator on turbulent structure characterization

Nicolas Papadakis\(^1\)  Juan D’adamo\(^2\)  Etienne Mémin\(^1\)  Guillermo Artana\(^2\)  

\(^1\)IRISA, Université de Rennes 1  
Campus de Beaulieu  
35 042 Rennes Cedex, France  
{npapadak,memin}@irisa.fr

\(^2\)Facultad de Ingeniería  
Buenos Aires, Argentina  
{jdadamo,gartana}@f.uba.ar

Abstract

This deliverable is provided together with source files that allow to construct Low order dynamical system from a Proper Orthogonal Decomposition. It describes how this program can be used. Some results are shown for a simple configuration.

1 Presentation of the method

1.1 Introduction

The reduction of the Navier-Stokes equation to a system of ordinary differential equation (ODE) has been largely studied by the Computational Fluid Dynamics (CFD) community. With different tools developed in this domain, it has been possible to reproduce or predict diverse flow characteristics with a large detail. However, the computational cost of these calculations becomes higher as Reynolds number increases and when turbulent models must be considered. For these reasons, since a couple of years, different research efforts are trying to deal with the same problem through the so called low dimensional dynamic models (LODS) or reduced order models. The goal, which is here less ambitious than a complete flow numerical simulation, consists in capturing and representing the essential characteristics of the flow. These methods aim
at describing only coherent structures dynamics, sacrificing all the flows detailed structures.

Some scenarios where these models are of major interest are in flow control applications where simple systems are required. Actuation on coherent structures may be largely amplified producing important modifications of the flow characteristics as it can take place in process like boundary layer separation, vortex shedding, transition to turbulence, etc. Another field of application of LODS concerns computational optimization problems where one seeks to avoid the repetition of complete CFD calculations for different initial conditions or for slightly different Reynold’s number. Working with adequate LODS would accelerate the tasks in these cases. Furthermore, LODS can be helpful in experimental fluid mechanics as they allow to organize and to interpret large data realizations as well as to extract a model from them.

A way to obtain LODS is by means of the Proper Orthogonal Decomposition (POD) technique. Also known as Karhunen-Loève Decomposition, Singular Value Decomposition, Principal Components Analysis, POD has been first introduced in the context of turbulence by Lumley (1967)[12]. It has been used by different authors (see for instance a review [9]) as a method to obtain approximate descriptions of the large scale or coherent structures in laminar and turbulent flows. Without any \textit{a priori} hypotheses on the flow, POD method provides a flow representation in terms of a mean and a linear combination of basis functions, or modes, ordered decreasingly by their kinetic energy content.

The estimation of the POD basis vectors relies on a set of flow field realizations $u(X,t)$ which can either be obtained from CFD or experiments. Such a technique has been proposed for many situations arising in fluid dynamic problem: a non exhaustive list includes analyses of: shear layers [16], transition in boundary layers [17], turbulent boundary layers [1], flow in a channel [6], flow around a circular cylinder [6], cavity flows [5], etc.

Experimental based POD models have to cope with numerical instability. Such problem is usually tackled adding forced artificial closure terms [3, 4]. These instabilities come mainly from evaluation of inner products and derivatives on the sparse grid of experimental realization. Efforts to overcome this kind of problems were made by different approaches. Particularly polynomial identification technique was proposed firstly in [3], in [14, 15]. The dynamic system coefficients are estimated not directly by Galerkin projection but from least square fitting of experimental data.

In this work, we propose to improve such solution, by relying on variationnal data assimilation framework. Such framework enables to estimate the state of variables of interest characterizing the flow under observation (such as pressure, density, velocity components, salinity etc.) given a dynamical law and sparse and possibly noisy measurements at different time instants.
This approach also allows to handle very large scale systems and as such are intensively used in environmental sciences [2, 11, 21, 19, 20] for atmospheric or oceanic analysis and forecasting. We rely on this technique to estimate in batch mode the complete trajectories along an image sequence of POD modes. As we will show it, such a method allows us to refine a first crude initial estimate obtained through polynomial identification. The method couples an imperfect noisy dynamic model with the whole sequence of observations. Given the whole trajectories of POD modes, the dynamic system coefficients initially provided by noisy PIV data can then be re-estimated from a mean square fitting of the assimilation results. As demonstrated in the experimental section, the technique we propose enables the reconstruction of the most salient characteristics of the flows for a long time range. The stability and the accuracy of the LODS is significantly improve.

The remainder of the article is organized as follow. First the necessary basis to the construction of a flow POD representation from particle image velocimetry (PIV) observations are recalled in section 1.2.1. The way PIV observations can be filtered and improved is described in section 1.2.2. The obtention of LODS from POD basis is described in section 1.2.3. In 1.2.4, we describe the experimental setup we used in this work, and discuss first results obtained through polynomial identification technique. Variational data assimilation principles are presented in section 1.3. The technique we propose for LODS identification is presented in section 1.3. Finally, results obtained for the POD-assimilation scheme are given and studied in section 2.3.

### 1.2 Proper Orthogonal Decomposition

#### 1.2.1 POD basis

POD method has been widely used by different authors as a technique to obtain approximate descriptions of the large scale or coherent structures in laminar and turbulent flows. Given an ensemble \( u(x, t_i) \) obtained experimentally, belonging to M different discrete instants, POD provides M mutually orthogonal basic functions, or modes, \( \phi_i(x) \), which are optimal with respect to average kinetic energy representation of the flux.

