Journal of Computational Physics 229 (2010) 3046–3063

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Reduced order models based on local POD plus Galerkin projection

María-Luisa Rapún [*], José M. Vega

ETSI Aeronáuticos, Universidad Politécnica de Madrid, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 12 March 2009
Received in revised form 16 December 2009
Accepted 23 December 2009
Available online 4 January 2010

Keywords:
Low dimensional models
Surrogate models
Low dimensional dynamics
Proper orthogonal decomposition

ABSTRACT

A method is presented to accelerate numerical simulations on parabolic problems using a numerical code and a Galerkin system (obtained via POD plus Galerkin projection) on a sequence of interspersed intervals. The lengths of these intervals are chosen according to several basic ideas that include an a priori estimate of the error of the Galerkin approximation. Several improvements are introduced that reduce computational complexity and deal with: (a) updating the POD manifold (instead of calculating it) at the end of each Galerkin interval; (b) using only a limited number of mesh points to calculate the right hand side of the Galerkin system; and (c) introducing a second error estimate based on a second Galerkin system to account for situations in which qualitative changes in the dynamics occur during the application of the Galerkin system. The resulting method, called local POD plus Galerkin projection method, turns out to be both robust and efficient. For illustration, we consider a time-dependent Fisher-like equation and a complex Ginzburg–Landau equation.

1. Introduction

Reduced order models (ROMs) are receiving a continuously increasing attention in the literature due to their interest in both understanding basic mechanisms of fluid systems [8,15,16] and improving prediction and design in industrial processes [5,12,21]. Concerning the latter, the need of reducing development time and cost in industry is nowadays enhancing a trend to substitute wind tunnel experiments by numerical simulations. But traditional numerics are far too slow in multi-parameter situations. These are frequent in e.g. aircraft industry [14], where design and certification based on simulations usually involve many parameters and thus require to perform thousands of runs of complete aircraft configurations.

Proper orthogonal decomposition (POD) combined with *Galerkin projection* has been used during the last 20 years to obtain ROMs of evolution problems. The idea is to first identify a low dimensional POD manifold that contains a good approximation of the true dynamics *in an attractor* of the system, and then use as ROM a *Galerkin system* (GS) obtained projecting the exact equations onto this manifold. A basis of the POD manifold is obtained by the so called *method of snapshots* [20], which consists of applying POD methodology to a set of numerically calculated snapshots that span the low dimensional manifold. Thus, a sufficiently precise *numerical code* (NC) is needed to calculate the snapshots. Such code must be run over a sufficiently large time interval, to ensure that the resulting orbit covers a representative part of the attractor, which requires some a priori knowledge of the attractor dynamics. The method was introduced to obtain ROMs of complex dynamics of incompressible fluid flow problems [4]. The fact that nonlinearity is quadratic in Navier–Stokes equations allowed to obtain a (quadratic) polynomial GS, whose coefficients are obtained by some preprocessing. The main difficulty is that the resulting GS may exhibit spurious dynamics in a somewhat unpredictable way. The reason for that is still controversial, but seems to be due to the fact that the POD manifold is not invariant under the true dynamics. Thus, intended solutions to this difficulty

[*] Corresponding author. Tel.: +34 913363287.
E-mail addresses: marialuisa.rapun@upm.es (M.-L. Rapún), josemanuel.vega@upm.es (J.M. Vega).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.12.029

relied on the idea of introducing some corrections on the GS, which can be seen as correcting the POD manifold with some nonlinear terms to make it invariant. This is done either by introducing some additional terms into the GS [7,17] or by correcting the reduced model projecting the error onto the POD temporal eigenfunctions (intrinsic stabilization [11]). An alternative way to stabilize the reduced dynamics consists of calculating the truncated dynamics using a few time steps of the original NC [19]. See also [18] for the effect of time-dependent boundary conditions and [22] for the derivation of ROMs in stochastic systems. In all these cases, the GS is intended to approach the dynamics of the system on a particular attractor, which can be periodic, quasi-periodic, or chaotic; turbulent states are the obvious target of such efforts. POD modes were calculated from the outset, as explained above. Thus, this method will be referred to as the *pre-processed POD + Galerkin projection method*.

Our approach is somewhat different, in the sense that our goal is to construct a ROM that approximates any dynamics of the system. In other words, we do intend to approximate not only attractors but also transients. Our method can also be seen as a way to accelerate any NC (see [6] for an alternative acceleration procedure based on POD methods) by combining the use of the NC itself with that of the GS, as in the method proposed in [19]; also, as in [19], we do not restrict ourselves to polynomial nonlinearities.

Let us now anticipate the main ideas behind the method. Assume that we have a time-dependent physical system, governed by a parabolic equation or system, and a NC to calculate trajectories of the system. Let us apply the NC over a time interval $I_{\rm NC}$, beginning at $t = 0$, and select some snapshots on $I_{\rm NC}$, such that they provide a good approximation of the whole solution in $I_{\rm NC}$. Then, we calculate the most energetic POD modes obtained from these snapshots in such a way that the associated POD manifold approaches well the solution in $I_{\rm NC}$. The question is: will such POD manifold still approach the orbit for a larger time interval? It turns out that the answer is yes. But extending (in time) the validity of the POD manifold requires to select a few more modes than necessary, as could be anticipated from the expected continuous dependence of the POD manifold on the time interval $I_{\rm NC}$. Then, the *basic method* to obtain an approximation of the solution in some time span $0 < t < T$ can be described as follows:

(i) Take the time interval $I_{\rm NC}$ as explained above, select a set of snapshots in this interval, and calculate the resulting POD modes.
(ii) Project the original parabolic equations onto these modes and integrate the resulting GS over an interval $I_{\rm GS}$, defined such that the solution of the GS approaches well the solution of the original NC in $I_{\rm GS}$.
(iii) Go back to step i in a new time interval $I_{\rm NC}$, to calculate new snapshots and a new set of POD modes.
(iv) Repeat the process as many times as necessary, until the final time, $t = T$, is reached.

The crucial point is of course to decide in step ii when the GS fails to approximate the true dynamics, and to do that without using the original NC. This can be done using an *a priori error estimate* that only relies on the assumption that the Galerkin approximation converges to the right solution if an appropriately large number of modes are kept. Specifically, in order to obtain an approximation within an error ε, we proceed as follows in step ii. We choose $\varepsilon_1 < \varepsilon$ (say, $\varepsilon_1 = \varepsilon/100$) and select the first n POD modes such that the *root mean square* (RMS) distance of the snapshots to the resulting POD manifold is bounded by ε_1. These modes will approximate well the orbit in the next Galerkin interval $I_{\rm GS}$. Then we choose the integer $n_1 > n$ such that the RMS distance of the snapshots to the resulting POD manifold is still smaller than (say) $\varepsilon_1/100$, and project the original equations into these n_1 modes; thus, the order of the Galerkin system is n_1. These additional $n_1 - n$ modes will be used to estimate the error of the solution of the GS.

Note that any instability of the Galerkin system produces errors that, once detected by our a priori error estimate, promote recalculating the POD manifold using a few steps of the numerical code; thus, no additional stabilization procedure (as, e.g. that presented in [11]) is necessary.

