Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Journal of Computational Physics 207 (2005) 568–587

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Equation-free/Galerkin-free POD-assisted computation of incompressible flows

Sirod Sirisup [a], George Em Karniadakis [a,*], Dongbin Xiu [b], Ioannis G. Kevrekidis [b]

[a] *Division of Applied Mathematics, Brown University, 182 George St., Box F, Providence, RI 02912, USA*

[b] *Department of Chemical Engineering, Princeton University, Princeton, NJ 08544, USA*

Received 2 September 2004; received in revised form 28 January 2005; accepted 28 January 2005
Available online 23 March 2005

Abstract

We present a Galerkin-free, proper orthogonal decomposition (POD)-assisted computational methodology for numerical simulations of the long-term dynamics of the incompressible Navier–Stokes equations. The approach is based on the "equation-free" framework: we use short, appropriate initialized bursts of full direct numerical simulations (DNS) of the Navier–Stokes equations to observe, estimate, and accelerate, through "projective integration", the evolution of the flow dynamics. The main assumption is that the long-term dynamics of the flow lie on a low-dimensional, attracting, and invariant manifold, which can be *parametrized*, not necessarily spanned, by a few POD basis functions. We start with a discussion of the consistency and accuracy of the approach, and then illustrate it through numerical examples: two-dimensional periodic and quasi-periodic flows past a circular cylinder. We demonstrate that the approach can successfully resolve complex flow dynamics at a reduced computational cost and that it can capture the long-term asymptotic state of the flow in cases where traditional Galerkin-POD models fail. The approach trades the overhead involved in developing POD-Galerkin and POD-nonlinear Galerkin codes, for the repeated (yet short, and on demand) use of an existing full DNS simulator. Moreover, since in this approach the POD modes are used to *observe* rather than *span* the true system dynamics, the computation is much less sensitive than POD-Galerkin to values of the system parameters (e.g., the Reynolds number) and the particular simulation data ensemble used to obtain the POD basis functions.

© 2005 Elsevier Inc. All rights reserved.

[*] Corresponding author. Tel.: +401 863 1217; fax: +401 863 3369.
E-mail addresses: gk@cfm.brown.edu (S. Sirisup), gk@dam.brown.edu (G.E. Karniadakis).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.01.024

1. Introduction

The "traditional" proper orthogonal decomposition (POD) is a methodology that first identifies the few most energetic modes in a time-dependent system, and subsequently provides a means of obtaining a low-dimensional description of the system's dynamics [4]. The *method of snapshots*, first proposed in [40] for flow systems, is a particularly effective and easy to implement approach for obtaining POD basis sets from an ensemble of signals. POD has been successfully implemented in conjunction with experimental (e.g. [1,7,9,18,19]) as well as with numerical studies (e.g. [2,6,8,27,32,38,40]) in thermal convection, shear layers, cavity flows and external flows, to mention just a few. In the aforementioned works, low-dimensional system dynamical systems are obtained directly from the Galerkin projection of the governing equations on the empirical basis set (the POD modes). However, it is well known that reduced systems resulting from truncated Galerkin projections may result, after long-term integration, in *spurious* aymptotic states (e.g. [11,31,39]). In [31] it is argued that this is due to the fact that the manifold spanned by the POD modes is, in general, not an invariant manifold of the Navier–Stokes operator. A technique based on *spectral viscosity convolution operator* was proposed in [39] to correct the long-term behavior of Galerkin POD models.

In this paper, we propose a Galerkin-free POD-assisted modeling methodology, which employs "equation-free" [41,25,24] projective integration [13,14,33]. The equation-free framework is designed for the efficient coarse-grained computational study of complex, multi-scale problems. The basic idea is to operate at two levels: (a) design and perform short-time numerical experiments with "the best available" microscopic model, and subsequently (b) use the numerical results of such microscopic computations to estimate quantities (residuals, action of Jacobians) required in numerical computations of the (unavailable) macroscopic equations for the coarse-grained system behavior [25,24]. Thus the closures required to obtain explicit macroscopic equations are estimated *on demand*; numerical analysis tasks (such as integration, solution of linear and nonlinear equations and eigenanalysis) are performed by acting on the microscopic simulation directly (equation-free) taking extensive advantage of matrix-free iterative linear algebra. This framework has been applied to a variety of problems, ranging from bifurcation analysis of complex systems to homogenization of random media [13,17,25,26,29,30,34,41].

In this paper we illustrate the use of the equation-free approach to incompressible Navier–Stokes simulations. In this context, the DNS simulator plays the role of the "detailed, microscopic" model; the coarse-grained dynamics are the long-term dynamics of the DNS, which are assumed to lie on a low-dimensional, attracting manifold. This manifold is *parametrized* by a relatively small number of variables (in our case we use the leading POD modes to parametrize the manifold). The "unavailable coarse-grained" equations are the dynamic equations for the evolution of the coefficients of the projection of the full DNS solution *on this manifold* on our observation variables (the leading POD modes). The normal procedure for obtaining these equations would be to perform a Galerkin projection of the Navier–Stokes on the POD basis; truncating this Galerkin projection would give an *approximation* of these equations. What we are after is not such a truncated POD-Galerkin; we want to observe the projection – on the hyperplane spanned by the first few POD modes – of the evolution of the true solution *on the slow manifold*. This manifold can be thought of as a graph of function from the first few ("governing" or "master") POD modes to the remaining ones. If this function were explicitly available, one could substitute the components of the long-term solution on higher-order POD modes as functions of the low-order ones in a POD Galerkin expansion, and obtain the "correct" low-dimensional *nonlinear* POD-Galerkin model. Our equation-free approach tries to solve this model without deriving it in closed form. Full DNS simulations are used to estimate the right-hand side (RHS) of this "correct" nonlinear Galerkin model on demand; the particular equation-free acceleration technique we will illustrate is *projective integration*. Since the equations of change are *estimated* rather than obtained from a Galerkin formalism, we consider this a *Galerkin-free* procedure.

The paper is organized as follows: In Section 2 we present the DNS/POD formulation for incompressible flow dynamics. In Section 3 the algorithm for the equation-free/Galerkin-free POD model is presented, and the consistency and accuracy of the method are demonstrated. Numerical results follow in Section 4, where two-dimensional flows past a circular cylinder are used as the illustrative prototype. We first verify the error estimates from Section 3 through short-time integration, and subsequently present long-term integration results to verify that the new model is able to capture the correct asymptotic state. An additional, dynamically more complicated quasi-periodic flow example is also presented, showing that multiple flow frequencies can be successfully resolved. We summarize the results in Section 5, with a discussion of open issues.

2. Computational formulation for incompressible flows

2.1. Navier–Stokes equations and discretization

We consider the incompressible Navier–Stokes equations

$$\nabla \cdot \mathbf{v} = 0, \tag{1}$$

$$\frac{\partial \mathbf{v}}{\partial \mathrm{t}} + (\mathbf{v} \cdot \nabla)\mathbf{v} = -\nabla \mathrm{p} + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{v}, \tag{2}$$

where $\mathbf{v} \equiv \mathbf{v}(t,\mathbf{x})$ is the velocity fields and $p \equiv p(t,\mathbf{x})$ the pressure, for $\mathbf{x} \in \Omega \subset R^d$ ($d = 1,2,3$) and $t \in R^+$. The equations are properly scaled by a characteristic velocity U and a characteristic length L, and the Reynolds number is defined as $Re = UL/v$, where v is the kinematic viscosity. We assume that appropriate boundary conditions are imposed on the boundary $\partial\Omega \subset R^{d-1}$, together with initial condition $\mathbf{v}_0 \equiv \mathbf{v}(0,\mathbf{x}) \in \Omega$. We also define an inner product $(f,g) = \int_\Omega f(\mathbf{x})g(\mathbf{x})\mathrm{d}\mathbf{x}$, and a norm $\|f\| = (f,f)^{1/2}$.

Eqs. (1) and (2) define an evolution process of the velocity field

$$\frac{\partial \mathbf{v}}{\partial \mathrm{t}}(\mathrm{t},\mathbf{x}) = \mathbf{f}(\mathrm{t};\mathbf{v}(\mathrm{t},\mathbf{x})) \tag{3}$$

characterized by a solution operator $\{\mathbf{s}(t)\}$, which forms a semigroup $\mathbf{v}(t,\cdot) = \mathbf{s}(t)\mathbf{v}(0,\cdot)$, constrained by the pressure field to satisfy the divergence-free condition (1).

In order to solve the Navier–Stokes equations, we introduce a computational mesh $\mathrm{X}_\mathrm{h}^\mathrm{d}$ that discretizes $\Omega \subset R^d$, where $h > 0$ is the maximum mesh spacing. We adopt a numerical scheme that is consistent, accurate and stable. Such a scheme defines a discrete evolution process for $\mathbf{u}(t,\mathbf{x})$, the numerical solution of $\mathbf{v}(t,\mathbf{x})$, i.e.,

$$\mathbf{u}(\mathrm{t}^{\mathrm{n}+1},\cdot) = \mathbf{u}(\mathrm{t}^\mathrm{n},\cdot) + \delta\mathrm{t} \cdot \Phi(\mathrm{t}^\mathrm{n};\delta\mathrm{t},\mathrm{h},\mathbf{u}^\mathrm{n},\mathbf{u}^{\mathrm{n}+1},\ldots), \tag{4}$$

where δt is the time step such that $t^{n+1} = t^n + \delta t$, and the function $\Phi(\delta t,h;\cdot)$ is the incremental function. Hereafter, we adopt the notation $g^n \equiv g(t^n)$. Consistency of the discrete scheme requires that

$$\lim_{\delta\mathrm{t},\mathrm{h}\to 0} \Phi(\mathrm{t}^\mathrm{n};\delta\mathrm{t},\mathrm{h},\mathbf{v}^\mathrm{n},\mathbf{v}^{\mathrm{n}+1},\ldots) = \mathbf{f}(\mathrm{t}^\mathrm{n};\mathbf{v}^\mathrm{n}). \tag{5}$$

We also assume that the *local truncation error* (LTE) of such scheme is of the order $\mathrm{O}(\delta t^p, h^q)$, i.e.,

$$\mathbf{v}^{\mathrm{n}+1}(\mathbf{x}) = \mathbf{v}^\mathrm{n}(\mathbf{x}) + \delta\mathrm{t} \cdot \Phi(\mathrm{t}^\mathrm{n};\delta\mathrm{t},\mathrm{h},\mathbf{v}^\mathrm{n},\mathbf{v}^{\mathrm{n}+1},\ldots) + \delta\mathrm{t} \cdot \epsilon_\mathrm{f}, \tag{6}$$

where

$$\epsilon_\mathrm{f} \sim \mathrm{O}(\delta\mathrm{t}^\mathrm{p},\mathrm{h}^\mathrm{q}), \quad \mathrm{p},\mathrm{q} \geqslant 1.$$