Considering such a decomposition enables to write the velocity field as an average \( \bar{u} \) with fluctuations captured by a finite set of modes:

\[
  u(x, t) = \bar{u} + \sum_{i=1}^{M} a_i(t) \phi_i(x).
\]

(1)

Being fields of finite kinetic energy, \( u \in L^2 \) and denoting by \((,\)\) the inner product of functions defined in \(L^2(S)\):
\[(u, \psi) = \int_S u \psi ds, \quad (2)\]

where \(S\) represents the spatial domain occupied by the flux. Seeking a subspace such that the projection of \(u(x, t_i)\) on it is optimal along the sampling time comes to find an ensemble of functions that maximize:

\[\frac{\langle |(u, \psi)|^2 \rangle}{\langle \psi, \psi \rangle},\]

where \(\langle \bullet \rangle\) denotes a temporal average. It can be demonstrated [9] that the optimal functions \(\phi\) also satisfy the following eigenvalue problem:

\[\int_S K(x, x') \phi_k(x) \, dx = \phi_k(x) \lambda_k, \quad (3)\]

with

\[K(x, x') = \langle u(x, t) u(x', t) \rangle = \frac{1}{M} \sum_{i=1}^{M} u(x, t_i) u(x', t_i).\]

To solve our problem, it is easier numerically to follow Sirovich’s Snapshots method [18] which states that each spatial mode \(\phi\) can be constructed by a superposition of the velocities fields:

\[\phi_k(x) = \sum_{i=1}^{M} u(x, t_i) a_k(t_i).\]

Projecting (3) into a snapshot \(u(x, t_j)\), we obtain another eigenvalue problem, whose eigenvectors are the temporal modes \(a_k\).

### 1.2.2 Gappy POD

A problem arises when the snapshot \(u(x, t_i)\) given through a PIV technique contains erroneous vectors. To correct this deficiency, Everson and Sirovich [7] have proposed an iterative scheme. We consider the same kind of setup in this work. It first consists in creating an ensemble of masks \(m(x, t_i)\), with values either zero or unity if there is at position \(x\) of time \(t_i\) a wrong or a reliable vector \(u(x, t_i)\). We can respectively write the velocity fields with data missing, \(\tilde{u}(x, t_i)\) in terms of the masks \(m(x, t_i)\) and the complete velocity fields \(u(x, t_i)\):

\[\tilde{u}(x, t_i) = m(x, t_i) u(x, t_i).\]

To compute the temporal average of \(\tilde{u}(x, t_i)\), we only take into account the reliable values,

\[\tilde{u}_m(x) = \langle \tilde{u}(x, t_i) \rangle = \frac{1}{\sum_{m(x, t_i)} \sum_{i} m(x, t_i) u(x, t_i)}.\]
A first correction of the erroneous data is achieved by replacing wrong values by the temporal average. So for each \((x, t_i) \notin S\)

\[
\tilde{u}^{(0)}(x, t_i) = \tilde{u}_m(x).
\]

With these corrected values, a POD can be settled to provide an initial set of spatial and temporal modes: \(\phi^{(0)}(x) = \{\phi_k^{(0)}(x), k = 1, \ldots, M\}\) and \(a^{(0)}(t_i) = \{a_k^{(0)}(t_i), k = 1 \ldots, M\}\).

The restored vector field at the following iteration, \(\tilde{u}^{(1)}(x, t_i)\), is obtained by fitting each member of the original ensemble, \(\tilde{u}(x, t_i)\), to a superposition of \(M\) eigenfunctions \(\phi_k^{(0)}(x)\) as follows:

\[
\tilde{u}(x, t_i) = \sum_{k=1}^{M} a_k^{(1)}(t_i) \phi_k^{(0)}(x) \quad \forall (x, t_i) \in S,
\]

\[
\left( \phi_j^{(0)}, \tilde{u}(x, t_i) \right) = \left( \phi_j^{(0)}, \sum_{k=1}^{M} a_k^{(1)}(t_i) \phi_k^{(0)}(x) \right).
\]

From \(\phi_k\) orthonormality, we can write:

\[
a_j^{(1)}(t_i) = \left( \phi_j^{(0)}, \tilde{u}(x, t_i) \right) \quad \forall j = 1 \ldots M. \tag{4}
\]

From the set of estimated temporal modes \(a_j^{(1)}(t_i)\) and the basis functions \(\phi^{(0)}(x)\), the updated set of velocity values \(\tilde{u}^{(1)}\) is obtained. This process is iteratively repeated until a convergence criteria is met. Further details on this method can be found in [22] for the particular case of PIV data.

1.2.3 Formulation of a low order dynamical system (LODS)

Galerkin projection From the modal decomposition, it is possible to consider a truncated model with \(s\) modes to approximate the velocity field \(u(x, t_i)\), with \(x \in S\), \(1 \leq i \leq M\) and \(s \ll M\).

From the properties of POD, it is easy to measure the kinetic energy percentage contained in this model:

\[
\frac{\sum_{i=1}^{s} \lambda_i}{\sum_{i=1}^{M} \lambda_i}.
\]

A Galerkin projection enables to rewrite a partial differential equation (PDE) system as a system of ordinary differential equation (ODE). According to this procedure, the functions which define the original equation are projected on a finite dimensions subspace of the phase space (in this case, the subspace generated by the first \(s\) modes).