We call this method the *local POD + Galerkin projection method* because the POD manifold is calculated at each interval $I_{\rm NC}$ using the local dynamics, as done in [19]. Thus, as in [19] we expect that the method will provide a good approximation of the true dynamics over any time interval, eliminating spurious behaviors. The differences with the method presented in [19] are that (a) we do not limit ourselves to a given attractor, (b) we use the short runs with the NC to recalculate the POD modes, not to calculate the GS (which is calculated in a standard way), and (c) our method does not require to pre-calculate the POD modes.

It turns out that the above mentioned error estimate is quite good, and that the resulting method is robust. Namely, the method can be applied to various parabolic problems and is fairly insensitive to changes in the factor 1/100 appearing above. These suggest that the ideas in this paper rely on (deeper than the obvious ones) mathematical properties of parabolic problems, which are ahead of the scope of this paper. Here, we shall limit ourselves to the description and improvement of the method. In fact, the basic method described above can be improved in various ways that become clear from some simple ideas, which are inspired on the former work of one of us on related stationary problems on Aerodynamics and thermal systems [1,2,13]:

(a) As described above, POD modes are obtained each time using information from snapshots calculated in the last $I_{\rm NC}$ interval. In other words, available information on POD modes calculated at former $I_{\rm NC}$ intervals is not used. On the other hand, continuous dependence of the local POD manifold on time, and the fact that the POD manifold provided a good approximation in the last $I_{\rm GS}$ interval, suggest that the POD manifold only suffers a small rotation between two consecutive $I_{\rm NC}$ intervals. Therefore, it is only such rotation that needs to be calculated, which seemingly requires much less information (namely, shorter $I_{\rm NC}$ intervals) than calculating the whole POD manifold.

(b) Standard Galerkin projection on the POD manifold must be done in each numerical integration step of the GS, which requires to calculate integrals over the whole computational domain. This could be expensive when the spatial mesh exhibits a large number of points. But the essence of the method is that the whole set of snapshots is well approximated by a set of n_1 POD modes. Thus, projection of the exact equations on the POD modes should require only information from a number of points in the computational domain that is not much larger than n_1; in fact, if all calculations were exact, n_1 points should be enough. In some sense, we are using here the same underlying idea that justifies the method of snapshots. It turns out that projection can be made using a few points, which leads to a computationally inexpensive way of calculating the right hand sides of the GS.

(c) The method and the improvement described in item (a) rely on the assumption that the POD manifold spanned by the first n_1 modes provides a good approximation of the solution in the next I_{GS}-interval, and that both n_1 and n can be selected from information on the solution in the previous I_{NC} and I_{GS} intervals. But it might happen that some transition occurs in the next I_{GS} interval that is associated with dynamic features of the system that are not present in the solution in previous intervals. Such transition can be anticipated and the whole approximation can be checked using the method itself, by running a second GS with $n_2 > n_1$ modes and appropriately comparing the solutions provided by the two Galerkin systems. The resulting test turns out to be quite effective.

(d) As it will be seen in Fig. 5 below, the method is quite robust in connection with the factors $\varepsilon_1/\varepsilon$ and $\varepsilon_2/\varepsilon_1$, which have been taken as 1/100 above. Thus, these factors are to be selected after some simple calibration. Of course, an optimal choice of these factors could be made, but the benefit of such strategy is dubious. The lengths of the interspersed intervals I_{NC} and I_{GS}, instead have a stronger effect in the computational efficiency, and will be selected by the method itself using simple selection criteria. See the remark at the end of Section 3.

In order to explain and apply the ideas above in a simple way, we shall consider two one-dimensional parabolic equations, in the spatial interval $I =]0,1[$, namely:

(A) The *nonsteady, nonsymmetric Fisher*-like (NSNSF) equation

$$\partial_t u = \nu \partial_{xx}^2 u + \partial_x u + f(u,t), \quad \text{with } u = 0 \text{ at } x = 0, 1, \tag{1.1}$$

where the state variable u is real, $\partial_t, \ \partial_x$, etc., stand hereafter for partial derivatives, $\nu > 0$, and

$$f(u,t) = 1 + \mu_1 \cos(\omega_1 t) + \mu_2 u \cos(\omega_2 t) - u^3, \tag{1.2}$$

for some real parameters $\mu_1, \ \mu_2, \ \omega_1$, and ω_2, which will be chosen below. We call this equation nonsymmetric because of the convective and spatially constant terms, which break the reflection symmetries $x \to 1 - x$ and $u \to -u$, respectively. Also, time dependence with two frequencies allows nontrivial large time behavior (at least, periodic or quasi-periodic). As initial condition, we take

$$u = 0 \quad \text{at } t = 0. \tag{1.3}$$

This simple equation will be used in the first part of the paper to illustrate both the method and the improvements (a) and (b) mentioned above. But temporal complexity in this NSNSF equation is limited because it is essentially due to the time-dependent forcing terms. In other words, complexity is not intrinsic. Thus, the method will be checked applying it to a second equation that exhibits richer dynamics.

(B) The *complex Ginzburg–Landau* (CGL) *equation*,

$$\partial_t u = (1 + \mathrm{i}\alpha)\partial_{xx} u + \mu u - (1 + \mathrm{i}\beta)|u|^2 u, \quad \text{with } u = 0 \text{ at } x = 0, 1, \tag{1.4}$$

which is a well known paradigm of a simple equation that exhibits intrinsically complex dynamics [3]. Here, the state variable u is complex and the parameters μ, α, and β are real. The initial condition (1.3) yields the trivial solution $u = 0$. Thus, we impose a nonzero initial condition, namely

$$u = \mathrm{i}\sin(2\pi x) + (1 + \mathrm{i})\sin(3\pi x) \quad \text{at } t = 0, \tag{1.5}$$

which is selected such that it is not invariant under the continuous symmetry group of the equation, namely $x \to 1 - x, \ u \to u\mathrm{e}^{\mathrm{i}c}$. This equation exhibits the *modulational instability* if $\alpha\beta < -1$ (Newell's condition) and μ is larger than a threshold value, which yields complex behaviors at large time.

The NC to integrate (1.1) and (1.4) is constructed discretizing first and second order derivatives with centered differences, in an equispaced mesh with M_0 points; the resulting ODE system is numerically integrated using MATLAB ode15s.

With these ideas in mind, the remaining of the paper is organized as follows. POD methodology and Galerkin projection are first recalled in Section 2, where some comments are also made for convenience and notation is established. The basic version of the local POD + Galerkin projection method is described in Section 3, and the three improvements mentioned in items (a)–(c) above are introduced in Sections 4–6. The paper ends with some summarizing remarks, in Section 7.