2.2. Proper orthogonal decomposition basics

The POD procedure extracts an empirical orthogonal basis using a modal decomposition from an ensemble of signals (computational or experimental). It is a linear procedure which produces a useful reduced basis (which, under appropriate conditions, is an optimal one). In the POD framework one can represent evolving flow fields in the form

$$\mathbf{u}(t,\mathbf{x}) = \sum_k a_k(t)\boldsymbol{\phi}_k(\mathbf{x}), \tag{7}$$

where $\{\boldsymbol{\phi}_k(\mathbf{x})\}_{k=0}^{\infty}$ is the basis extracted from the eigenvalue problem for the temporal modes $\{a_k(t)\}_{k=0}^{\infty}$,

$$\int_A C(t,t')a_k(t')dt' = \lambda_k a_k(t), \quad t \in A, \ \forall k, \tag{8}$$

and A is a specified time interval. Here $C(t,t')$ is the correlation function defined as

$$C(t,t') = \int_\Omega \mathbf{u}(t,\mathbf{x}) \cdot \mathbf{u}(t',\mathbf{x})d\mathbf{x}. \tag{9}$$

The POD basis in space is determined by

$$\boldsymbol{\phi}_k(\mathbf{x}) = \int_A a_k(t)\mathbf{u}(t,\mathbf{x})dt \quad \forall k. \tag{10}$$

These basis functions are normalized so that $\|\boldsymbol{\phi}_i\| = 1$. The non-negative definiteness of the velocity correlation function (9) assures that $\lambda \geqslant 0$ and we order the eigenvalues by $\lambda_i \geqslant \lambda_{i+1}$. After determining the POD basis, the POD expansion coefficients can be recovered via $a_k(t) = (\mathbf{u}(t,\mathbf{x}), \boldsymbol{\phi}_k(\mathbf{x}))$. By denoting $\mathbf{a}(t) = \{a_k(t)\}$, we define a *restriction operator* $\mathscr{P}$ such that

$$\mathbf{a}(t) = \mathscr{P}\mathbf{u}(t,\mathbf{x}) \equiv \{(\mathbf{u}(t,\mathbf{x}), \boldsymbol{\phi}_k(\mathbf{x})), \ t \in A, \ \forall k\}, \tag{11}$$

and a *lifting operator* $\mathscr{Q}$ such that

$$\mathbf{u}(t,\mathbf{x}) = \mathscr{Q}\mathbf{a}(t) \equiv \sum_k a_k(t)\boldsymbol{\phi}_k(\mathbf{x}), \quad t \in R^+. \tag{12}$$

Obviously, both $\mathscr{P}$ and $\mathscr{Q}$ are linear operators and $\mathscr{P}\mathscr{Q}\mathbf{a} = \mathbf{a}$. The evolution of the POD coefficients $\mathbf{a}(t)$ is governed by a process

$$\frac{d\mathbf{a}}{dt}(t) = \mathbf{g}(t;\mathbf{a}(t)), \tag{13}$$

where the explicit form of the RHS terms is unknown. In the standard truncated POD-Galerkin procedure, the RHS terms are derived from the Navier–Stokes equations via a Galerkin approach, resulting in a (possibly large) set of coupled ODEs.

Let us denote the finite-term POD expansion as

$$\mathbf{u}_K(t,\mathbf{x}) = \sum_{k=1}^{K} a_k(t)\boldsymbol{\phi}_k(\mathbf{x}), \tag{14}$$

and correspondingly, the *truncated restriction* and *truncated lifting* operators as $\mathscr{P}_K$ and $\mathscr{Q}_K$, respectively. In other words, $\mathbf{u}_K = \mathscr{I}_K\mathbf{u} \equiv \mathscr{Q}_K\mathscr{P}_K\mathbf{u}$. The convergence of the POD expansion is assumed to be

$$\|\mathbf{u} - \mathbf{u}_K\| \to K^{-\gamma} \quad \text{as } K \to \infty, \tag{15}$$

where the real number $\gamma > 0$ quantifies the convergence rate. We remark that the theoretical estimate for such rate is hard to obtain, and it depends strongly on the correlation function $C(t, t')$ and the smoothness of the target function $\mathbf{u}$.

3. The equation-free/Galerkin-free POD model

Experience with "traditional" POD-Galerkin truncations show that – depending on the application, the parameter regime and the original data set – they can be extremely successful and that they can also fail miserably (especially in predicting long-term dynamics). There are, of course, technical difficulties involved in writing down (for simple nonlinearities) or approximating via quadrature the RHS of the POD-Galerkin truncation. Beyond these, however, an important reason for this potential failure is that the component of the solution on the "higher" neglected POD modes is *not* zero: the hyperplane spanned by the first few POD modes *does not coincide* with a true invariant slow manifold of the equation we want to solve. Using enough POD modes to accurately span this "tail" of the solution and its effect on the low-POD-mode dynamics gives rise to increasingly larger truncations, at cross-purposes with model reduction and simplification. The main idea here is that a few POD modes can be used to *parametrize* the (assumed) low-dimensional attracting slow manifold on which the long-term flow dynamics lie *as opposed to* spanning it. Assuming a separation of time scales between a few "slow, master" POD modes and the remaining "fast, slave" ones, after a transient has approached this manifold the components of the solution in the higher-order POD modes become functions of the components in the lower-order ones. Normally, one would explicitly approximate this function and use it (substitute it) in the low-POD mode equations as a "closure": a way to obtain the effect of the high-POD modes on the dynamics of the low-POD ones. Such a closed set of equations for the low modes is a *nonlinear* Galerkin, as opposed to the straight, linear Galerkin truncation. Our approach uses bursts of full DNS to *estimate* the right hand sides of these "nonlinear Galerkin" ODEs on demand, and accelerate their numerical integration, without approximating them in closed form. In this paper we will illustrate the equation-free implementation of a particular continuum numerical task: numerical integration, which in an equation-free context is called "projective integration"; most other continuum numerical tasks (fixed point computation, linearized stability analysis, controller design and optimization) can also in principle be implemented "equation-free" [25].

Equation-free single step projective integration, starting at $t = t^n$ and ending at $t = t^{n+1}$, consists, in general, of the following main components:

- $n_f \geqslant 1$ steps of fine-scale computation (reported at time step δt where we define $\Delta \mathrm{t_f} = \mathrm{n_f}\,\delta \mathrm{t}$ and $\mathrm{t_c^n} = \mathrm{t^n} + \Delta \mathrm{t_f}$);
- *restriction* to coarse grained variables and *estimation* of the time-derivatives of their evolution;
- one step of coarse-grained projective integration with step size Δt_c;
- and *lifting* from the projected values of the coarse grained variables to *consistent* fine-scale initial conditions.

The coarse step $\Delta t_c \geqslant \delta t$ will be usually chosen to be $\Delta t_c = n_c \delta t$ where $n_c \geqslant 1$. The global time step is $\Delta t = t^{n+1} - t^n = \Delta t_f + \Delta t_c = (n_f + n_c)\delta t$. Fig. 1 is a graphical illustration of the notation. Variations of the approach, including cases where the "fine scale" simulator is a stochastic (kinetic Monte Carlo based) or a molecular dynamics-based one can be found in [35,24,33].

In our case, the "inner, fine scale" simulator is the fully resolved DNS discretization. The "coarse-grained" model is the unavailable-in-closed-form nonlinear Galerkin set of ODEs describing the evolution of the long-term flow dynamics on the slow manifold; the dynamics are *observed on* the first few low-POD modes which parametrize this manifold.

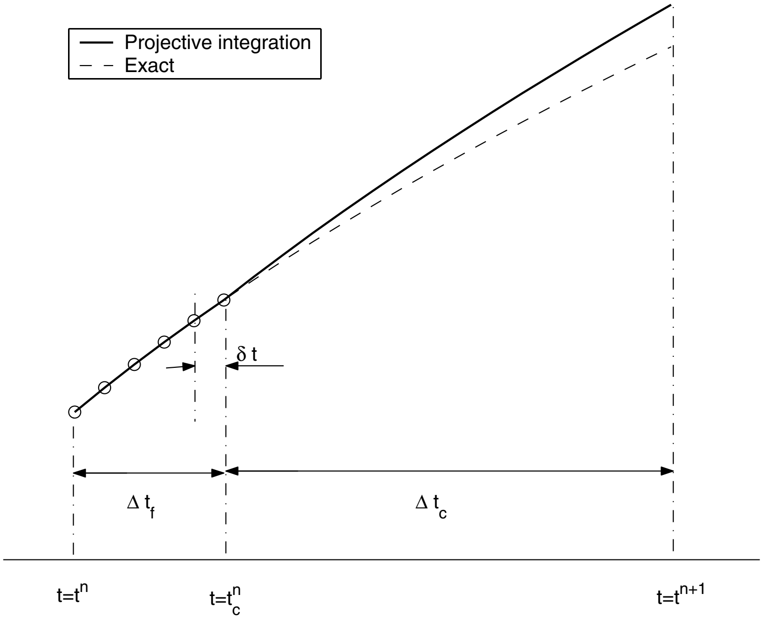


Fig. 1. Sketch of a projective integration over one time step.

Specifically, our POD-assisted projective integration consists of the following steps: Given $\mathbf{a}^n \equiv \mathbf{a}(t^n)$,

1. *Lifting:* At $t = t^n$, generate the flow field $\mathbf{u}_K(t^n, \mathbf{x}) = \mathscr{Q}_K \mathbf{a}(t^n)$.
2. *Fine-scale computation:* Resolve the Navier–Stokes equations (1) and (2) for a short period of time, i.e., $\mathbf{v}(t, \cdot) = \mathbf{s}(t)\mathbf{v}(t^n, \cdot)$ for $t^n \leqslant t \leqslant t^n_c = t^n + n_f \delta t$. Such computation is conducted via scheme (4) with a small time step δt. Here $n_f \geqslant 1$ such that $\Delta t_f = n_f \delta t \sim t_R \ll t_M$, where t_R is the local relaxation time (approach to the slow manifold) which is assumed to be much shorter than t_M, the typical coarse-grained flow time scales (13).
3. *Restriction:* Evaluate the POD coefficients $\mathbf{a}(t) = \mathscr{P}_K \mathbf{u}(t, \cdot)$ for $t^n \leqslant t \leqslant t^n_c$, and estimate the time derivative $d\mathbf{a}/dt$ at $t = t^n_c$.
4. *Projective integration:* Integrate Eq. (13) to t^{n+1} by standard ODE techniques to obtain $\mathbf{a}(t^{n+1})$. The time step here is $\Delta t_c \equiv n_c \delta t = t^{n+1} - t^n_c$, where $n_c \geqslant n_f \geqslant 1$.
5. Return to step 1 until the final integration time is reached.