Following a scheme proposed by Rajaee, Karlsson and Sirovich [16], we project the Navier
Stokes equations under the Reynolds decomposition, in order to have an explicit expression for the fluctuating quantities:

\[
\left( \frac{\partial u'}{\partial t} + u' \nabla \bar{u} + \bar{u} \nabla u' + u' \nabla u' - \bar{u} \nabla u' + \frac{\nabla p'}{\rho} - \nu \nabla^2 (\bar{u} + u') \right), \phi_j = 0. \tag{5}
\]

These equations are obtained by separating flow velocity into mean $\bar{u}$ and fluctuating $u'$ parts: $u = \bar{u} + u'$. Rewriting (5) in terms of POD (1), the resulting equation is a quadratic ODE of order 1. For every $j \leq s$ modes, the system reads:

\[
\frac{da_k}{dt} = F(a_k) = \sum_{i=1}^s l_{ik} a_i + \sum_{i=1}^s \sum_{j=i}^s a_i c_{ijk} a_j, \quad k = 1 \cdots s \tag{6}
\]

where

\[
l_{ij} = \int_S \bar{u} \nabla \phi_i \phi_j ds + \int_S \phi_i \nabla \bar{u} \phi_j ds - \int_S \frac{1}{Re} \Delta \phi_i \phi_j ds, \tag{7}
\]

\[
c_{ijk} = \int_S \phi_j \nabla \phi_i \phi_k ds, \tag{8}
\]

\[
i_k = \int_S \nabla p' \phi_k ds - \frac{1}{Re} \int_S \Delta \bar{u} \phi_k ds - \sum_{j=1}^s \lambda_j \int_S \phi_j \nabla \phi_j \phi_k ds. \tag{9}
\]

Regarding these expressions, (7) describes the interaction between the mean flow and fluctuating field, it also includes viscous effects from the modes. Nonlinear effects are reported by (8). The independent term (9) takes into account mean flow dissipation, convective contribution of the modes and the pressure field influence.

Boundary conditions and symmetry make the pressure term vanish in particular case of wake flow. As a matter of fact, each of the modes functions satisfies the continuity equation, to give:

\[
\int_S \nabla p' \phi_k ds = \oint_C p' \phi_k dc,
\]

where $C$ is the boundary curve of domain $S$. Works of Deane [6] and Noack [13] demonstrated that for wake flow configuration, the latter expression is negligible compared to the other terms.
Nevertheless, the inclusion of $p'$ term can be modeled through an additional quadratic expression of the temporal modes $a$, which is achieved by polynomial identification. A noisy version of the momentum equation (6) can also be considered to deal with the $p'$ term. Such a situation will be the core of the POD-assimilation technique.

Direct calculation of each term of the system (6) can be avoided by using polynomial identification, described as follows.

### 1.2.4 First results and LODS adjustment

A common problem regarding reduced order models is how to model the unresolved modes of the flow. Although we have a model that enables to represent the most energetic structures, associated with the flow greatest scales, the smaller ones should also be included as they play an important role in the dissipation process. Practical implementations showed that (6) can only be solved for short time range when neglecting the effect of small scales.

This problem was first pointed out by Holmes et al. [9], where a polynomial similar to (6) is modified after an estimation of the incoherent, unresolved modes in terms of the resolved modes. Long time behaviour of LODS solution, can be seen as an attracting set. We can consider every observation $a_i$ as an element of this set. Another way to refine the model, first proposed by Braud [3, 4] consists in estimating the polynomial coefficients through least squares fitting. Provided each observation $a_i$ and its derivative $\dot{a}_i$, we can write (6) as the solution of a linear system. His polynomial identification technique, has been tested on the Lorenz dynamical model and on POD dynamical systems. This method presents the great advantage to avoid the accurate computation of the basis functions spatial derivatives that are required to construct the projected model (7-9). It can pointed out also, that even if closed pairs of observation $a_i(t_k)$ and $a_i(t_{k+1})$ are needed to compute the temporal derivative (6), those pairs are not required to be correlated.

The experimental configuration we settled for this work consists of a flow around a circular cylinder at low Reynolds, $Re = 125$. The velocity measurements have been done in a closed loop wind tunnel with a probe section of 18x18 cm$^2$. A 2 cm diameter cylinder was placed in order to have snapshots of 5 by 4 diameters. We chose it regarding the wake structures we wanted to identify.

As for the PIV system, we used a Pixelfly PCO VGA camera, with a resolution of 640x240 pixels, 1/100 s of time between images. A green laser Intelite GM32-150IH, 150 mW, combined with a rotating polyhedric mirror, was used to provide illumination on the probe section.
The algorithms used in this work belong to GPIV software [8] which is under GNU General Public License. We have considered for our images a two step grid refinement, so the final interrogation size is 16x16 pixels with a 50% overlapping.

The time resolution was suitable to recover the flow dynamics, in a way that the vortex shedding frequencies are lower than the acquisition frequency. Although the technique does not require time correlated data, this condition allowed us to validate our models.