2. Proper orthogonal decomposition, the method of snapshots, and Galerkin projection

Let us consider a real or complex parabolic system, with state variable u, and calculate N snapshots, namely N portraits of the state of the system at N values of $t, t_1, \dots, t_N$,

$$u_1 = u(x,t_1), \dots, u_N = u(x,t_N). \tag{2.1}$$

Applying POD machinery, we obtain the *POD modes* and the *singular values* associated with the snapshots (2.1), denoted as

$$U_1, \dots, U_N \quad \text{and} \quad \sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_N \geqslant 0. \tag{2.2}$$

These are calculated from the (Hermitian) *covariance matrix* R, defined as

$$R_{ij} = \langle u_i, u_j \rangle, \tag{2.3}$$

where (unless otherwise stated) we consider the usual L_2-inner product and norm

$$\langle u_1, u_2 \rangle = \int_0^1 \bar{u}_1(x) u_2(x)\, dx, \quad \|u\|_{L_2} = \sqrt{\int_0^1 |u|^2\, dx}, \tag{2.4}$$

with the overbar standing hereafter for the complex conjugate. *POD modes* are *orthonormal* with this inner product, and can be written in terms of the snapshots as

$$U_j = \frac{1}{\sigma_j} \sum_{k=1}^{N} \alpha_j^k u_k, \tag{2.5}$$

where $(\sigma_j)^2$ are the eigenvalues and α_j^k are the eigenvectors of the matrix R, namely

$$\sum_{k=1}^{N} R_{ik} \alpha_j^k = (\sigma_j)^2 \alpha_j^i, \quad \text{for } j = 1 \dots, N. \tag{2.6}$$

Since POD modes are orthonormal, we have

$$\sum_{k=1}^{N} \bar{\alpha}_j^k \alpha_l^k = \delta_{jl}, \quad \sum_{k=1}^{N} \bar{\alpha}_k^j \alpha_k^l = \delta_{jl}, \tag{2.7}$$

where δ_{jl} (=1 if $j = l$ and 0 otherwise) is the Krönecker delta.

Using all these, we can obtain the following expression for the original snapshots in terms of the POD modes

$$u_l = \sum_{j=1}^{N} \sigma_j \bar{\alpha}_j^l U_j. \tag{2.8}$$

The POD modes defined as above are such that for each $n < N$, the expansion (2.8) truncated to n terms provides *the best joint RMS approximation of the snapshots (2.1) among the expansions with n terms*. In fact, invoking (2.7) and (2.8) we obtain the following bound of the square of the L_2-error resulting from reconstructing the snapshots using n POD modes

$$\sum_{l=1}^{N} \|u_l - \sum_{j=1}^{n} \sigma_j \bar{\alpha}_j^l U_j\|_{L_2}^2 = \sum_{l=1}^{N} \| \sum_{j=n+1}^{N} \sigma_j \bar{\alpha}_j^l U_j\|_{L_2}^2 = \sum_{j=n+1}^{N} (\sigma_j)^2, \tag{2.9}$$

which means that the RMS error when reconstructing the N snapshots after truncation to n modes is

$$\text{RMS Error} = \sqrt{\frac{1}{N} \sum_{j=n+1}^{N} (\sigma_j)^2}. \tag{2.10}$$

Now, the solutions of the original problem are projected onto the POD manifold as

$$u \simeq \tilde{u} = \sum_{j=1}^{n} a_j(t) U_j, \quad \text{with } a_j = \langle U_j, u \rangle, \tag{2.11}$$

where we have taken into account that POD modes are orthonormal. The coefficients a_j will be called below the *amplitudes of the POD modes*.

For convenience, the following notation will be used in the paper. The (instantaneous) spatial error associated with the expansion (2.11) is $u - \tilde{u}$, where the solution provided by the NC, u, is considered as exact below. This is consistent with the fact that spatial derivatives will be discretized using the same (finite differences) scheme in both the numerical code and the Galerkin system. The error can be measured using the L_2-norm as

$$\text{Error}_{L_2}^n = \|u - \tilde{u}\|_{L_2} \equiv \|u - \sum_{j=1}^{n} a_j(t) U_j\|_{L_2}. \tag{2.12}$$

Now, if $\mathrm{Error}_{L_2}^{n_1}$ is sufficiently small for some $n_1 > n$, then the quantity

$$E_n^{n_1} = \sqrt{\sum_{j=n+1}^{n_1} (a_j)^2} \tag{2.13}$$

is a good estimate of $\mathrm{Error}_{L_2}^{n}$. This is because if $\mathrm{Error}_{L_2}^{n_1}$ is sufficiently small, then $u \simeq \sum_{j=1}^{n_1} a_j(t)U_j$ and thus (recall that POD modes are orthonormal)

$$\|u - \sum_{j=1}^{n} a_j(t)U_j\|_{L_2}^2 \simeq \| \sum_{j=n+1}^{n_1} a_j(t)U_j\|_{L_2}^2 \equiv \sum_{j=n+1}^{n_1} (a_j)^2,$$

namely $E_n^{n_1} \simeq \mathrm{Error}_{L_2}^{n}$. This error estimate is fairly standard (similar to, e.g. the one used in spectral methods for dealiasing [9]) and will play an essential role below.

Let us now project the NSNSF equation (1.1) or the CGL equation (1.4) into the POD manifold. Seeking an approximation of the solution of the form (2.11), we must substitute this expansion into (1.1) (or into (1.4)) and project (orthogonally) the resulting equation on the POD manifold, which yields the following GS

$$a_j' = \varphi_j(a_1, \ldots, a_n, t) \quad \text{for } j = 1, \ldots, n, \tag{2.14}$$

where

$$\varphi_j(a_1, \ldots, a_n, t) = \left\langle U_j, F\left(\sum_{k=1}^{n} a_k U_k, t\right)\right\rangle, \tag{2.15}$$

with

$$F(u,t) = \mathcal{L}u + f(u,t) \tag{2.16}$$

denoting the right hand side of (1.1) or (1.4). Here $\mathcal{L}$ stands for the linear, differential part of F, with spatial derivatives discretized as in the NC. Thus,

$$\varphi_j(a_1, \ldots, a_n, t) = \sum_{i=1}^{n} \ell_{ij} a_i + \left\langle U_j, f\left(\sum_{k=1}^{n} a_k U_k, t\right)\right\rangle, \tag{2.17}$$

where the matrix ℓ,

$$\ell_{ij} = \langle U_j, \mathcal{L}U_i\rangle, \tag{2.18}$$

can be calculated from the outset, but the second term in (2.17) must be calculated at each step of the numerical integration of (2.14). Some preprocessing would be also possible in connection with this latter term taking advantage of the fact that nonlinearity is polynomial (cubic) in the equations considered in this paper, but (a) this would require to calculate a fourth order tensor, which is fairly expensive, and (b) we intend not to restrict ourselves to polynomial nonlinearities. Also note that no integration by parts is applied in the right hand side of (2.18) (as sometimes done to reduce the order of the derivatives appearing in the integrals), which will allow us in Section 5 to substitute the L_2-inner product by a more convenient inner product.

Note that the boundary conditions have not been imposed. This is because the boundary conditions are homogeneous and hold for all snapshots, which means that they are also satisfied by all POD modes and thus by the expansion (2.11). Nonhomogeneous boundary conditions can be accounted for introducing a change of variable that transforms them into homogeneous ones.