We now present a detailed discussion of each of the steps.

3.1. Full DNS computation

Any conventional numerical method satisfying (4)–(6) can be employed to resolve equations (1) and (2). In this paper, we employ the high-order spectral/*hp* element method, a well-developed method that combines the high accuracy of spectral methods and the flexibility in handling complex geometry of finite element methods [21]. The spatial discretization is sufficiently fine to guarantee resolution independent solutions. A high-order splitting scheme is employed to integrate the incompressible Navier–Stokes equations [20]; we typically chose the third-order scheme in time, i.e., $p = 3$ in (6).

3.2. Restriction and lifting

We employ the *snapshot method formulation* to extract the POD bases. In this method, the time average defined in (8) is replaced by an average of an ensemble of flow field snapshots within a time interval J. Specifically, the vectors $\{\boldsymbol{\phi}_k(\mathbf{x})\}$ in (10) are determined by

$$\boldsymbol{\phi}_k(\mathbf{x}) = \sum_{i=1}^{N} a_k(t_i)\mathbf{u}(t_i, \mathbf{x}) \quad \forall k, \tag{16}$$

where $\{t_i\}_{i=1}^{N} \subset J$ denotes the ensemble of N number of snapshots within the time interval J and $\{a_k\}$ are obtained by solving (8).

Once the POD basis functions $\{\boldsymbol{\phi}_k(\mathbf{x})\}$ are determined from (16), we can restrict a given flow field $\mathbf{u}(t, \mathbf{x})$ for any given t to obtain its POD coefficients $\{a_k\}$, via (11). These coefficients will be further integrated in time by the "projective integration" technique, described in the following section. The lifting procedure is the reverse, i.e., for a given set of numerically computed POD coefficients $\{a_k(t)\}$ at a time t, we can construct the corresponding flow field by using (12).

3.3. Projective integration

As discussed above, the RHS $\mathbf{g}(t; \mathbf{a}(t))$ of equations (13) can be obtained from the Navier–Stokes equations through a Galerkin projection. Such a procedure may result in rather complicated forms and often suffers from spurious long-term dynamics (see [11,39]). Our "equation-free" approach, which does not require the explicit form of the RHS of (13), can circumvent these difficulties by using "just enough" full DNS. Having already performed a short DNS simulation run, our procedure is as follows:

- Obtain the POD variable time series $\mathbf{a}(t) = \mathscr{P}\mathbf{u}(t, \mathbf{x})$ for $t^n \leqslant t \leqslant t_c^n = t^n + n_f\,\delta t$.
- Approximate the RHS of (13) at $t = t_c^n$, here through
$$\mathbf{g}(t_c^n) = \sum_{j=0}^{n_e} \alpha_j \mathbf{a}(t_j) = \frac{d\mathbf{a}}{dt}(t_c^n) + O(\delta t^{J_f}), \tag{17}$$
where $1 \leqslant n_e \leqslant n_f$, $t_j = t_c^n - j \cdot \delta t$, and J_f denotes the order of the approximation. $\{\alpha_j\}_{j=0}^{n_e}$ is a set of consistent coefficients such that $\sum \alpha_j f(t_j) = df/dt(t_c^n) + O(\delta t^{J_f})$.
- Once the RHS of (13) is estimated numerically, one can effectively integrate it via standard ODE solvers. For instance, given a coarse time step $\Delta t_c \equiv n_c \delta t$ where $n_c \geqslant 1$, such that $t^{n+1} = t_c^n + \Delta t_c = t^n + (n_f + n_c)\delta t$, the single step forward Euler projective integrator takes the form
$$\mathbf{a}(t^{n+1}) = \mathbf{a}(t_c^n) + \Delta t_c \cdot \mathbf{g}(t_c^n) + O(\Delta t_c^2). \tag{18}$$

Other integration schemes – possibly implicit ones – can be used for higher-order accuracy and/or better stability properties. For instance, we can employ the following scheme:

$$\mathbf{a}(t^{n+1}) = \mathbf{a}(t_c^n) + \sum_{k=1}^{J_c} \frac{(\Delta t_c)^k}{k!} \frac{\partial^{(k-1)}}{\partial t^{k-1}} \mathbf{g}(t_c^n) + O(\Delta t_c^{J_c+1}). \tag{19}$$

The higher-order temporal derivatives of $\mathbf{g}(t)$ are approximated in a way similar to (17). Note that (19) is a high-order single-step method. For projective integrations, multi-step methods such as Adam-Bashforth type integrators have been studied (cf. [33]), and they can be used here as well. Remarkably, for problems with separation of time scales, estimating the RHS from accurate results of detailed integration may result in stability for much longer explicit projective steps (commensurate with the slow dynamics [13]). In the following analysis, we will focus on single-step methods (19).

3.4. Consistency and accuracy

Here we present an analysis of the consistency and accuracy of the equation-free POD-assisted method. For clarity, we suppress the symbol **x** and use the first-order Euler forward projective integrator in (18). By construction, the solution after one global time step Δt can be written as

$$\begin{aligned}\mathbf{u}_K^{n+1} &\equiv \mathbf{u}_K(t^{n+1}) = \mathcal{Q}_K\mathbf{a}(t^{n+1}) = \mathcal{Q}_K[\mathbf{a}(t_c^n) + \Delta t_c\cdot\mathbf{g}(t_c^n) + O(\Delta t_c^2)]\\ &= \mathcal{Q}_K[\mathbf{a}(t_c^n) + \Delta t_c\sum_j \alpha_j\mathbf{a}(t_j) + O(\Delta t_c^2)] = \mathcal{Q}_K\mathcal{P}_K[\mathbf{u}(t_c^n) + \Delta t_c\sum_j\alpha_j\mathbf{u}(t_j) + O(\Delta t_c^2)]\\ &= \mathcal{I}_K\left[\mathbf{u}(t^n) + \delta t\cdot\sum_{m=1}^{n_f}\Phi(t_m^n;\mathbf{u}) + \Delta t_c\frac{\partial\mathbf{u}}{\partial t}(t_c^n) + O(\delta t^{J_f}) + O(\Delta t_c^2)\right] = \mathbf{u}_K^n + \Delta t\Psi(t^n;\mathbf{u},\ldots),\end{aligned} \tag{20}$$

where $t_m^n = t^n + (m-1)\cdot\delta t$ and the *global incremental function* is defined as

$$\Psi(t^n;\mathbf{u},\ldots) = \frac{1}{\Delta t}\mathcal{I}_K\left[\delta t\cdot\sum_{m=1}^{n_f}\Phi(t_m;\mathbf{u}) + \Delta t_c\frac{\partial\mathbf{u}}{\partial t}(t_c^n) + O(\delta t^{J_f}) + O(\Delta t_c^2)\right]. \tag{21}$$

In the limit of $\Delta t\to 0$, $\{t_m^n\}_{m=1}^{n_f}$, $t_c^n\to t^n$ and $\delta t, \Delta t_c\to 0$. Since $\Delta t = n_f\cdot\delta t + \Delta t_c$, we have

$$\lim_{\Delta t,h\to 0}\Psi(t^n;\mathbf{v},\ldots) = \frac{1}{\Delta t}\mathcal{I}_K\left[\delta t\cdot n_f\cdot\mathbf{f}(t^n;\mathbf{v^n}) + \Delta t_c\frac{\partial\mathbf{v}}{\partial t}(t^n) + O(\delta t^{J_f}) + O(\Delta t_c^2)\right] = \mathcal{I}_K[\mathbf{f}(t^n;\mathbf{v}^n)], \tag{22}$$

where we have used property (5). Assuming the POD convergence (15), we immediately have the *consistency* of the POD-assisted projective integration method

$$\lim_{\Delta t,h\to 0,K\to\infty}\Psi(t^n;\mathbf{v}) = \mathbf{f}(t^n;\mathbf{v}^n). \tag{23}$$

To obtain the LTE, we start with the exact solution of $\mathbf{v}^n(\mathbf{x})$ at time t^n and evolve the scheme by one global time step Δt to obtain

$$\begin{aligned}\tilde{\mathbf{u}}_K^{n+1} &= \mathbf{v}_K^n + \Delta t\cdot\Psi(t^n;\mathbf{v}) = \mathcal{I}_K\left[\mathbf{v}(t^n) + \delta t\cdot\sum_{m=1}^{n_f}(\Phi(t_m;\mathbf{v}) + \epsilon_f) + \Delta t_c\frac{\partial\mathbf{v}}{\partial t}(t_c^n) + O(\delta t^{J_f}) + O(\Delta t_c^2)\right]\\ &= \mathcal{I}_K\left[\mathbf{v}(t_c^n) + n_f\,\delta t\cdot\epsilon_f + \Delta t_c\frac{\partial\mathbf{v}}{\partial t}(t_c^n) + O(\delta t^{J_f}) + O(\Delta t_c^2)\right]\\ &= \mathbf{v}_K(t^{n+1}) + \mathcal{I}_K[\Delta t_f\cdot\epsilon_f + O(\delta t^{J_f}) + O(\Delta t_c^2)] \equiv \mathbf{v}_K(t^{n+1}) + \tau_K.\end{aligned} \tag{24}$$

Since $O(\Delta t_f)\sim O(\Delta t)$ (in favorable cases, $O(\Delta t_f)\ll O(\Delta t)$) and $O(\Delta t_c)\sim O(\Delta t)$, the LTE measured by the POD variables $\mathbf{v}_K$ is

$$\epsilon_K \equiv \frac{1}{\Delta t}\|\tilde{\mathbf{u}}_K^{n+1} - \mathbf{v}_K(t^{n+1})\| = \frac{1}{\Delta t}\|\tau_K\| \sim \epsilon_f + O(\delta t^{J_f-1},\Delta t_c), \tag{25}$$

where $\epsilon_f\sim O(\delta t^p, h^q)$ is the error of the Navier–Stokes solver.

If we measure the LTE against the flow field **v**, instead of the POD projected field $\mathbf{v}_K$, then

$$\epsilon = \frac{1}{\Delta t}\|\tilde{\mathbf{u}}_K^{n+1} - \mathbf{v}(t^{n+1})\| \leqslant \frac{1}{\Delta t}(\|\tilde{\mathbf{u}}_K^{n+1} - \mathbf{v}_K(t^{n+1})\| + \|\mathbf{v}_K(t^{n+1}) - \mathbf{v}(t^{n+1})\|) \sim \epsilon_K + O\left(\frac{K^{-\gamma}}{\Delta t}\right) \tag{26}$$

where $O(K^{-\gamma})$ is the error of POD projection from (15).