A set of 1000 images was used to construct the reduced order model. The data has been filtered by the gappy technique and then the fluctuating velocity field was decomposed with POD. As expected in a low turbulence flow, the first modes concentrate the most of the fluctuating kinetic energy. As illustrated in Figure 1, the spatial modes extracted from the decomposition represent the recurrent structures of the flow. We can observe that structures with higher fluctuating kinetic energy concentration results appears to be more spatially organized.

![Figure 1: Contour maps of spatial modes: a) φ₁(𝐱) b) φ₃(𝐱)](image)

In a first experiment, we choose to keep only $s = 2$ modes to be assured to recover the 91% of kinetic energy. The coefficients of equation (6) have been estimated by polynomial identification based on mean squares fitting. The corresponding solution of the LODS is plotted on Figure (2a) where we have plotted also the original data for comparison purpose. Although the short term behaviour of the model is quite accurate, significant errors in magnitude and phase appear very quickly. These errors amplify as time goes by. For a system with a larger number of modes the solution does not converge at all. This is illustrated in Figure (2b) where
we have plotted the solution for a LODS with 4 modes. In order to improve the estimation of the

![Graph A](image1.png)

![Graph B](image2.png)

Figure 2: First estimation of the ODE. Comparison between ODE solution (solid line) and original data (symbols). a) for \( s = 2 \). b) Solution diverges for systems \( s \geq 4 \).

temporal modes we will place ourself within the framework of variationnal data assimilation. We briefly describe its principles in the next section.

### 1.3 Data Assimilation principle

#### 1.3.1 Introduction

Data Assimilation is a technique that enables to perform the estimation over time of state variables representing a system of interest. The method enables to perform a smoothing of noisy measurements of the system’s state according to a given initial state of the system and a dynamic law. Let us note \( X \in \Xi \) the state variable of interest. This variable may represent any quantities associated to the observed flow such as temperature, velocity, vorticity, pressure etc. Assuming
the evolution in time of these quantities is described through a (non linear) differential model $\mathbb{M}$
we get the following dynamical system:
\[
\begin{align*}
\frac{dX}{dt} + \mathbb{M}(X, v) &= 0 \\
X(t_0) &= X_0 
\end{align*}
\]  
(10)

This system is monitored by a control variable $u = (v, X_0) \in P$, defined in control space. This control variable may be set to the initial condition or to any free parameter of the evolution law.

Let us also assume that some observations $Y \in O_{obs}$ of the state variable components are available. These observations may live in a different space (a reduced space for instance) from the state variable. We will nevertheless assume that there exists a matrix operator $H$, that goes from the variable space to the observation space. A least squares estimation of the control variable regarding the whole sequence of measurements available within a considered time range comes to minimize with respect to the control variable a cost function of the following form:
\[
J(u) = \frac{1}{2} \int_{t_0}^{t_f} ||Y - HX(v, X_0))||^2 dt. 
\]  
(11)

A first approach consists to compute the functional gradient through finite differences:
\[
\nabla_u J(u) \simeq \frac{J(u + \epsilon e_k) - J(u)}{\epsilon},
\]
where $\epsilon \in \mathbb{R}$ is an infinitesimal perturbation and $\{e_k, k = 1, \ldots, p\}$ denotes the unitary basis vectors of the control space. Such a computation is impractical for space of large dimension since it requires $p$ integrations of the evolution model for each required value of the gradient functional. Adjoint models as introduced first in meteorology by Le Dimet and Talagrand in [11] will allow us to compute the gradient functional in a single integration.

Denoting $\delta u = X(du)$ a perturbation of the solution corresponding to $du \in P$ we have:
\[
\delta J = <\nabla J_u, \delta u>
= - <H^T(Y - H(X(v, X_0))), \delta u>
\]  
(12)

This perturbation evolves according to:
\[
\frac{d}{dt}\delta u + \partial_u \mathbb{M}\delta u = \frac{d}{dt}\delta u + \partial_{X,v}\mathbb{M}\delta X_0 + \partial_v\mathbb{M}\delta v = 0.
\]
The right part of this linear evolution law involves the so called tangent linear model of $\mathbb{M}$. It is define as the the Gâteaux derivative at point $X$ of the operator $\mathbb{M}$:
\[
\lim_{\beta \to 0} \frac{d\mathbb{M}(X + \beta \theta)}{d\beta} = \partial_X \mathbb{M}\theta.
\]  
(13)
This linear operator describes how arbitrary perturbations of the control evolves in time.

Now considering the inner product of the tangent linear model with an arbitrary vector \( \lambda \) and an integration by part leads to:
\[
\int_O^T < \frac{d}{dt} \delta u, \lambda > dt = < \delta u(T), \lambda(T) > - < \delta u(0), \lambda(0) > - \int_O^T < \delta u, \frac{d}{dt} \lambda > dt,
\]

Imposing that \( \lambda(T) = 0 \), introuducting of the tangent linear model and using the definition of an adjoint operator associated to a given inner product (i.e. \( < x, L y > = < L^* x, y > \)), we get:
\[
< \delta u(0), \lambda(0) > = \int_O^T < \delta u, \partial_u M^* \lambda > dt - \int_O^T < \delta u, \frac{d}{dt} \lambda > dt
\]
\[
= \int_O^T < \delta u, - \frac{d}{dt} \lambda + \partial_u M^* \lambda > dt.
\]

We define the adjoint model requiring in addition that:
\[
- \frac{d}{dt} \lambda + \partial_u M^* \lambda = H^T (Y - H(X(v, X_0))),
\]
This adjoint model together with the definition of the gradient functional (12) enables to write:
\[
\nabla_u J = - \lambda(0).
\]

As a consequence, the functional gradient can be computed as a single backward integration of an adjoint model. The value of this adjoint variable at the initial time provides the value of the gradient. This first approach is widely used in environmental sciences for the analysis of geophysical flows. However, these methods rely on a perfect dynamical model. Considering, imperfect models, defined up to a gaussian noise comes to an optimization problem where the control variable is constituted by the whole trajectory of the state variable. This is the kind of problem we are facing in this work.