3. The basic local POD plus Galerkin projection method

In the *basic method* we proceed as anticipated in Section 1, using the estimate (2.13) of the L_2-error (2.12). More precisely, the basic method consists of the following steps:

(i) To begin with, set $t_0 = 0$.

(ii) Integrate (1.1) with the NC in the interval $I_{\mathrm{NC}} : t_0 < t < t_1 = t_0 + \delta_{\mathrm{NC}}$. Select as snapshots the flow portraits at N equispaced values of t, namely $u_j(x) = u(x, t_0 + j\delta_{\mathrm{NC}}/N)$, for $j = 1, \ldots, N$. The number of snapshots, N, should be larger than the required number of modes (say, $N \sim 2n_1$ or larger, with n_1 as defined below), which requires some calibration.

(iii) Calculate the associated POD modes and singular values, as explained in Section 2. Select the integer n as the smallest integer such that the RMS error (2.10) is bounded by $\varepsilon/100$, namely

$$\sqrt{\frac{1}{N}\sum_{j=n+1}^{N} (\sigma_j)^2} < \varepsilon_1 = \frac{\varepsilon}{100}, \tag{3.1}$$

and select n_1 as the smallest integer such that

$$n_1 \geqslant n+1, \quad \sqrt{\frac{1}{N}\sum_{j=n_1+1}^{N}(\sigma_j)^2} < \frac{\varepsilon_1}{100} = \frac{\varepsilon}{10000}. \tag{3.2}$$

(iv) Consider the GS (2.14) truncated to n_1 modes, taking as initial condition at $t = t_1$ the projection on the POD manifold of the NC solution calculated in step ii at $t = t_1$. Integrate the GS (using, e.g. MATLAB ode15s) monitoring the error estimate $E_n^{n_1}$, defined in (2.13), until the last value of t, t_2, such that

$$E_n^{n_1} < \varepsilon. \tag{3.3}$$

(v) If $t_2 < T$, then set $t_0 = t_2$, take as initial condition the value of u at t_2 reconstructed from the last Galerkin state, using (2.8), and go back to step ii. Otherwise, the procedure ends.

In order to measure the effectiveness of the method, we define the *compression factor* as the ratio between the total time span and the total length of the I_{NC} intervals, namely

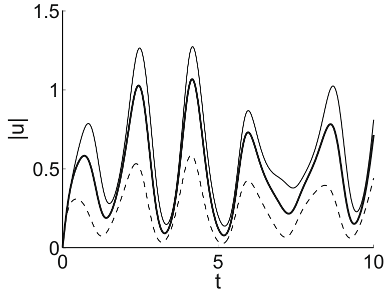


Fig. 1. NC solution (using a mesh of $M_0 = 300$ equispaced points) of the solution of the NSNSF equations (1.1)–(1.3), with $\nu = 0.1$, $\mu_1 = 1$, $\mu_2 = -2$, $\omega_1 = \pi$, and $\omega_2 = 4$ at $x = 1/2$ (——), $x = 1/4$ (——) and $x = 3/4$ (– – –).

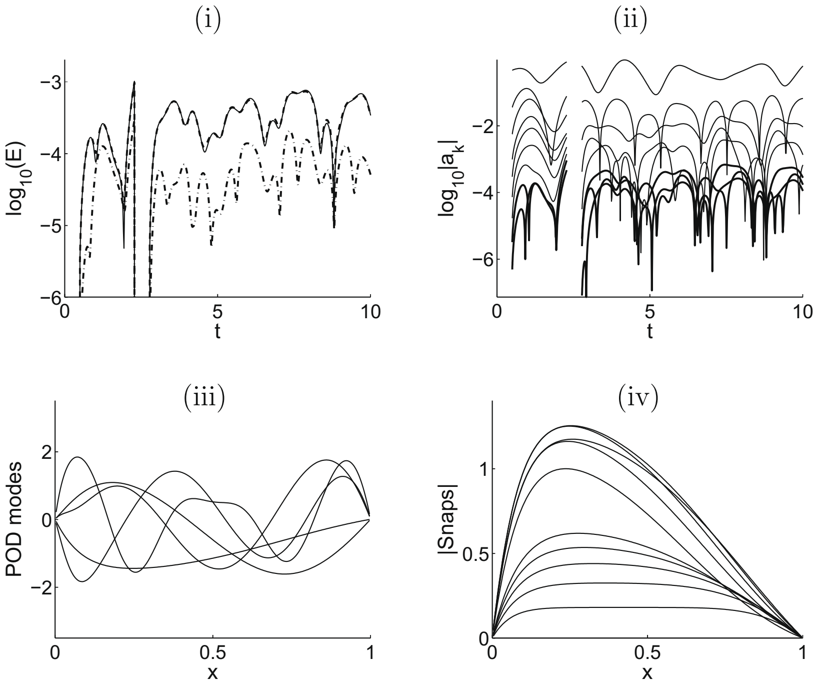


Fig. 2. Results of applying the basic procedure to the NSNSF equation for the same parameter values as in Fig. 1, with $\varepsilon = 10^{-3}$, $\varepsilon_1 = \varepsilon/100$, $N = 20$, and $\delta_{\text{NC}} = 0.5$. (i) The error estimate $E_n^{n_1}$ (——) and the errors $\text{Error}_{L_2}^{n}$ (– – –) and $\text{Error}_{L_2}^{n_1}$ (–·–·–); (ii) the absolute value of the mode amplitudes of the first n modes (——) and the remaining $n_1 - n$ modes (——); (iii) the first $n = 5$ POD modes in the last Galerkin interval and (iv) the absolute value of the snapshots (five equispaced of them in each of the two NC intervals).

$$\text{Compression factor} = \frac{T}{\sum \delta_{\rm NC}}.$$

For illustration, we apply this basic method to the NSNSF equation (1.1), with initial condition (1.3) and the coefficient values indicated in the caption of Fig. 1. The NC is constructed as explained at the end of Section 1, with $M_0 = 300$ mesh points. The solution is illustrated in Fig. 1. Note that since ω_2/ω_1 is not rational, such solution is quasi-periodic for large time, and exhibits five oscillations in the considered time span. In order to analyze how the basic method works, we plot in Fig. 2 the result of applying the method with $\varepsilon = 10^{-3},\ \varepsilon_1 = \varepsilon/100$, and a length of the $I_{\rm NC}$ intervals $\delta_{\rm NC} = 0.5$, taking $N = 20$ snapshots in each interval. Only two $I_{\rm NC}$ intervals are needed, which gives a compression factor of 10; the numbers of modes in the two $I_{\rm NC}$ intervals are $(n, n_1) = (7, 9)$ and $(5, 8)$. The $I_{\rm NC}$ intervals are appreciated in Fig. 2(i) because the errors are set to zero in these intervals; since no Galerkin system is considered in these intervals, plot (ii) exhibits holes in them. The estimate $E_n^{n_1}$ is calculated in the $I_{\rm GS}$ intervals using (2.13) and the two errors appearing in this figure are calculated using (2.12) and taking as u a NC solution satisfying the same initial condition as that applied to the GS. Those results indicate that:

(a) The estimate $E_n^{n_1}$ of the L_2-error with n modes, $\text{Error}_{L_2}^{n}$, is quite good (both plots are indistinguishable in Fig. 2(i)). We intend to maintain this, which is the basic property for the validity of the method. Namely, the method will be considered to work well if $E_n^{n_1}$ and $\text{Error}_{L_2}^{n}$ are undistinguishable within plot accuracy except of course when both are much smaller than ε. Also, the L_2-error with n_1 modes, $\text{Error}_{L_2}^{n_1}$, is smaller than $\text{Error}_{L_2}^{n}$, which is a further indication of the consistency of the method. Namely, the approximation with n_1 modes is better than that with $n < n_1$ modes; when the method fails such consistency condition will be violated.