In summary, the truncation error of the POD-assisted projective integration method proposed in this paper, with an integrator of order $O(\Delta t_c^{J_c})$ as in (19), is

$$\epsilon \sim \mathrm{O}\left(\delta t^{p}, h^{q}, \delta t^{J_f - 1}, \Delta t_c^{J_c}, \frac{K^{-\gamma}}{\Delta t}\right). \tag{27}$$

Note that the error depends on Δt (or Δt_c since $\Delta t_c/\Delta t \sim 1$) non-monotonically, when all other parameters are fixed. When the fine computation of Navier–Stokes equations is conducted with high accuracy and $\delta t \ll \Delta t$, the two dominant error contributions stem from $\mathrm{O}(\Delta t_c^{J_c})$ and $\mathrm{O}(K^{-\gamma}/\Delta t)$. These two terms "compete" in opposite ways with respect to the size of Δt, e.g., decreasing Δt will decrease the former term, but amplify the latter one. Clearly, a fast decay of the POD spectrum will result in a dominant error contribution of the term $\mathrm{O}(\Delta t_c^{J_c})$.

4. Numerical results

In this section we present numerical results of the equation-free POD model for incompressible Navier–Stokes equations. The prototype flow considered here is the two-dimensional flow past a circular cylinder with diameter $D = 1$. The Reynolds number is based on D and fixed at $Re = 100$. The computational domain is shown in Fig. 2. Uniform inflow $\mathbf{u} = (1,0)$ is prescribed on the left, top and bottom boundaries. The outflow condition (zero Neumann) is specified on the right exit, and a no-slip condition is enforced on the surface of the cylinder. A history point is placed at $x = 2.5, y = -0.5$, where we record the time evolution of velocities. A high-resolution spectral/*hp* element method is employed to solve the Navier–Stokes equations [21]. The domain is discretized into 412 triangular elements, where eighth-order Jacobi polynomial basis are used to obtain resolution independent solution. A third-order splitting scheme is employed in time [20], with time step $\delta t = 0.001$. Such fine discretizations in space and time make the errors from Navier–Stokes solution subdominant compared to other error terms in (27), in most cases presented below. Also, we remark that there is no need to conduct the projective integration for the pressure field, because in splitting schemes the pressure serves as a constraint to satisfy the divergence-free condition and is not required upon initialization of the flow velocity field.

4.1. Error estimates verification

At Reynolds number $Re = 100$, the flow becomes periodic with period $T \approx 5.824$. The POD modes $\{\phi(\mathbf{x})\}$ are obtained from direct simulation in the neighborhood of this limit cycle by the method-of-snap-

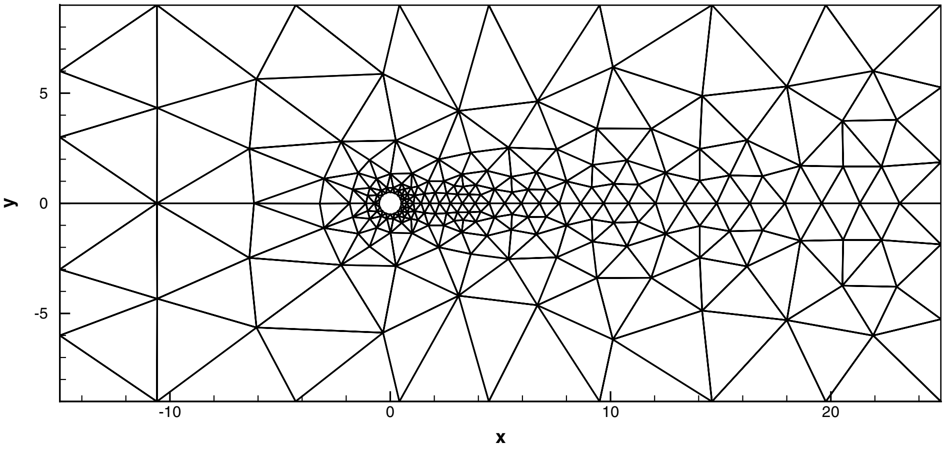


Fig. 2. Computational domain for flow past a circular cylinder.

shots; we typically use 64 snapshots distributed over one oscillation period, and $K = 11$ modes are kept in the finite-term expansion (14). We state once more that the results of POD-Galerkin methods may crucially depend on the particular ensemble used for POD mode extraction.

The error estimate in (27) suggests that the dominant error contribution comes from the coarse integration $(\mathrm{O}(\Delta t_c^{J_c}))$ and the POD approximation error $(\mathrm{O}(K^{-\gamma}/\Delta t))$. The rest of the terms are subdominant as $\Delta t \gg \delta t$. The overall error is non-monotonic with respect to the size of time step Δt. To verify numerically the LTE error estimate (27), we conduct the equation-free POD projective integration, starting from the "exact" flow field $\mathbf{u}(t, \mathbf{x})$, for one global time step Δt, and compute the error of the solution against the "exact" solution at the new time level $\mathbf{u}(t + \Delta t, \mathbf{x})$. Since the "exact" solution $\mathbf{u}(t, \mathbf{x})$ is not available analytically, we use the high resolution DNS solution $\mathbf{v}(t, \mathbf{x})$ to represent it, as the error from DNS is negligible.

Two errors are computed: the error compared to the "exact" flow field $\mathbf{v}$, i.e., $\tau \equiv \|\tilde{\mathbf{u}}_\mathrm{K}^{\mathrm{n}+1} - \mathbf{v}(\mathrm{t}^{\mathrm{n}+1})\|$, and the error compared to the flow field reconstructed from a K-mode POD truncation $\mathbf{v}_K$, i.e., $\tau_\mathrm{K} \equiv \|\tilde{\mathbf{u}}_\mathrm{K}^{\mathrm{n}+1} - \mathbf{v}_\mathrm{K}(\mathrm{t}^{\mathrm{n}+1})\|$. From the error estimates (25) and (27), we have

$$\tau_\mathrm{K} = \Delta \mathrm{t} \cdot \epsilon_\mathrm{K} \sim \mathrm{O}(\Delta \mathrm{t}^{\mathrm{J_c}+1}), \quad \tau = \Delta \mathrm{t} \cdot \epsilon \sim \mathrm{O}(\Delta \mathrm{t}^{\mathrm{J_c}+1}, \mathrm{K}^{-\gamma}), \tag{28}$$

where the subdominant error terms have been neglected.

In Table 1, the local errors τ_K and τ are shown with fixed time step $\Delta t = T/32$ and different integration order J_c. One can see that, as the order of projective integration increases, the error decreases. However, the decrease of error in τ slows down at higher-order $J_c = 3,4$. This is because at higher-order J_c the error term $\mathrm{O}(\Delta t^{J_c+1})$ in (28) becomes subdominant and the overall error reaches the dominant term $\mathrm{O}(K^{-\gamma})$, which remains constant in this case.

In Table 2, we show τ_K and τ with fixed integration order $J_c = 1,2$ and different time steps. Both the errors and the corresponding convergence rates are shown. We can see that the convergence rate of τ_K remains constant equal to $(J_c + 1)$, as predicted by (28), except at $J_c = 2$ *and* very small time steps when the other errors (notably the spatial and temporal errors from DNS) contaminate the rate. On the other hand, the errors in τ converge with rate (J_c+1) only at large Δt and lower-order $J_c = 1$. At smaller time steps the convergence saturates and the errors approach a constant value. Such saturation in errors is more evident for $J_c = 2$. In Fig. 3, both τ_K and τ are plotted against the size of Δt, and we clearly observe the con-

Table 1
Error after one step integration with different order of $J_c (\Delta t = T/32)$

J_c	1	2	3	4
τ_K	7.9373(−2)	2.0632(−2)	8.9058(−3)	2.2198(−3)
τ	8.0399(−2)	2.4282(−2)	1.5597(−2)	1.2995(−2)

Table 2
Local errors τ_K and τ for fixed integration order $J_c = 1,2$, with different time steps Δt

Δt	$J_c = 1$				$J_c = 2$			
	τ_K	R_K	τ	R	τ_K	R_K	τ	R
$T/8$	1.1323(0)	1.90	1.1323(0)	1.90	6.9416(−1)	2.87	6.9429(−1)	2.85
$T/16$	3.0321(−1)	1.97	3.0339(−1)	1.95	9.5245(−2)	2.95	9.6126(−2)	2.44
$T/32$	7.7229(−2)	1.99	7.8283(−2)	1.76	1.2264(−2)	2.95	1.7730(−2)	0.47
$T/64$	1.9400(−2)	1.97	2.3190(−2)	0.77	1.5866(−3)	2.78	1.2683(−2)	0.01
$T/128$	4.9620(−3)	2.00	1.3617(−2)	0.10	2.3080(−4)	2.43	1.2683(−2)	0.00
$T/256$	1.2392(−3)	–	1.2737(−2)	–	4.2821(−5)	–	1.2676(−2)	–

The convergence rate R is defined as $R(\Delta t) = \ln[\tau(\Delta t)/\tau(\Delta t/2)]/\ln 2$; R_K is defined similarly.

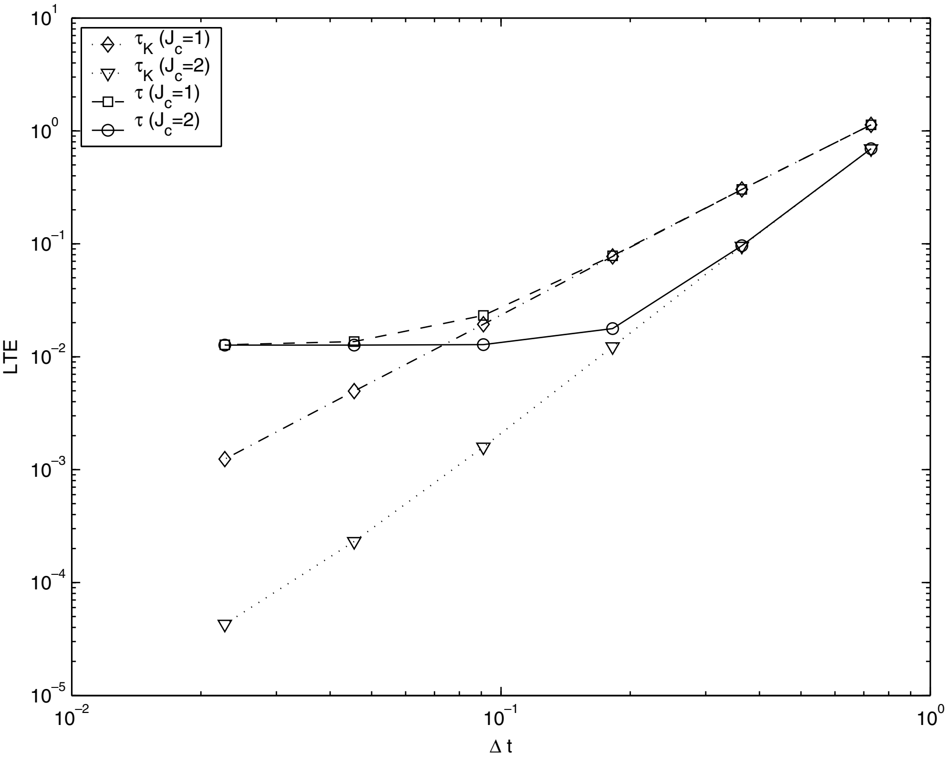


Fig. 3. Local errors τ and τ_K versus Δt (J_c = 1,2).

stant convergence rate of τ_K and the saturation of τ. This observation is consistent with the error estimates (28).