The ingredients of the new data assimilation problem is now composed by an imperfect dynamic model of the target, an initialization of the state variable and an observation equation which relates the state variables to some measurements:
\[
\begin{align*}
\frac{dX}{dt} + M(X) &= \nu(t) \\
X(t_0) &= X_0 + \eta \\
Y(t) &= HX + \epsilon(t).
\end{align*}
\]

In these three equations \( \eta, \nu \) and \( \epsilon \) are time varying zero mean Gaussian noise vector functions. They are respectively associated to covariance matrices \( W(t, t') \), \( B \) and \( R(t, t') \). The noise
functions represent the different errors involved in the different components of the system (i.e. model errors, initialization errors and measurement errors) and are assumed to be uncorrelated in time. The goal is to minimize the new functional:

$$J(X) = \frac{1}{2} \int_{t_0}^{t_f} \left| \frac{dX}{dt} + M(X) \right|^2 dX + \frac{1}{2} \|X(t_0) - X_0\|^2_T + \frac{1}{2} \int_{t_0}^{t_f} \|HX(U, V) - Y\|^2_R dt$$ (15)

The minimization has now to be done according to the state variable $X$.

1.3.2 Minimization of the functional

A minimizer $X$ of functional $J$ is also a minimum of a cost function $J(X + \beta \theta(t))$, where $\theta(t)$ belongs to a space of admissible function and $\beta$ is a positive parameter. In other words, $X$ must cancel the directional derivative:

$$\delta J_X(\theta) = \lim_{\beta \to 0} \frac{dJ(X + \beta \theta(t))}{d\beta} = 0.$$ (16)

The functional with an infinitesimal perturbation reads:

$$J(X + \beta \theta) = \frac{1}{2} \int_{t_0}^{t_f} \left( \frac{dX}{dt} + \beta \frac{d\theta}{dt} + M(X + \beta \theta) \right)^T \int_{t_0}^{t_f} W^{-1}(t, t') \left( \frac{dX}{dt} + \beta \frac{d\theta}{dt} + M(X + \beta \theta) \right) dt' dt
+ \frac{1}{2} (X + \beta \theta - X_0)^T B^{-1}(X + \beta \theta - X_0)
+ \frac{1}{2} \int_{t_0}^{t_f} \int_{t_0}^{t_f} (Y - H(X + \beta \theta))^T (t) R^{-1}(t, t')(Y - H(X + \beta \theta))(t') dt' dt.$$

In order to derive a practical definition of the gradient functional we introduce again an adjoint variable $\lambda$ defined as:

$$\lambda(t) = \int_{t_0}^{t_f} W^{-1}(t, t') \left( \frac{dX}{dt} + M(X) \right) dt'. (17)$$

By taking the limit $\beta \to 0$, the derivative of expression (16) then reads

$$\lim_{\beta \to 0} \frac{dJ}{d\beta} = \int_{t_0}^{t_f} \left( \frac{d\theta}{dt} + \partial_X M \theta \right)^T (t) \lambda(t) dt + \theta^T (t_0) B^{-1}(X(t_0) - X_0)
- \int_{t_0}^{t_f} \int_{t_0}^{t_f} H^T \theta^T (t) R^{-1}(t, t')(Y - HX)(t') dt' dt
= 0.$$ (18)
Applying integrations by parts, we can get rid of the partial derivatives of the admissible function \( \theta \) in expression (18). This equation (18) can be then rewritten as

\[
\lim_{\beta \to 0} \frac{dJ}{d\beta} = \theta^*(t_f)\lambda(t_f) + \theta^T(t_0) \left[ B^{-1}(X(t_0) - X_0) - \lambda(t_0) \right] \\
+ \int_{t_0}^{t_f} \theta^T(t) \left[ \left( -\frac{d\lambda}{dt} + \partial_X M^* \lambda \right)(t) - \int_{t_0}^{t_f} H^T R^{-1}(t, t') (Y - HX'')dt' \right] dt \\
= 0.
\]

### 1.3.3 Forward/backward equations

Since the functional derivative must be null for arbitrary independent admissible functions in the three integrals of expression (19), all the other members appearing in the three integral terms must be identically null. We finally obtain a coupled system of forward and backward PDE’s with two initial and end conditions:

\[
\lambda(t_f) = 0 \quad (20)
\]

\[
-\frac{d\lambda}{dt} + \partial_X M^* \lambda = \int_{t_0}^{t_f} \partial_X H^* R^{-1}(t, t')(Y - HX')dt' \quad (21)
\]

\[
\lambda(t_0) = B^{-1}(X(t_0) - X_0) \quad (22)
\]

\[
\frac{dX}{dt} + M(X) = \int_{t_0}^{t_f} W(t, t') \lambda(t') dt' \quad (23)
\]

