C. R. Mecanique 342 (2014) 410–416

Contents lists available at ScienceDirect

Comptes Rendus Mecanique

www.sciencedirect.com

Separation flow control

Model reduction using Dynamic Mode Decomposition

Réduction de modèle par décomposition en modes dynamiques

Gilles Tissot *, Laurent Cordier, Nicolas Benard, Bernd R. Noack

PPRIME Institute, CEAT, 43, route de l'Aérodrome, 86000 Poitiers, France

ARTICLE INFO

Article history:
Received 6 September 2013
Accepted 2 December 2013
Available online 10 June 2014

Keywords:
Dynamic Mode Decomposition
Reduced-order model
Cylinder wake

Mots-clés :
Décomposition en modes dynamiques
Réduction de modèle
Sillage de cylindre

ABSTRACT

Dynamic Mode Decomposition (DMD) is a recent post-processing technique that extracts from snapshots dynamic relevant information for the flow. Without explicit knowledge of the dynamical operator, the DMD algorithm determines eigenvalues and eigenvectors of an approximate linear model. The ability of DMD to extract dynamically relevant features of the flow predispose it for building a representative reduced-order subspace from the data and for deriving a reduced-order model. The use of the DMD for reduced-order modelling will be investigated in this paper on experimental flow data of a cylinder wake.

RÉSUMÉ

La décomposition en modes dynamiques (DMD) est une technique récente de post-traitement qui extrait des informations liées à la dynamique de l'écoulement à partir d'une séquence de snapshots. Sans connaissance explicite de l'opérateur dynamique, l'algorithme DMD détermine les valeurs et vecteurs propres d'un modèle linéaire approché. La capacité de la DMD à extraire des structures pertinentes en termes de dynamique de l'écoulement la prédispose à la construction, à partir de données, d'un sous-espace de dimension réduite représentatif et au développement d'un modèle de dimension réduite. L'utilisation de la DMD pour la réduction de modèle sera étudiée dans ce papier sur des données expérimentales d'un écoulement de sillage autour d'un cylindre.

1. Introduction

For a turbulent flow, the number of active degrees of freedom is so important that a preliminary step of model reduction is often necessary for having a chance to understand the flow physics or to derive a control strategy. The general objective of model reduction is to extract first, from physical insights or mathematical tools, the building blocks—called modes—that play a dominant role in terms of modelling, and then to derive a dynamical model for the time evolution of the system. As we will see in the following, these two steps can be considered independently or jointly depending on the technique

* Corresponding author.
E-mail addresses: Gilles.Tissot@univ-poitiers.fr (G. Tissot), Laurent.Cordier@univ-poitiers.fr (L. Cordier), Nicolas.Benard@univ-poitiers.fr (N. Benard), Bernd.Noack@univ-poitiers.fr (B.R. Noack).

http://dx.doi.org/10.1016/j.crme.2013.12.011
1631-0721/

used for extracting the modes. From this perspective, the idea of model-based flow control is to exploit the knowledge of a dynamical model for reacting in real-time to the modifications encountered by the flow. Reduced-order models (ROMs) are then well adapted for developing an efficient control strategy. However, finding the appropriate basis for representing the flow in a low-dimensional space is strongly related to a given objective. Indeed, it is somewhat different for a flow to understand the instability mechanisms, to educe the coherent structures mainly responsible for the energy or to represent the non-linear dynamics. Moreover, different strategies can be used to determine the time evolution propagator of the flow states. The dynamical evolution can be intrinsically linked to the method used for determining the modes, or can be classically obtained by Galerkin projection of the governing equations onto the set of modes defining the low-dimensional subspace.

Reduced-order models based on Proper Orthogonal Decomposition (POD) are the most commonly used as the POD modes are optimal in terms of energy content. Despite this property, POD-based ROM is well known to be inaccurate, essentially due to truncation errors, if the model is not improved. Since the energy content is important but is not sufficient in general to catch the dynamical behaviour, we are focusing in this paper on a procedure recently introduced by Schmid [1] called Dynamic Mode Decomposition (DMD). This algorithm was proposed as a method able to extract dynamically relevant flow features from time-resolved experimental or numerical data. It generalises the global stability modes and approximates the eigen-elements of the Koopman operator [2,3]. Our objective is to derive a ROM which will inherit the good dynamical properties of the projection basis.