4.2. Periodic flows

To further examine the computational properties of the equation-free POD model, we use projective integration to evolve the aforementioned problem over longer times. Again, at Re = 100, the flow is periodic with period $T \approx 5.824$. In the following results, the order of accuracy for the RHS estimate (17) is fixed at second-order, i.e., J_f = 2.

In Figs. 4 and 5, we plot the results with the same fine integration steps (Δt_f = T/32) and order of projective integrator (J_c = 2), but different projective integration time step at Δt_c = T/64 and Δt_c = T/16, respectively. The number of POD modes used for observation and projective integration is fixed at K = 11. It is seen that with relatively smaller projective time step Δt_c = T/64, the results agree well with DNS (Fig. 4). However, at larger projective step Δt_c = T/16, the equation-free POD model becomes unstable, as shown in Fig. 5. Under the same parameter settings of Fig. 5, the results become accurate and stable if we increase the order of projective integration to J_c = 3, as shown in Fig. 6. In Fig. 7, we use the third-order projective integrator and reduce the steps of fine integration to Δt_f = T/78. Again, the solutions agree well with DNS and they remain stable. In this case, the computational cost of the equation-free POD model is about 20% of full-scale DNS. Fig. 8 shows the results with the same parameters as those in Fig. 4, except that here a smaller number of POD modes (K = 5) is employed. We observe that the results are less accurate and a weak instability has developed.

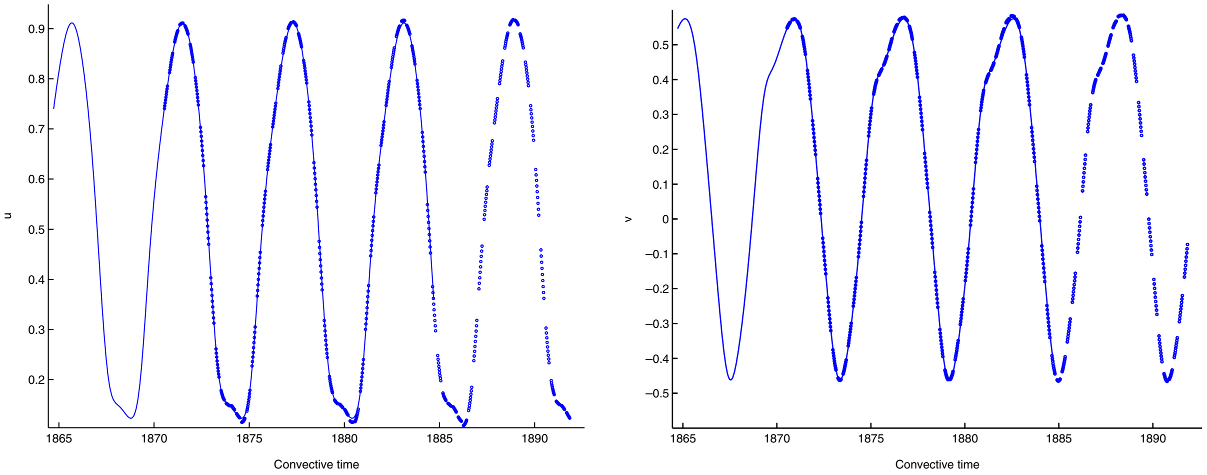


Fig. 4. Solutions at history point with $J_c = 2$ and $\Delta t_c = T/64$ ($J_f = 2$, $\Delta t_f = T/32$, $K = 11$). Left: u-velocity. Right: v-velocity. The lines correspond to DNS, while the symbols to equation-free POD model.

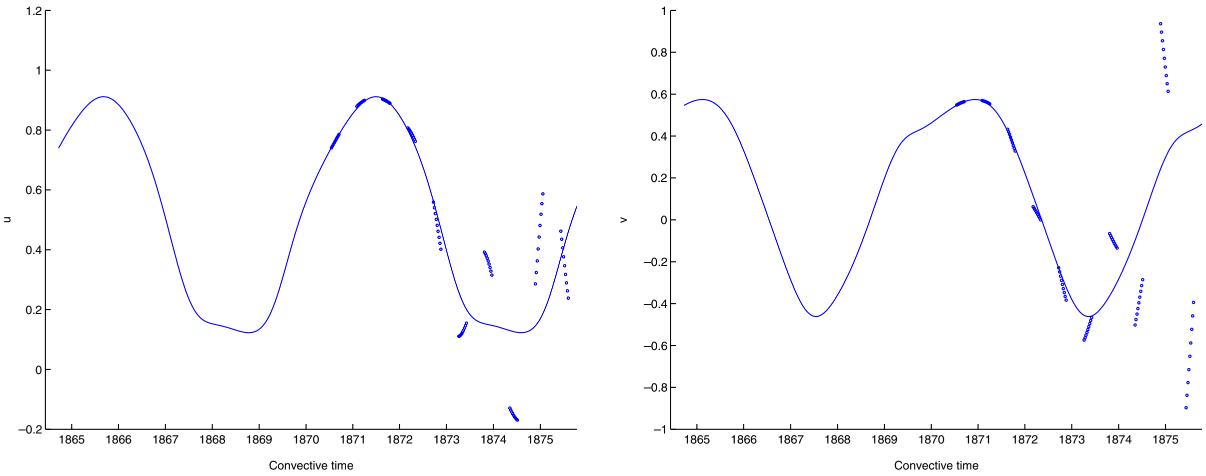


Fig. 5. Solutions at history point with $J_c = 2$ and $\Delta t_c = T/16$ ($J_f = 2$, $\Delta t_f = T/32$, $K = 11$). Left: u-velocity. Right: v-velocity. The lines correspond to DNS, while the symbols to equation-free POD model.

These results suggest that a third-order projective integrator, in addition to its higher accuracy, has better stability property than the second-order one. Also, the resolution of POD expansions, i.e., the number of POD modes employed, has an effect both on the overall accuracy of the algorithm and its stability. We remark that the complete analysis on the stability properties of the equation-free POD model for fluids remains an open issue. For stability analysis of general projective integrations for ordinary differential equations, see [13].

4.3. Quasi-periodic flows

In this section, we examine the equation-free POD model for flows with two incommensurate frequencies. At Reynolds number $Re = 100$, the natural frequency of the flow is $f_s = 1/T \approx 0.17$. We impose a time

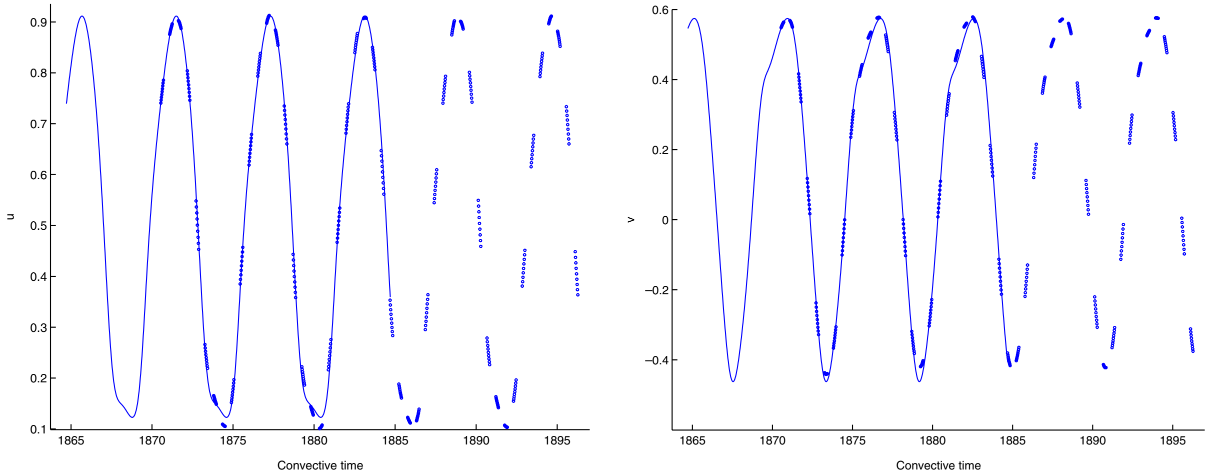


Fig. 6. Solutions at history point with $J_c = 3$ and $\Delta t_c = T/16$ ($J_f = 2$, $\Delta t_f = T/32$, $K = 11$). Left: u-velocity. Right: v-velocity. The lines correspond to DNS, while the symbols to equation-free POD model.

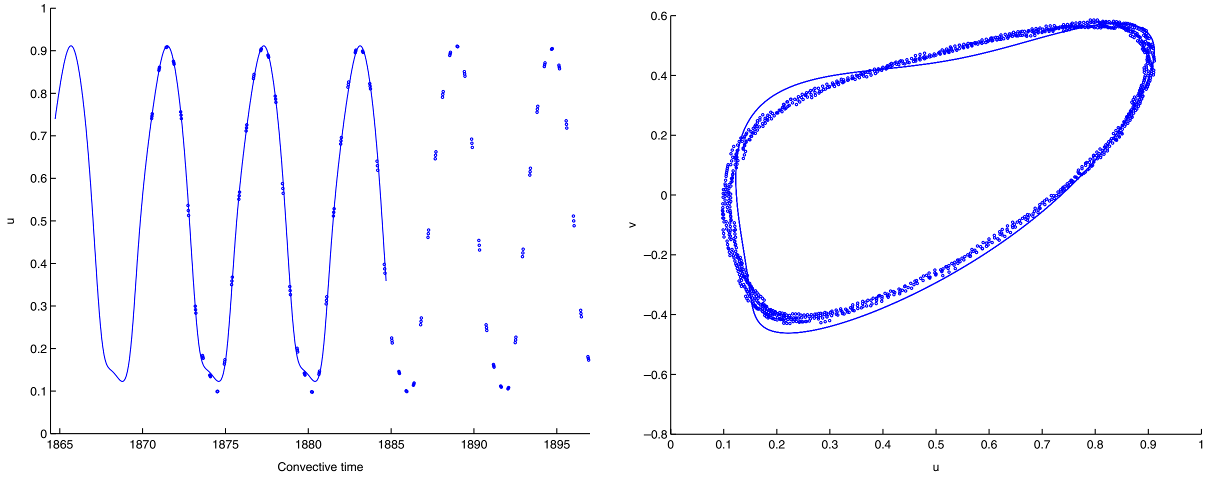


Fig. 7. Solutions at history point with $J_c = 3$ and $\Delta t_c = T/16$ ($J_f = 2$, $\Delta t_f = T/78$, $K = 11$). Left: u-velocity. Right: phase portrait of u versus v. The lines correspond to DNS, while the symbols to equation-free POD model.

dependent inflow condition on the left boundary (see Fig. 2), $\mathbf{u} = (1 + 0.1 \times \sin(2\pi f_i t), 0)$. The forcing frequency is $f_i = 1/T_i = 0.09$; the interplay of the forcing and the natural frequencies gives rise to an apparently quasiperiodic (possibly periodic of extremely high period) flow.