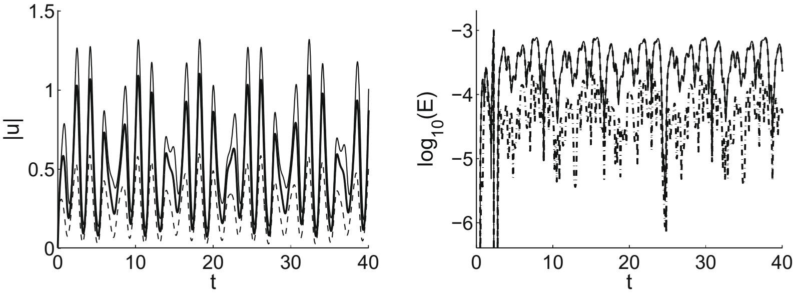
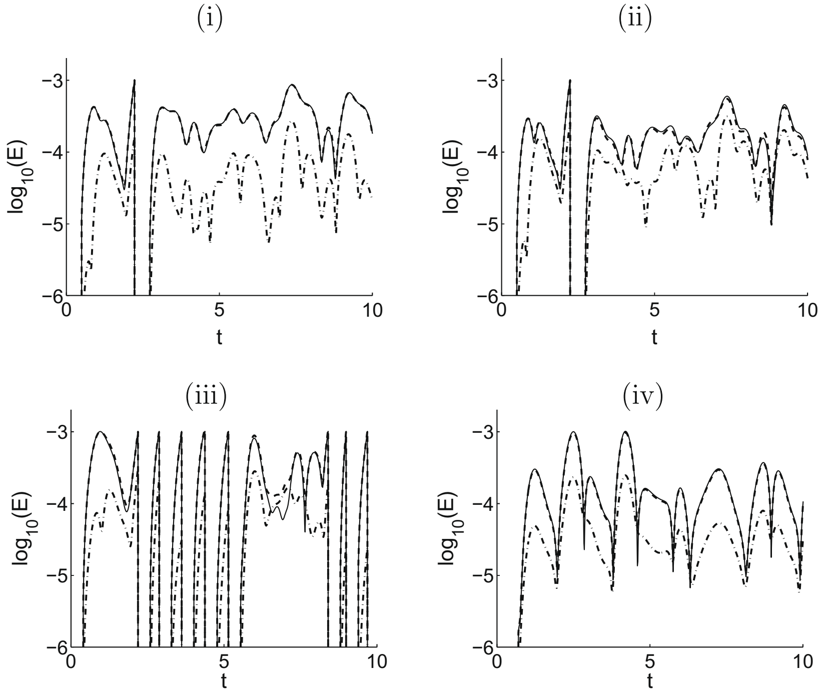


Fig. 3. As in Fig. 1 (left) and Fig. 2(i) (right), but with $T = 40$.


(i)

(ii)

(iii)

(iv)

Fig. 4. As in Fig. 2(i) ($\varepsilon = 10^{-3},\ \varepsilon_1 = \varepsilon/100,\ N = 20$, and $\delta_{\rm NC} = 0.5$), but with: (i) $N = 50$, (ii) $N = 16$, (iii) $\delta_{\rm NC} = 0.4$, and (iv) $\delta_{\rm NC} = 0.7$.

(b) The amplitude of the various modes oscillate around constant values (Fig. 2(ii)), indicating that the POD manifold is fairly stable as time proceeds. For comparison, we note that 24 and 105 standard Fourier modes are needed to approximate a typical spatial snapshot within a precision of $\varepsilon = 10^{-3}$ and $\varepsilon/100 = 10^{-5}$, respectively; note that in both cases, the number of Fourier modes is much larger than its counterpart for POD modes, namely n_1.

(c) POD modes (Fig. 2(iii)) exhibit standard properties of complete Galerkin basis, like Fourier basis or orthogonal polynomials. For instance, the number of zeroes increases with the order of the mode. This illustrates the power of POD, namely that POD modes exhibit a quite rich structure, which does not seem to be present in the apparently similar to each other snapshots (plotted in Fig. 2(iv)) that have been used to obtain the modes.

The POD manifold has been captured with only two DNS intervals. In fact, increasing the time span T does not require new DNS intervals. This is illustrated in Fig. 3, where the solution and the various errors are plotted in the time interval $0 < t < T = 40$, which gives a compression factor of 40.

The method turns out to be quite robust in connection with the various calibration parameters. In particular:

(1) The selected (after some calibration) number of snapshots, $N = 20 \sim 2n_1$, is larger than the number of required modes, n_1. Either increasing or slightly decreasing N (without violating the condition that $N \sim 2n_1$ or larger) does not produce major differences, as illustrated in Fig. 4(i)–(ii), where the cases $N = 50$ and 16 are considered. In fact, comparison with Fig. 2(i) shows that increasing N does not produce any benefit, but decreasing it slightly deteriorates the estimate of the error.