The forward equation (23) corresponds to the definition of the adjoint variable (17) and has been obtained introducing \( W \), the pseudo-inverse of \( W^{-1} \), defined as [2]:

\[
\int_{t_0}^{t_f} W(t, t')W^{-1}(t', t'')dt' = \delta(t - t'').
\]

We can see that eq. (20) constitutes an explicit end condition for the adjoint evolution model eq.(21). As previously, the adjoint evolution model has to be integrated backward from the end condition assuming the knowledge of an initial guess for \( X \) to compute the discrepancy \( Y - HX \). This model is defined from the expression of the adjoint evolution operator. A discrete expression of this operator can be easily obtained when the discretization of the linear tangent operator can be expressed as a matrix. It consists in that case to the transpose of that matrix.

Knowing a first solution of the adjoint variable, an initial condition for the state variable can be obtained from (22) and a pseudo inverse expression of the covariance matrix \( B \). From this initial condition, (23) can be finally integrated forward.
1.3.4 Incremental state function

The previous system can be slightly modified to produce an adequate initial guess for the state variable. Considering a function of state increments linking the state function and an initial condition function, \( \delta X = X - X_0 \), and linearizing the operator \( \mathcal{M} \) around the initial condition function \( X_0 \):

\[
\mathcal{M}(X) = \mathcal{M}(X_0) + \partial_{X_0} \mathcal{M}(\delta X),
\]

we can split equation (23) into two PDE’s with an explicit initial condition:

\[
\begin{align*}
X(t_0) &= X_0 \quad \text{(24)} \\
\frac{dX_0}{dt} + \mathcal{M}(X_0) &= 0 \quad \text{(25)} \\
\frac{d\delta X}{dt} + \partial_{X_0} \mathcal{M} \delta X &= \int_{t_0}^{t_f} W(t, t') \lambda(t') dt'. \quad \text{(26)}
\end{align*}
\]

Let us note that if the model is assumed to be perfect as in the introduction case, we would have \( W = 0 \) and recover the initial system of equation. The incremental system associated to an imperfect dynamical model highlights a major difference with the classic assimilation scheme. As \( W \) is not null, the solution is updated with all the values of the adjoint variable trajectory.

Combining equations (20-22) and (24-26) leads to the final assimilation algorithm. The method consists first in a forward integration of the initial condition \( X_0 \) with the state variable’s evolution model (25). The current solution is then corrected by performing a backward integration (20, 21) of the adjoint variable. The evolution of \( \lambda \) is guided by a discrepancy measure between the observation and the estimate: \( Y - HX \). The initial condition is then updated through equation (22) and a forward integration of the increment \( \delta X \) is realized through the equation (26). The estimation is updated at each step: \( X := X + \delta X \). The overall process is iteratively repeated until convergence (see figure 3).

1.4 Application to the LODS

We describe now, how such a frameworks has been applied to the problem of LODS coefficient estimation.

\footnote{The linearization is equivalent to the Gâteaux derivative defined previously}
1.4.1 Description of the problem

We want to use the assimilation system in order to enhance the estimation of the coefficient of the LODS. The following LODS-assimilation problem is introduced:

\[
\begin{align*}
\frac{da}{dt} + M(a) &= \nu(t) \\
a(t_0) &= a_0 + \eta \\
Y(t) &= Y(a) + \epsilon(t).
\end{align*}
\]  

The right hand side of the first equation describes, through a differential operator $M$, the evolution of the state function $a = [a_1(t) \cdots a_s(t)]$ composed of the POD temporal coefficients and defined over the whole time range $[t_0; t_f]$. In our case, the model and the associated operator $M$ is given through equation (6):

\[
\frac{da_k(t)}{dt} = i_k + \sum_{i=1}^s l_{ik}a_i + \sum_{i=1}^s \sum_{j=1}^s a_i c_{ijk}a_j \\
M(a_k) = 1; \ldots; s.
\]  

We assume here that considering an evolution model defined up to a gaussian variable will allow us to model the effect of unresolved modes of the flow and therefore will enable a better accuracy of the recovered solution on a longer time range. The second equation of the system fixes an initial condition for the state vector through a given initialization $a_0$. The last equation links an observation function $Y(t)$, constituted by noisy measurements of the state function components, to the state function. In the current application, the measurements are given by
the temporal modes estimated from PIV snapshots and a POD technique. As the measurements belong to same space as the state variable we have $H = Id$ in this case.

1.4.2 Linear Tangent operator

In this section, we describe the discretization of the linear tangent operator of the considered reduced dynamical system. Starting from the dynamical equation

$$\frac{da_k(t)}{dt} = i_k + \sum_{i=1}^{s} l_{ik} a_i + \sum_{i=1}^{s} \sum_{j=1}^{s} a_i c_{ijk} a_j = -M(a_k) \quad k = 1, \ldots, s, \quad (29)$$

we simply have to compute the linear tangent operator $\partial aM(\theta)$ for a small perturbation $\theta(t) = [\theta_1(t) \cdots \theta_s(t)]^T$:

$$\partial aM(\theta_k) = -\left[ \sum_{i=1}^{s} l_{ik} \theta_i + \sum_{i=1}^{s} \sum_{j=1}^{s} (a_i c_{ijk} \theta_j + \theta_i c_{ijk} a_j) \right] \quad k = 1, \ldots, s. \quad (30)$$