The POD basis function are extracted from 780 snapshots obtained through a simulation over a sampling time interval of $\hat{\mathrm{T}} = \mathrm{T} \cdot \mathrm{T_i} \approx 66.13$, and we retain 93 POD modes in the lifting so that 99.99997% of the energy is captured. Note that $\hat{\mathrm{T}}$ is the least common multiplier of the two characteristic time scales of the problem. The projective integrator is third-order, i.e. $J_c = 3$. The time interval of fine integration is $\Delta \mathrm{t_f} = \hat{\mathrm{T}}/880$ and the time step for projective integration is $\Delta \mathrm{t_c} = \hat{\mathrm{T}}/256$. The computational cost of the equation-free POD model is about 25% of that of a full-scale DNS.

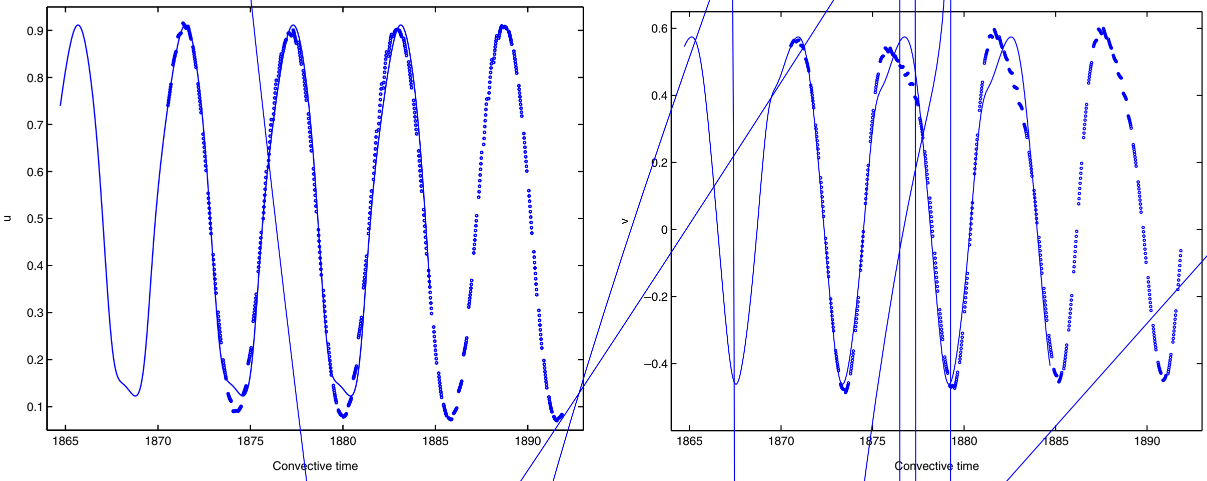


Fig. 8. Solutions at history point with $J_c = 2$ and $\Delta t_c = T/64$ ($J_f = 2$, $\Delta t_f = T/32$, $K = 5$). Left: u-velocity. Right: v-velocity. The lines correspond to DNS, while the symbols to equation-free POD model.

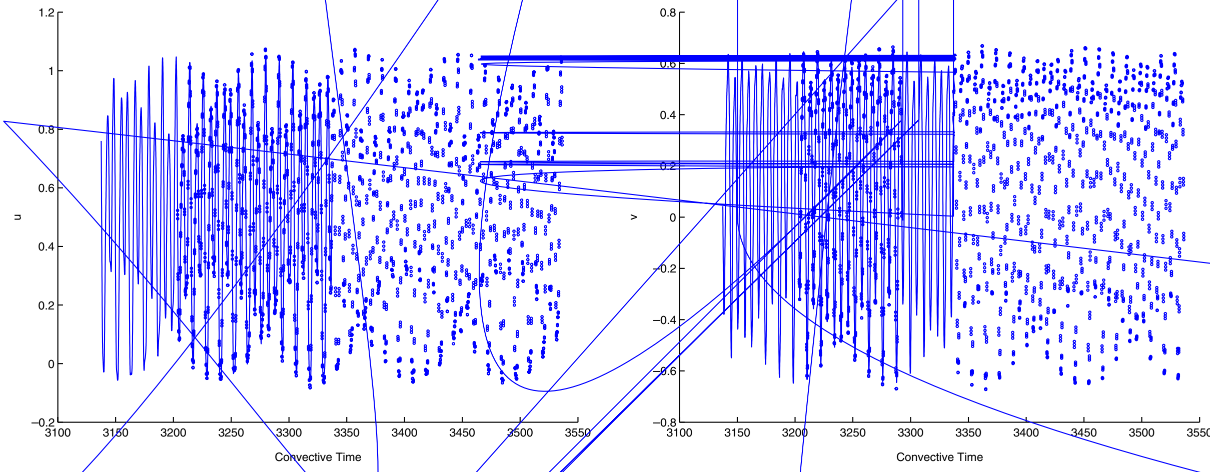


Fig. 9. Evoluation of velocities at history point. Left: u-velocity. Right: v-velocity. The lines correspond to DNS while the symbols to equation-free POD model.

The evolutions of velocities at the history point are plotted in Fig. 9. To illustrate the closeness between equation-free POD model and DNS results, we plot the spectrum of the streamwise velocity in Fig. 10; we observe good agreement. In Fig. 11 (left), we plot the phase portrait of u-velocity versus v-velocity at the history point; in Fig. 11 (right) we plot the Poincaré map corresponding to a value of $u = 0.5$. Again, the equation-free POD produces results with good accuracy, compared to the DNS results.

4.4. Long-term integration

It has been demonstrated that the low-dimensional POD system obtained from Galerkin projection may be quite inaccurate for the long-term integration (e.g., see [11] for ordinary differential equations, [3] for

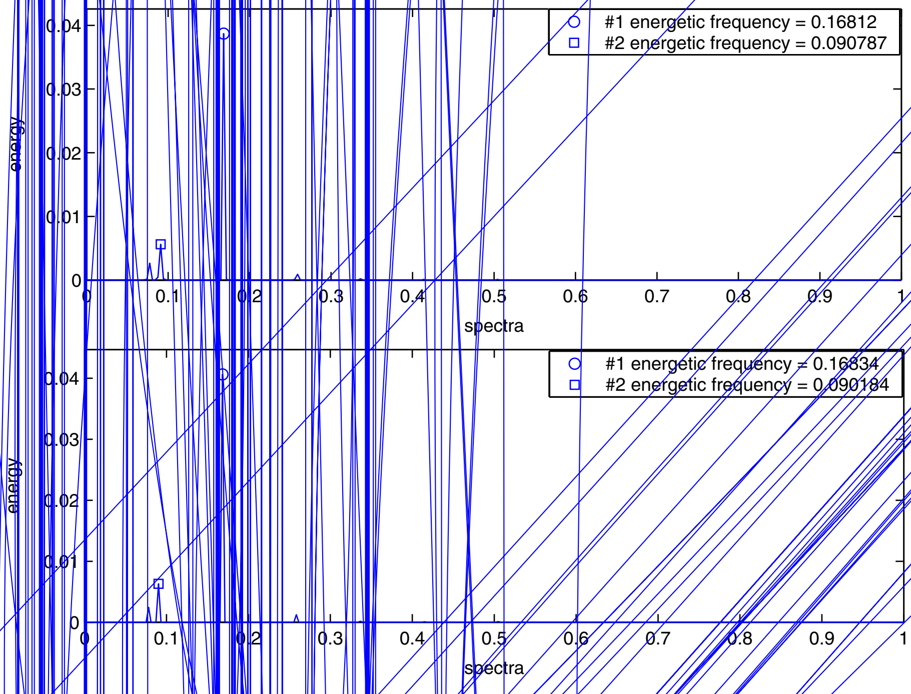


Fig. 10. Spectrum of the *u*-velocity signal at the history point. Upper: DNS; lower: equation-free POD.

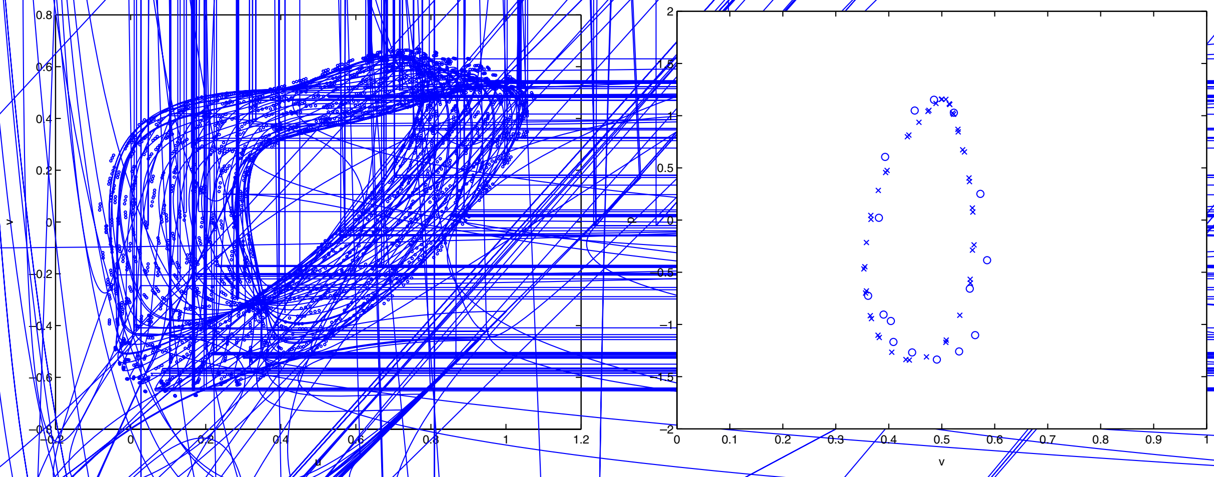


Fig. 11. Left: The phase portrait of *u*-velocity versus *v*-velocity. DNS results in line, equation-free POD in symbols. Right: Poincaré map at the history point with $u = 0.5$. DNS (cross); equation-free POD (circle).