In Section 2.1, the classical DMD algorithm is presented, followed in Section 2.2 by a discussion of the crucial point of modes' selection. In Section 3, the optimized DMD algorithm is then introduced and an improvement of the original algorithm based on the use of a gradient method is shortly described. Section 4 then presents two ways of predicting the temporal behaviour of the flow outside the time horizon of the snapshots, and ends with the introduction of a data assimilation formalism for combining the two approaches. Finally, in Section 5, we present results obtained on PIV data for a cylinder wake in turbulent regime.

2. Dynamic mode decomposition

2.1. General description

The data is represented in the form of a snapshot sequence, given by a matrix V_1^N defined as

$$V_1^N = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N) \in \mathbb{R}^{N_x \times N} \tag{1}$$

where $\boldsymbol{v}_i$ is the ith snapshot. In (1), the subscript 1 denotes the first member of the sequence, while the superscript N denotes the last entry in the sequence. Here, we consider the temporal framework of DMD, and assume that the snapshots are separated by a constant sampling time Δt. The DMD algorithm is built on two main assumptions. The first hypothesis is that there exists a linear operator $\mathcal{A}$ to step forward in time the snapshots. Since V_1^N is finite-dimensional, this operator is written as a matrix $A \in \mathbb{R}^{N_x \times N_x}$ such that

$$\boldsymbol{v}_{i+1} = A\boldsymbol{v}_i, \quad \text{for } i = 1, \cdots, N-1 \tag{2}$$

It follows that the subspace spanned by the data set $V_1^N = (\boldsymbol{v}_1, A\boldsymbol{v}_1, \ldots, A^{N-1}\boldsymbol{v}_1)$ corresponds to the Nth Krylov subspace $\mathcal{K}_N(A, \boldsymbol{v}_1)$ generated by A from $\boldsymbol{v}_1$. The goal of DMD is to determine eigenvalues and eigenvectors of A but without first determining A. As such, DMD can be interpreted as an extension of the classical Arnoldi algorithm used to determine eigen-elements of large size problems [4]. In the Arnoldi algorithm, the knowledge of A is exploited to determine an orthonormal basis for the projection subspace of the Rayleigh–Ritz procedure. In the DMD algorithm, the basis of the projection subspace is determined with a "matrix-free" point of view by considering that only snapshots obtained from a time-stepper are available. The matrix A is no longer necessary but the price is an ill-conditioning of the procedure. Since we are interested by eigenvalues of A, we are searching for the spectrum of a similarity matrix of the Galerkin projection of A onto the subspace spanned by the snapshots. When the number of snapshots of the sequence V_1^N increased, it is reasonable to assume that, beyond a given number of snapshots, $\boldsymbol{v}_i$ becomes linearly dependent. The second hypothesis is then to consider that the Nth iterate writes as a linear combination of the previous iterates, i.e.

$$\begin{aligned} \boldsymbol{v}_N &= c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_{N-1}\boldsymbol{v}_{N-1} + \boldsymbol{r} \\ &= V_1^{N-1}\boldsymbol{c} + \boldsymbol{r} \end{aligned} \tag{3}$$

where $\boldsymbol{c}^{\mathrm{T}} = (c_1, c_2, \cdots, c_{N-1})^{\mathrm{T}}$ and $\boldsymbol{r} \in \mathbb{R}^{N_x}$ is the residual vector. A straightforward calculation leads to

$$AV_1^{N-1} = V_2^N = V_1^{N-1}C + \boldsymbol{r}\boldsymbol{e}_{N-1}^{\mathrm{T}} \tag{4}$$

where $\boldsymbol{e}_i$ is the ith Euclidean unitary vector of length $(N-1)$ and C the companion matrix associated with $\boldsymbol{c}$. C is uniquely defined by the coefficients c_i, which may be found using the Moore–Penrose pseudo-inverse of V_1^{N-1}, i.e. $\boldsymbol{c} = (V_1^{N-1})^+\boldsymbol{v}_N = ((V_1^{N-1})^* V_1^{N-1})^{-1}(V_1^{N-1})^*\boldsymbol{v}_N$.