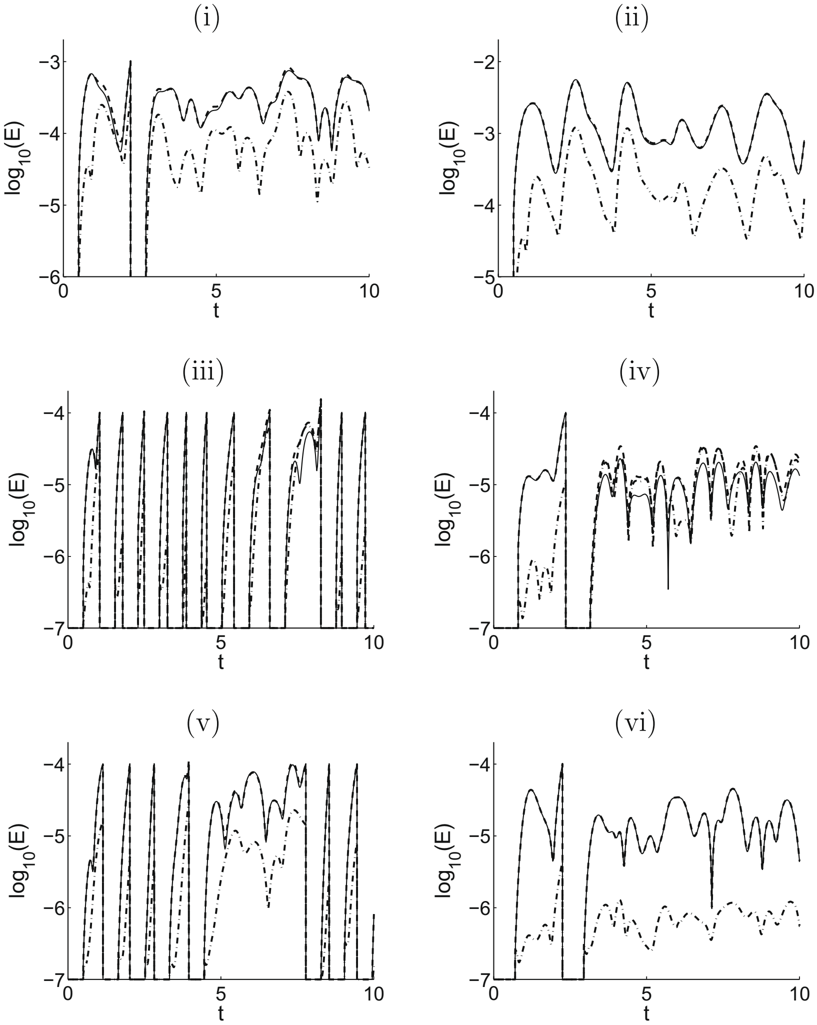


Fig. 5. As in Fig. 2 ($\varepsilon = 10^{-3}$, $\varepsilon_1 = \varepsilon/100$, $N = 20$, and $\delta_{NC} = 0.5$), but with: (i) $\varepsilon_1 = \varepsilon/20$, (ii) $\varepsilon = 0.01, \varepsilon_1 = \varepsilon/100$, $\delta_{NC} = 0.4$; (iii) $\varepsilon = 10^{-4}$, $\varepsilon_1 = \varepsilon/100$, $\delta_{NC} = 0.5$; (iv) $\varepsilon = 10^{-4}$, $\varepsilon_1 = \varepsilon/100$, $\delta_{NC} = 0.8$; (v) $\varepsilon = 10^{-4}$, $\varepsilon_1 = \varepsilon/1000$, $\delta_{NC} = 0.5$ and (vi) $\varepsilon = 10^{-4}$, $\varepsilon_1 = \varepsilon/1000$, $\delta_{NC} = 0.7$.

(2) Decreasing δ_{NC} to 0.4 (Fig. 4(iii)) worsens our error estimate and leads to the necessity of nine I_{NC} intervals, which decreases the compression factor to 2.77; (n, n_1) evolve in the nine I_{GS} intervals from (7,9) in the first interval to (4,6) in the last interval. The resulting I_{GS} intervals are now too small, which should be avoided. Increasing δ_{NC} to 0.7 (Fig. 4(iv)) instead requires only one I_{NC} interval, with $(n, n_1) = (7, 9)$ modes, and thus promotes an increase of the compression factor, which is now 14.28.

(3) Increasing ε_1 (but maintaining ε) (see Eq. (3.1)) decreases the numbers of modes n_1 and n, and slightly worsens our estimate of the error, as illustrated in Fig. 5(i), where the case $\varepsilon_1 = \varepsilon/20$ is considered, which requires $(n, n_1) = (6, 8)$ and (5,7) modes in the two I_{NC} intervals that are needed. Thus, the compression factor is now equal to 10.

(4) Requiring less precision allows to decrease δ_{NC}, as seen in Fig. 5(ii), where the case $\varepsilon = 0.01$ and $\delta_{\mathrm{NC}} = 0.4$ is considered which necessitates just one I_{NC} interval (a compression factor of 25) with $(n, n_1) = (5, 8)$ modes. If, instead, $\varepsilon = 10^{-4}$ (Fig. 5(iii)) and $\delta_{\mathrm{NC}} = 0.5$, then the number of needed I_{NC} intervals increases to twelve (which decreases the compression factor to 1.66), with numbers of modes oscillating between $(n, n_1) = (8, 10)$ and (7,15). Of course, the number of I_{NC} intervals can be decreased taking larger I_{NC} lengths, as shown in Fig. 5(iv), where $\delta_{\mathrm{NC}} = 0.8$, which involves just two I_{NC} intervals (a compression factor of 6.25), with $(n, n_1) = (8, 11)$ and (7,9) modes.

(5) Decreasing ε_1 (but maintaining ε) increases the numbers of modes n and n_1, and improves the accuracy of the error estimate. This is illustrated with the case $\varepsilon = 10^{-4},\ \varepsilon_1 = \varepsilon/1000$, and $\delta_{\mathrm{NC}} = 0.5$ (Fig. 5(v)), which requires 8 I_{NC} intervals (a compression factor of 2.5), with numbers of modes oscillating between (9,11) and (6,9). As above, the number of I_{NC} intervals decreases as δ_{NC} increases, which is illustrated in Fig. 5(vi), where $\delta_{\mathrm{NC}} = 0.7$ which necessitates only two I_{NC} intervals (which yields a compression factor of 7.14), with $(n, n_1) = (9, 12)$ and (7,11) modes.

(6) The factor 1/100 in (3.2) can also be changed to 1/50 or 1/200, with consistent results that are not plotted for the sake of brevity. As expected, increasing this factor too much worsens our error estimate, and decreasing it too much, increases the number of modes n_1, without any benefit for the method.

Summarizing the results above, the basic method easily provides compression factors of the order of 15–20 (in the interval $0 < t < 10$ if an RMS error of $\varepsilon = 10^{-3}$ is required) and provides a quite good estimate of the error of the GS-approximation. The method is quite robust concerning the various calibration parameters, namely: (a) the number of snapshots, provided that this be somewhat large compared to the number of modes ($N \sim 2n_1$ is a good choice for the NSNSF equation); and (b) the factors 1/100 appearing in (3.1) and (3.2). The length of the I_{NC} intervals instead, must be chosen with care since slight variations of this can produce large variations in the compression factor. Thus, appropriate selection of the length of these intervals might produce a significant improvement of the method. This is done in the next section, where a redefinition of the POD modes after each I_{GS} interval is also made.

4. Using I_{NC} intervals to just update the POD manifold

As already pointed out, the calculation of the POD manifold in the basic method described above begins from zero in each I_{NC} interval, ignoring the information about this manifold we already have. A first way of taking former information into account is to consider as snapshots both (a) those calculated in the last I_{NC} interval and (b) the newly calculated ones. It turns out that such strategy somewhat enlarges the I_{GS} intervals, as intended, but leads to a contamination of the POD manifold; namely, more POD modes are necessary to approximate the same trajectory. This is due to the fact that the POD manifold must approximate all snapshots in both the former and the new I_{NC} intervals, which means that the dimension of the resulting POD manifold increases until the orbit approaches an attractor. In order to avoid that, some filter must be included on former information about the POD manifold. This is done using the POD modes (instead of the snapshots) calculated in the last I_{NC} interval as modified snapshots for the new calculation of the POD manifold, and multiplying these modes by a weight, according to the associated mode amplitudes in the last I_{GS} interval. Specifically, instead of the snapshots defined above (see (2.1)), which were calculated directly from the NC solution, the POD modes for the next I_{GS} interval are calculated from the following *modified snapshots*

$$v_1, \ldots, v_{N_1}, w_1, \ldots, w_{N_2}, \tag{4.1}$$

which are defined as follows:

(a) The first N_1 modified snapshots are the following weighted modes

$$v_1 = \langle a_1 \rangle U_1^{\mathrm{GS}}, \ldots, v_{N_1} = \langle a_{N_1} \rangle U_{N_1}^{\mathrm{GS}}. \tag{4.2}$$

where $U_1^{\mathrm{GS}}, \ldots, U_{N_1}^{\mathrm{GS}}$ are the POD modes used in the last GS interval, and for $j = 1, \ldots, N_1$, the weight $\langle a_j \rangle$ is the following average of the associated POD-mode amplitude

$$\langle a_j \rangle = \frac{1}{\delta_{\mathrm{GS}}} \int_{t_1}^{t_1 + \delta_{\mathrm{GS}}} |a_j| \, dt. \tag{4.3}$$

Also, the sequence (4.2) is truncated ignoring those modes such that $\langle a_j \rangle$ is too small (say, smaller than ε_1). Thus, we require that

$$\langle a_j \rangle \geqslant \varepsilon_1 \quad \text{for } j = 1, \dots, N_1. \tag{4.4}$$

The weight (4.3) can be seen as a measure of the (square root of the) averaged energy of each mode in the last Galerkin interval. Without this weight, the less energetic modes in the last Galerkin interval would have the same contribution to the new POD manifold as the most energetic modes, and the resulting POD manifold would be increasingly contaminated, somehow as it happened when the new POD manifold was calculated using all available snapshots.

(b) The last N_2 modified snapshots are calculated from N standard snapshots (see (2.1)) computed in the new I_{NC}-interval, $u_1, \dots, u_N$, but modified as follows. First, we calculate the POD modes resulting from these snapshots, $U_1^{\text{NC}}, \dots, U_N^{\text{NC}}$, and then we consider the following modified snapshots

$$w_1 = \sigma_1 U_1^{\text{NC}}, \dots, w_{N_2} = \sigma_{N_2} U_{N_2}^{\text{NC}}, \tag{4.5}$$

where $\sigma_1, \dots, \sigma_{N_2}$ are the associated singular values and the sequence is truncated requiring that

$$\sqrt{\frac{1}{N} \sum_{j=N_2+1}^{N} (\sigma_j)^2} < \frac{\varepsilon_1}{100} = \frac{\varepsilon}{10000}. \tag{4.6}$$

(c) Once N_2 has been calculated, the sequence (4.2) is further truncated requiring that $N_1 < 1 + N_2/2$. In other words, we ignore one old POD mode for each pair of new POD modes that is added, which decreases the rank of the vector system (4.2), neglecting some of the less energetic old POD modes, on the assumption that the relevant dimension of the final POD manifold does not increase too much suddenly.

Now we are prepared to modify step ii in the basic procedure described at the beginning of Section 3 as follows:

(ii-a) At the end of the first I_{NC} interval we proceed as in the basic method.

(ii-b) At the end of each of the remaining I_{NC} intervals, we use as snapshots $v_1, \dots, v_{N_1}, w_1, \dots, w_{N_2}$, calculated as explained above, and proceed as explained in step iii.

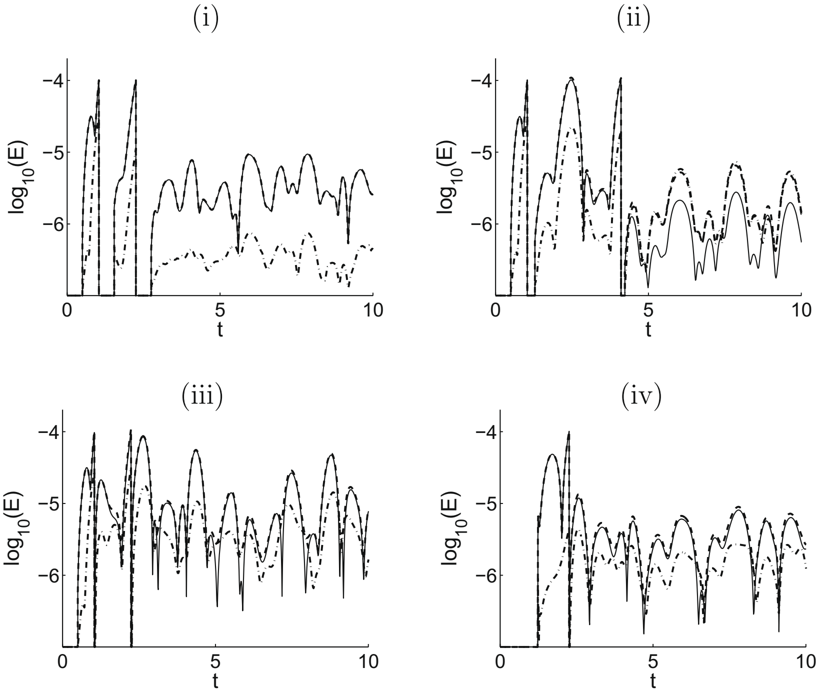


Fig. 6. As in Fig. 2(i) ($\varepsilon = 10^{-3}$, $\varepsilon_1 = \varepsilon/100$, and $N = 20$), but with $\varepsilon = 10^{-4}$ and: (i) the method with memory and $\delta_{\text{NC}} = 0.5$ fixed; (ii) the method with memory, dividing δ_{NC} by 2 in subsequent I_{GS} intervals; (iii) the method with memory and adaptive δ_{NC}, with $\delta_{\text{GS,min}} = 0.5$; (iv) the method with memory and adaptive δ_{NC}, with $\delta_{\text{GS,min}} = 1$.

The application of this new procedure is illustrated in Fig. 6(i), where the same parameter values as in Fig. 5(iii) are considered. Applying the modified method explained above requires only three-I_{GS} intervals, which gives a compression factor of 6.66, with $(n, n_1) = (8, 10)$, $(9, 11)$, and $(10, 12)$ modes; recall (see Fig. 5(iii)) that the basic procedure required 12 I_{GS} intervals and gave a compression factor of only 1.6. This clearly illustrates our main argument above, namely that if the information about the POD manifold we already have is appropriately used at the end of each Galerkin interval, then the new I_{NC} interval only needs to provide changes in the POD manifold. Thus, if the length of the I_{NC} intervals (δ_{NC}) is maintained, then the new modes calculated from the modified snapshots provide a better POD manifold and the number of required I_{NC} intervals decreases. Also, the I_{NC} interval can be shorter than before. In order to illustrate this, we maintain $\delta_{\text{NC}} = 0.5$ in the first I_{NC} interval, but divide it by 2 in each of the subsequent I_{NC} intervals; the number of snapshots in each of these I_{NC} intervals is also divided by 2. The result of applying this procedure is plotted in Fig. 6(ii), where the lengths of the three I_{NC} intervals are 0.5, 0.25, and 0.125, which gives a compression factor of 11.42, with the same numbers of modes as above.