Kuramoto-Sivashinsky equation, and [39] for Navier–Stokes equations). On the other hand, the equation-free POD model presented in this paper can overcome this difficulty because the "nonlinear Galerkin" equations for the POD modes are provided numerically by observing the actual fine scale dynamics. To investigate its performance in long-term integration, we return to the periodic flow of $Re = 100$ and integrate the equation-free POD model for a time interval with length $t = 375 \cdot T \approx 2250$, that is equal to 375 shedding periods.

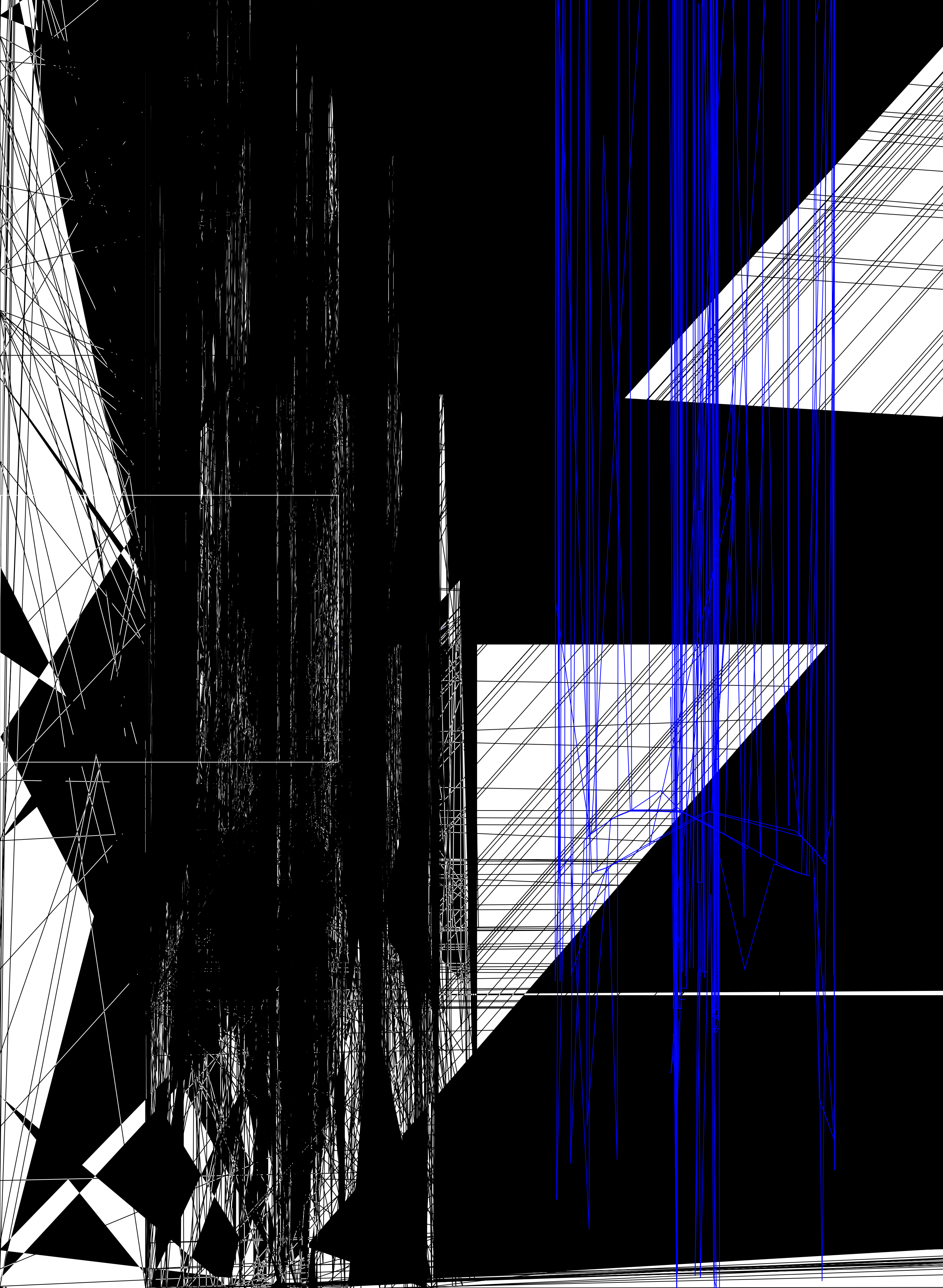
In Fig. 12, phase portraits of the first four POD modes $\{a_k(t)\}$ are shown. The results are obtained via the equation-free POD model with $J_f = 2$, $J_c = 3$, $\Delta t_f = \Delta t_c = T/32$ and $K = 11$. On the left of Fig. 12, we show the limit cycles from the equation-free POD simulations and those of DNS. We observe very good agreement between the two even after long-term integration. On the right of Fig. 12 we show the asymptotically stable solution of the low-dimensional POD models from the Galerkin projection, along with the "correct" DNS results. The Galerkin POD produces clearly erroneous limit cycles. In Fig. 13, we present the results obtained by the Galerkin-free POD model with fewer number of POD expansion modes $K = 7$. The results are less accurate compared to those obtained by $K = 11$ in Fig. 12; assuming that the long-term dynamics live on a manifold of dimension less or equal to 7, the reason for this difference lies most probably in the better lifting (after projection) with 11 (rather than only 7) modes. However, the Galerkin-free model can still capture the nature of the asymptotic states, in contrast to the failure of direct Galerkin POD, even at higher resolution of the latter. (The asymptotically stable solution of the truncated Galerkin POD system is obtained from AUTO, see [10]; for detailed analysis on the asymptotic behavior of Galerkin POD for Navier–Stokes equations, see [39].)

5. Summary and discussion

In this paper, we presented an equation-free/Galerkin-free POD-assisted model for numerical simulations of the long-term dynamics of incompressible flows. Consistency and accuracy analysis of the approach at appropriate limits was conducted, and numerical examples were presented to validate the error estimates. Numerical simulations of two-dimensional flow past a circular cylinder were also presented. The present method produces accurate results while reducing the computational effort by up to 80%, compared to the full-scale DNS. Long-term integrations are conducted to demonstrate that the Galerkin-free POD model can capture the asymptotic state of the flow correctly; this is in sharp constrast with POD-Galerkin truncations, which – even at higher number of modes – produce erroneous asymptotic states.

One of the most important features affecting the performance of equation-free methods is the accuracy of the *lifting* step – given the values of a sufficient number of observables (here, low-POD coefficients) we need to produce initial conditions *on or close* to the slow manifold consistent with these observables. In this paper we initialized with exactly zero component on the higher modes, and relied on pure direct integration to bring us close to the slow manifold. This was sufficient in our case, and will indeed provide a good approximation if the observation variables are "pure slow" variables, and the time of integration – before we start estimating derivatives – is appropriately chosen with respect to the gap between fast and slow system dynamics. When we have a sufficient number of observation variables to parametrize the slow manifold, yet these observables have both fast and slow components, it is still possible to successfully "lift" (initialize close to the slow manifold) with a constrained evolution or even non-intrusively through a legacy DNS simulator [16,12].

Clearly, the computation can benefit from adaptive step-size selection, as well as adaptive selection of the number of POD modes used to observe the evolution of the solution; the latter corresponds to mesh adaptation in finite differences or finite element computations. The "computational technology" required for this adaptation is the same as the one for traditional, Galerkin-based codes (i.e., refinement or coarsening are based on a posteriori error estimates). An important quantity to estimate locally is the location of a possible *spectral gap* between slow and fast dynamics of the full system; this gives us an estimate of the number of variables required to parametrize the slow manifold, and thus an indirect indication of the number of POD modes that we need to keep in our simulation.



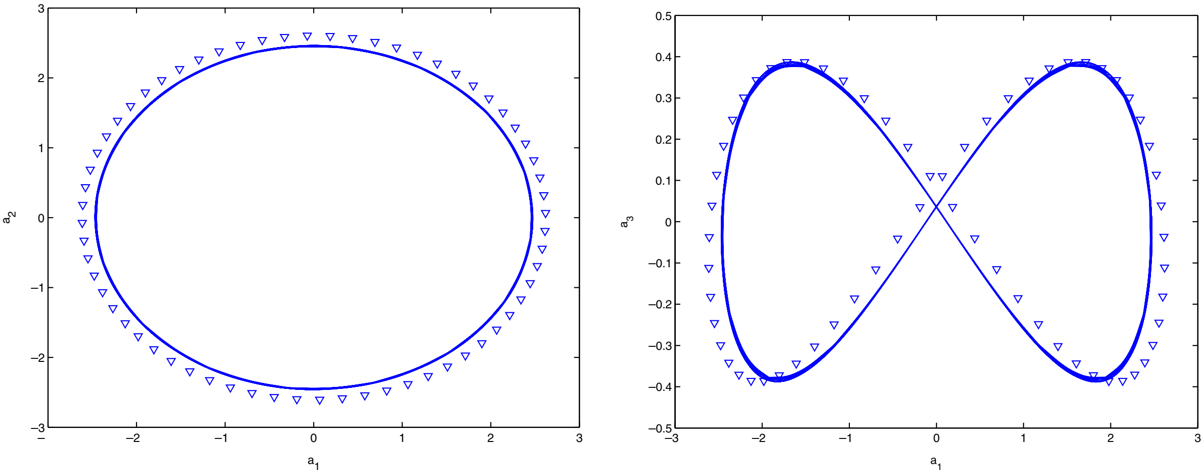


Fig. 13. Limit cycles of the temporal POD modes ($K = 7$). Left: $a_1(t)$ versus $a_2(t)$; Right: $a_1(t)$ versus $a_3(t)$. The lines correspond to Galerkin-free POD model, while the symbols to DNS.