And finally:

$$\partial aM(\theta_k) = -\left[ \sum_{i=1}^{s} l_{ik} \theta_i + 2 \sum_{i=1}^{s} \sum_{j=1}^{s} a_i c_{ijk} \theta_j \right] \quad k = 1, \ldots, s. \quad (31)$$

Hence, we obtain

$$\partial aM(\theta) = -(L + 2C)\theta. \quad (32)$$

where $L$ and $C$ are matrix $(s \times s)$:

$$L = \begin{bmatrix} l_{11} & l_{12} & \cdots & l_{1s} \\ l_{21} & l_{22} & \cdots & l_{2s} \\ \vdots \\ l_{s1} & l_{s2} & \cdots & l_{ss} \end{bmatrix}, \quad (33)$$

$$C = \begin{bmatrix} \sum_{j=1}^{s} a_j c_{1j1} & \sum_{j=1}^{s} a_j c_{1j2} & \cdots & \sum_{j=1}^{s} a_j c_{1js} \\ \sum_{j=1}^{s} a_j c_{1j2} & \sum_{j=1}^{s} a_j c_{1j2} & \cdots & \sum_{j=1}^{s} a_j c_{1js} \\ \vdots \\ \sum_{j=1}^{s} a_j c_{1js} & \sum_{j=1}^{s} a_j c_{1js} & \cdots & \sum_{j=1}^{s} a_j c_{1js} \end{bmatrix}. \quad (34)$$
1.4.3 Convergence and numerical stability

Recalling that Euler-Lagrange equations consists to the functional $J$ around a small perturbation $\theta$:

$$J(a + \theta) = J(a) + \nabla J \cdot \theta,$$

(35)

and that equation (19) give us an analytic representation of the second part of (35):

$$\nabla J = \lim_{\alpha \to 0} \frac{dJ}{dx},$$

allows to define a natural convergence criterion:

$$\nabla J \cdot \theta < \epsilon.$$  

(36)

These two expressions of the gradient provides a practical way to determine if the adjoint equation is well discretized:

$$\lim_{\alpha \to 0} \frac{J(a + \alpha \theta) - J(a)}{\alpha \nabla J \cdot \theta} \to 1.$$  

(37)

The adjoint computation of the functional gradient is here compared to finite differences implementation. We have checked the validity of our implementation considering this test on a set of real data. The curve showing the ratio for different values of $\alpha$ is presented in figure 4. It can be observed that our discretization is valid up to a $10^{-8}$ variation. This bound is due to the numerical round off errors. This study gives us an adhoc way to set the convergence threshold in equation (36). For the present study, we fixed it to $\epsilon = 10^{-7}$.

![Figure 4: Gradient test realized with $\alpha \to 0$.](image-url)
2 Illustrations

The program corresponding to the previously described method has been implemented in matlab. The different input and output of the program are firstly given. Afterwards, some illustrations of results obtained on test sequences are presented.

2.1 User guide

To run the program, the user must first run matlab. The code is composed of one single matlab file. The construction of the low order dynamical system necessitates a previous computation of the POD basis. All the information concerning the POD system (temporal and spatial modes, time interval between observations, etc....) must then be saved in a file called "data_file.mat". An example of such a file is given with the program. It must contains:

- \( at \): the matrix of temporal modes.
- \( fiu \) and \( fiv \): the matrix of the two components \( u \) and \( v \) of the spatial modes.
- \( dt \): the time interval between 2 observations.
- \( D \): the diameter of the cylinder.
- \( U \): the density of the temporal modes (if the PIV data have one vector for 10 pixels, \( U = 0.1 \)).
- \( h1 \): the x discretization step of the spatial modes.
- \( h2 \): the y discretization step of the spatial modes.
- \( Re \): the Reynolds number of the flow.

The user must then only give the following arguments to the function pod.m

2.1.1 Inputs

- \( T \): length of the time interval.
- \( s \): the number of considered modes.
- The value of the covariance of the observations (default recommended value : 10).
2.2 Outputs

There are 2 types of outputs to the program:

- **Results.mat**: a file containing the assimilated values of the low order dynamical systems parameters: I, L and C.
- **Matlab figures showing the trajectories of the temporal modes obtained with the initial and the assimilated system.**

To show the results on the POD basis contained in "data_file.mat" for \( s = 4 \) modes a time interval of 100 observations and an observation covariance of 0.02, one can run in matlab:

```
pod(4,100,0.02)
```

The program show the trajectories of the temporal modes obtained with the initial and the assimilated low order dynamical systems, as illustrated in figure 5.

![Figure 5: Estimation for a 4 modes system. Trajectories obtained with the initial (a) and the assimilated (b) low order dynamical systems.](image)

2.3 Results

We now present and analyse results obtained for the assimilation of POD modes.

2.3.1 Analysis of robustness

As mentioned in the previous section, it is crucial for assimilation technique associated to imperfect evolution models to have good initial state trajectory. This first guess, can be provided
knowing a good initialization of the state vector at the initial state and integrating this initial condition with the model dynamics. Such initial trajectories of the state variable can be also provided by other estimation techniques. In this works, the initial trajectory is assume to be provided through polynomial identification technique described in 1.2.4. In the case of a 6 modes system we estimate an artificial viscosity for the linear term (7) \( \nu_A = 2.1\nu \), the corresponding solution is presented in Figure 6a. The inclusion of such artificial vorticity allows system to remain stable. Even if it has been not possible not obtain a limit cycle for the temporal modes, this solution provides us a reliable first guess for the assimilation technique. Greater values of \( \nu_A \) produced a too strong damping, and leads to initial solution of to bad quality.