Projective integration is but one of the traditional continuum numerical procedures implemented in an equation-free context. Most other continuum procedures (e.g., fixed point computations, parametric continuation, stability and bifurcation analysis, as well as tasks like controller design and optimization) can also be implemented. Remarkably, even reverse projective integration of highly dissipative systems *on the slow manifold* can be attempted: short forward integration runs can be used to take the dynamics (projectively) backward in time [15]. Short runs of the "full DNS" code can be used to estimate not only the RHS of the equation for the coarse-grained observables, but also to estimate the action of the slow Jacobian of the dynamics, and so implement Krylov subspace methods (e.g. [22]). One can, thus, attempt to turn a DNS dynamic simulator – through a computational "wrapper" – into a fixed point solver, capable of finding both stable and unstable steady states, quantifying their stability and analyzing their bifurcations (e.g., see [37,23]). We are actively pursuing these research directions for complex flows. What we presented here shows that – beyond simple data compression and POD-Galerkin – POD modes can be used (as observation variables in a multiscale context) to accelerate DNS; in principle, they can also be used to enable DNS simulators to become fixed point solvers, to integrate backward in time – to perform computational tasks they have not been designed for. One of the most important points is that -for our equation-free framework – we only need enough POD modes to *observe* the slow dynamics – to parametrize the slow manifold, not to span it. This has the potential to alleviate the severe sensitivity of POD-Galerkin models to the particular data ensemble that was used for basis function extraction. We believe that this opens many interesting possibilities for "on-the-fly" reduction for complex geometry flows, and more generally, for POD-assisted multiscale system computations.

In the current work we have considered the flow past a circular cylinder as a test problem due to the previous experience with this flow that shows that a low-dimensional representation indeed exists, see [28,5,8]. An important limitation of the method presented here is that it relies on the existence of a relatively low-dimensional, attracting slow manifold, parametrized by the POD modes we retain in our procedure; as more modes of the flow become slow the manifold becomes higher dimensional. This simplicity is lost when such a separation of time scales is not present, e.g., in wall-bounded turbulent flows characterized by sudden bursts and the resulting interactions of fast and slow scales. It is not inconceivable, however, that the simplicity might be restored when one does not study the dynamics *of a single flow realization* but the effective, averaged dynamics of of appropriately selected ensembles of realizations. Such examples have been demonstrated in kinetic Monte Carlo or Brownian Dynamics contexts, see [29,30,36]; whether such effective descriptions may be useful for complicated transitional and turbulent flows remains to be seen.

Acknowledgments

The first author gratefully acknowledges the DPST (Development and Promotion of Science and Technology Talents) project from Thailand for providing his scholarship during his graduate studies at Brown University. This work was supported by ONR and NSF, and computations were performed at the facilities of NCSA (University of Illinois at Urbana-Champaign) and at DoDs NAVO MSRC.

References

[1] R.E. Arndt, D.F. Long, M.N. Glauser, The proper orthogonal decomposition of pressure fluctuations surrounding a turbulent jet, J. Fluid Mech. 340 (1997) 1–33.

[2] N. Aubry, P. Holmes, J.L. Stone, J.L. Lumley, The dynamics of coherent structures in the wall region of a turbulent boundary layer, J. Fluid Mech. 192 (1988) 115.

[3] N. Aubry, W.Y. Lian, E.S. Titi, Preserving symmetries in the proper orthogonal decomposition, SIAM J. Sci. Comput. 14 (2) (1993) 483–505.

[4] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Ann. Rev. Fluid Mech. 25 (1993) 539–575.

[5] N.-Z. Cao, N. Aubry, Numerical simulation of wake flow via a reduced system, in: Proc. of the ASME Fluids Engineering Conference, Washington, DC, 1993.

[6] W. Cazemier, R.W. Verstappen, A.E. Veldman, Proper orthogonal decomposition and low-dimensional models for driven cavity flows, Phys. Fluids 10 (7) (1998) 1685–1699.

[7] J.H. Citriniti, W.K. George, Reconstruction of the global velocity field in the axisymmetric mixing layer utilizing the proper orthogonal decomposition, J. Fluid Mech. 418 (2000) 137–166.

[8] A.E. Deane, I.G. Kevrekidis, G.E. Karniadakis, S.A. Orszag, Low-dimensional models for complex geometry flows: application to grooved channels and circular cylinders, Phys. Fluids A 3 (10) (1991) 2337–2354.

[9] J. Delville, L. Ukeiley, L. Cordier, J.P. Bonnet, M. Glauser, Examination of large-scale structures in a turbulent plane mixing layer. Part 1. Proper orthogonal decomposition, J. Fluid Mech. 391 (1999) 91–122.

[10] E.J. Doedel, R.C. Paffenroth, A.R. Champneys, T.F. Fairgrieve, Yu.A. Kuznetsov, B. Sandstede, X. Wang, Auto 2000: continuation and bifurcation software for ordinary differential equations (with homcont), Technical Report, Caltech, 2001.

[11] C. Foias, M.S. Jolly, I.G. Kevrekidis, E.S. Titi, Dissipativity of the numerical schemes, Nonlinearity 4 (1991) 591–613.

[12] C.W. Gear, T.J. Kaper, I.G. Kevrekidis, A. Zagaris, Projecting on a slow manifold: singularly perturbed systems and legacy codes, SIAM J. Appl. Dyn. Sys., to appear (original version can be found as Physics/0405074 at arXiv.org).

[13] C.W. Gear, I.G. Kevrekidis, Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum, SIAM J. Sci. Comput. 24 (2003) 1091–1106 (also NEC Technical Report NECI-TR 2001-029, can be obtained as http://www.neci.nj.nec.com/homepages/cwg/projective.pdf).

[14] C.W. Gear, I.G. Kevrekidis, Telescopic projective methods for parabolic differential equations, J. Comput. Phys. 187 (1) (2003) 95–109 (also a NEC Technical Report, November 2001, can be obtained as http://www.neci.nj.nec.com/homepages/cwg/itproje.pdf).

[15] C.W. Gear, I.G. Kevrekidis, Computing in the past with forward integration, Phys. Lett. A 321 (2004) 335–343 (original version can be found as nlin.CD/0302055 at arXiv.org).

[16] C.W. Gear, I.G. Kevrekidis, Constraint-defined manifolds: a legacy-code approach to low-dimensional computations. J. Sci. Comput., in press.

[17] C.W. Gear, J. Li, I.G. Kevrekidis, The gap-tooth method in particle simulations, Phys. Lett. A 316 (2003) 190–195.

[18] A. Glezer, Z. Kadioglu, A.J. Pearlstein, Development of an extended proper orthogonal decomposition and its application to a time periodically forced plane mixing layer, Phys. Fluids 1 (8) (1989) 1363.

[19] S.V. Gordeyev, F.O. Thomas, Coherent structure in the turbulent planar jet. Part 1. Extraction of proper orthogonal decomposistion eigenmodes and their self-similarity, J. Fluid Mech. 414 (2000) 145–194.

[20] G.E. Karniadakis, M. Israeli, S.A. Orszag, High-order splitting methods for incompressible Navier–Stokes equations, J. Comput. Phys. 97 (1991) 414.

[21] G.E. Karniadakis, S.J. Sherwin, Spectral/hp Element Methods for CFD, Oxford University Press, Oxford, 1999.

[22] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, PA, 1995.

[23] C.T. Kelley, I.G. Kevrekidis, Q. Liang, Newton–Krylov solvers for time-steppers. SIAM J. Appl. Dyn. Sys., submitted (original version can be found as math.DS/0404374 at arXiv.org).

[24] I.G. Kevrekidis, C.W. Gear, G. Hummer, Equation-free: the computer-assisted analysis of complex, multiscale systems, AIChE J. 50 (7) (2004) 1346–1354.

[25] I.G. Kevrekidis, C.W. Gear, J.M. Hyman, P.G. Kevrekidis, O. Runborg, C. Theodoropoulos, Equation-free coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis, Commun. Math. Sci. 1 (4) (2003) 715–762 (original version can be found as physics/0209043 at arXiv.org).

[26] J. Li, P.G. Kevrekidis, C.W. Gear, I.G. Kevrekidis, Deciding the nature of the coarse integration through microscopic simulations: the baby-bathwater scheme, SIAM J. Multiscale Model. Simulation 1 (2003) 391–407.

[27] A. Liakopoulos, P.A. Blythe, H. Gunes, A reduced dynamical model of convective flows in tall laterally heated cavities, Proc. Roy. Soc. London Ser. A 453 (1997) 663.

[28] X. Ma, G.S. Karamanos, G.E. Karniadakis, Dynamics and low-dimensionality of the turbulent near-wake, J. Fluid Mech. 410 (2000) 29–65.

[29] A. Makeev, D. Maroudas, I.G. Kevrekidis, Coarse stability and bifurcation analysis using stochastic simulators: kinetic Monte Carlo examples, J. Chem. Phys. 116 (2002) 10083–10091.

[30] A. Makeev, D. Maroudas, A. Panagiotopoulos, I.G. Kevrekidis, Coarse bifurcation analysis of kinetic Monte Carlo simulations: a lattice-gas model with lateral interactions, J. Chem. Phys. 117 (2002) 8229–8240.

[31] D. Rempfer, On low-dimensional Galerkin models for fluid flow, Theoret. Comput. Fluid Dyn. 14 (2) (2000) 75–88.

[32] D. Rempfer, H.F. Fasel, Evolution of three-dimensional coherent structures in a flat-plate boundary layer, J. Fluid Mech. 260 (1994) 351.

[33] R. Rico-Martinez, C.W. Gear, I.G. Kevrekidis, Coarse projective kMC integration: forward/reverse initial and boundary value problems, J. Comput. Phys. 196 (2004) 474–489 (original version can be found as nlin.CG/0307016 at arXiv.org).

[34] O. Runborg, C. Theodoropoulos, I.G. Kevrekidis, Effective bifurcation analysis: a time-stepper based approach, Nonlinearity 15 (2002) 491–511.

[35] S. Setayeshgar, C.W. Gear, H.G. Othmer, I.G. Kevrekidis, Application of coarse integration to bacterial chemotaxis. SIAM J. Multiscale Model. Simulation, in press (also physics/0308040 at arXiv.org).

[36] C. Siettos, M.D. Graham, I.G. Kevrekidis, Bifurcation, projective integration and control via stochastic simulation, J. Chem. Phys 118 (2) (2003) 10149–10157.

[37] C.I. Siettos, C.C. Pantelides, I.G. Kevrekidis, Enabling dynamic process simulators to perform alternative tasks: a time-stepper based toolkit for computer-aided analysis, Ind. Eng. Chem. Res. 42 (26) (2003) 6795–6801 (original version can be found as math.OC/0302040 at arXiv.org).

[38] S.N. Singh, H.M. James, A.A. Gregory, B. Siva, K.H. James, Optimal feedback control of vortex shedding using proper orthogonal decomposition models, Trans. ACME 123 (2001) 612–618.

[39] S. Sirisup, G.E. Karniadakis, A spectral viscosity method for correcting the long-term behavior of POD models, J. Comput. Phys. 194 (1) (2004) 92–116.

[40] L. Sirovich, Turbulence and the dynamics of coherent structures, Parts I, II and III, Quart. Appl. Math. XLV (1987) 561–590.

[41] C. Theodoropoulos, Y. Qian, I.G. Kevrekidis, Coarse stability and bifurcation analysis using time-steppers: a reaction-diffusion example, Proc. Natl. Acad. Sci. (2000) 97.