![Figure 6: Estimation for a 6 modes system. Equation solution in solid line and original data in symbols. a) POD estimation with strong damping. b) POD Assimilation result.](image-url)
As can be observed on Figure 6b, even though the considered initial guess was quite far from a good solution, the POD-Assimilation method provides a significant improvement.

2.3.2 New LODS

Another important point is that the coefficient matrices I, L and C can be updated from the assimilation results, using the least squares fit. The initial measurements used to evaluate the matrices are discrete and noisy. As a result of the assimilation process, since the solution provided for \( a(t) \) is continuous, the temporal derivatives are much better estimated. Polynomial identification can be performed again to obtain the new coefficient matrices which allows us to recompute new coefficient \( a(t) \) through least squares estimation. We compare in Figure 7, for the first 2 modes, the solution obtained by polynomial identification, the POD-Assimilation results and the results of a polynomial identification on the assimilation result.

2.3.3 Analysis of phase portraits

Although in the third mode, and the followings, the energy content is very low compared to the first two modes, we included it in our model. The resulting dynamical system for 3 modes presents a trajectory, plotted in Figure 8 IIa, which cannot be confounded with a noise.

To obtain a 3 modes LODS, an iterative scheme was applied. Firstly, we used polynomial identification to have a matrix coefficients’(6) estimation. Artificial viscosity, \( \nu_A = 0.1\nu \) was added to confine the ODE solution. Secondly, the Assimilation algorithm was applied to the bounded solution to give an assimilated result. By means of polynomial identification on this result, newer matrix coefficients were obtained to refine the model, as previously mentioned in section 2.3.2. We have adjusted again a value of artificial viscosity \( \nu_A \sim 0.1\nu \) to obtain the final solution. For this case, the method gave the best result with one iteration.

In order to exhibit these results, the phase portraits of the temporal coefficients are drawn on Figure 8. The two principal modes are perfectly recovered during the whole sequence. We can see that the third component of the assimilated curve moves slightly away from the realizations trajectory. This is due to the fact that the third mode is computed with less precision, as appears in Figure 1b, where more symmetric and organized small structures were expected. Nevertheless, in this case, a strong energy accumulation on the first two modes relativizes this comeback.
Figure 7: Equation solution in solid line and original data in symbols.  
a) POD estimation by polynomial identification.  
b) POD Assimilation result.  
c) Polynomial Identification on assimilated result.
Figure 8: I. Trajectories of the two principal modes coefficients. II. Trajectories of the three principal modes coefficients. a) 1000 Observations. b) First POD estimation with damping for 1000 images. c) POD-Assimilation result for 300 images. d) ODE solution from POD-Assimilation result for 1000 images.

2.3.4 Convergence analysis

The convergence of the assimilation process is experimentally illustrated in Figure 9. Both the mean square discrepancy between the estimation and the observation and the convergence criterion defined in (36) can be computed. In the assimilation method, each iteration enhances the results until the chosen criterion is met. We can see that the $\epsilon = 10^{-7}$ criterion is a good choice, since the iterations could not make the criterion exceed the value $10^{-8}$. There is a minimum error due to the observation noise that could not be cancelled out. Indeed, the
dynamical equation does not allow discontinuities in the solution, and it can be noticed that the observed modes \( s \geq 2 \) are more erratic (Figure 6b).

![Figure 9: Convergence study: a) Evolution of the mean square discrepancy between estimation and observations. b) Evolution of the convergence criterion.](image)

### 2.3.5 Flow reconstruction analysis

It is remarkable that with \( s = 6 \) modes, the 94% of the fluctuating kinetic energy is conserved and the LODS produces vorticity snapshots that are in good agreement with the real flow. It is evident that in this case the POD reduced order model can improve the noisy measures. Even more, POD-Assimilated model exhibits coherent structures closer to a validation case coming from a Direct Numerical Simulation (DNS). This DNS of 2D cylinder flow was conducted for \( \text{Re}=125 \), in a domain \( L_x/D = 19 \) by \( L_y/D = 12 \) with a grid resolution of \( 685 \times 433 \). This assures that the grid step \( dx \leq \eta \) where \( \eta \) is Kolmogorov dissipative scale, \( \eta \sim \frac{D}{\text{Re}^{3/4}} \). Therefore, every scale of the flow is resolved and the discretization scheme verifies the necessary stability conditions. The code is highly reliable in this regime and its performance has been reported previously [10]. The Strouhal numbers \( St = \frac{fD}{U_\infty} \sim 0.2 \) issued from simulation (0.1787) and experiments (0.1830) are in good agreement. To evaluate them, we have extracted the dominating frequencies of the lift coefficient from DNS, and the principal frequency of the firsts modes from POD.
3 Conclusions

This document describes the source code allowing to construct a low order dynamical system from POD decomposition. The method used, based on optimal control theory, is described in the first part of the document. A detailed description of the corresponding program is then given, as well as an illustration with some computation results.

References