This shows that it is convenient to decrease δ_{NC} after the first I_{NC} interval, which will be done in an adaptive way as follows. To begin with, since the number of snapshots that will be calculated in each I_{NC}-interval must be selected at the same time as the length of the interval, δ_{NC}, it is convenient to define the time interval between two consecutive snapshots, δ_{snaps}, which is taken as $\delta_{\text{snaps}} = \ell/N$, where ℓ is the characteristic time of the trajectory, namely the typical time interval in which $|u|$ varies in a quantity comparable to the typical total variation of $|u|$; in the case we are considering, Fig. 1 shows that $\ell \sim 0.5$, which taking $N \sim 2n_1 = 20$ as above, gives $\delta_{\text{snaps}} = 0.5/20 = 0.025$. Note that δ_{snaps} must be neither too small to avoid enlarging the number of snapshots without adding information about the dynamics, nor too large, to avoid enlarging the length of the I_{NC} intervals, which increases the computational cost. In addition, we select a minimum value of $\delta_{\text{GS}},\ \delta_{\text{GS,min}}$. Now, the length of each I_{NC} interval and the number of snapshots in the interval, will be calculated in an iterative way as follows:

(i) To begin with, the length of the I_{NC} interval in the iterative process is defined as $\delta_{\text{NC,init}} = N\delta_{\text{snaps}}$ (with $N \sim 2n_1$) in the first I_{NC} interval, and $\delta_{\text{NC,init}} = \delta_{\text{snaps}}$ in the remaining I_{NC} intervals.

(ii) Proceeding as in the former method, we use the δ_{snaps}-equispaced snapshots calculated in the I_{NC} interval to obtain the associated POD modes and launch a GS.

(iii) If the resulting value of δ_{GS} is larger than $\delta_{\text{GS,min}}$ then such interval is accepted.

(iv) If instead $\delta_{\text{GS}} < \delta_{\text{GS,min}}$, then we enlarge δ_{NC} according to the following simple formula

$$\delta_{\text{NC,new}} = \delta_{\text{NC,old}} + \max\left\{\delta_{\text{snaps}}, \frac{\delta_{\text{GS,min}} - \delta_{\text{GS}}}{\delta_{\text{GS,min}}}\delta_{\text{NC,old}}\right\}, \tag{4.7}$$

which (after imposing the natural requirement that $\delta_{\text{NC,new}} - \delta_{\text{NC,old}} \geqslant \delta_{\text{snaps}}$) relies on the rough assumption that the required value of δ_{NC} depends (locally) on δ_{GS} as a straight line of slope one; the slope could be of course calibrated, which will not be done below.

The resulting adaptive method is checked in Fig. 6(iii) with $\delta_{\text{GS,min}} = 0.5,\ \delta_{\text{snaps}} = 0.025$, and $N = 20$, which requires three I_{NC} intervals, with $(n, n_1) = (8, 10)$, (10,12), and (11,16), and $\delta_{\text{NC}} = 0.5$, 0.025, and 0.025; these give a compression ratio of 18.18. Note that the lengths of the second and third I_{NC} intervals are quite small, as anticipated; in fact, $\delta_{\text{NC}} = \delta_{\text{snaps}}$ in these intervals, which means that only one snapshot suffices to update the POD manifold in these two intervals, confirming our argument at the beginning of this section. Enlarging $\delta_{\text{GS,min}}$ enlarges the first I_{NC}-interval without significant improvements in the subsequent intervals, as illustrated in Fig. 6(iv), where $\delta_{\text{GS,min}} = 1$; now only two I_{GS} intervals are needed, with $(n, n_1) = (8, 11)$ and $(9, 12)$, and $\delta_{\text{NC}} = 1.123$ and 0.025, which gives a compression ratio of 8.71. The parameter $\delta_{\text{GS,min}}$ should be taken neither two small (to avoid a bad approximation) nor too large (to avoid generating too much information at the very beginning). A value of $\delta_{\text{GS,min}}$ of the order of the characteristic time of the equation is a good selection.

Now the method above, with adaptive I_{NC} turns out to be quite effective for the NSNSF equation. A first improvement of the method, dealing with calculation of the right hand side of the GS, is considered in next section.

5. Decreasing the number of points to calculate the GS

Calculation of the right hand side of the GS in (2.14) requires to compute the integrals appearing in (2.15), whose precise calculation involves all points in the mesh used to discretize the equation. As anticipated in Section 1, we expect that the right hand side of the GS (2.15) can be calculated using only information from a few number of points, M, somewhat larger than the number of modes involved, say $M \sim 2n_1$. In order to check this statement, we select M equispaced points in $0 < x < 1$ and replace the L_2-inner product (2.4) by

$$\langle u_1, u_2 \rangle = \frac{1}{M}\sum_{k=1}^{M} \bar{u}_1 u_2. \tag{5.1}$$

Such new inner product is used both to define the covariance matrix (2.3) and to calculate the right hand side of the GS, in (2.15), (2.17), and (2.18). Results obtained using this new inner product are presented in Fig. 7, where the cases already considered in Fig. 6(i), (iv) are recalculated using this new inner product with $M = 30$. Fig. 7 shows that both the number of I_{NC} intervals and their lengths remain unchanged, as do the number of modes used in each I_{NC}-interval. In fact, even the plots of

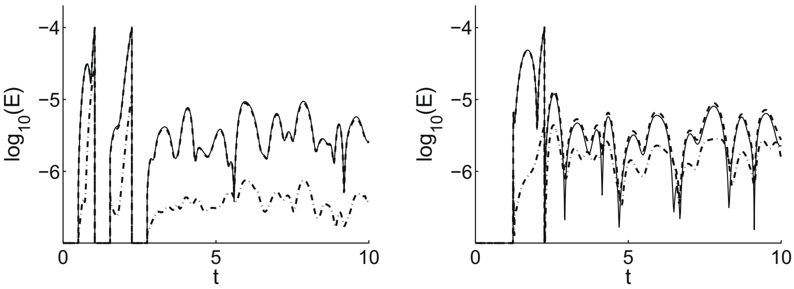


Fig. 7. As in Fig. 6(i) (left) and Fig. 6(iv) (right), but using the inner product (5.1), with $M = 30$ equispaced points.

the various errors remain almost identical, which shows that the simplification associated with using (5.1) does not involve any price in the method efficiency.

6. Using a second GS

After the improvements described in Sections 4 and 5, the basic method described in Section 3 turns out to be quite efficient for the NSNSF equation.

Let us now apply it to the CGL equation (1.4), with initial condition (1.5). It turns out that the method works quite well in some cases, but fails in some other cases, when transitions that are associated with high order modes occur, whose dynamics are not correctly described by the truncated Galerkin system. In fact, this is the same reason that promotes instability in standard postprocessed POD + Galerkin methods. As an example to illustrate this, we consider the CGL equation (1.4), with the parameter values indicated in the caption of Fig. 8. These parameter values have been carefully chosen precisely because they yield quite unstable transient dynamics. This is illustrated in Fig. 8, where results obtained with the NC, using $M_0 = 1000$ mesh points are given (in the time intervals $0 < t < 1$ and $0 < t < 5$) in plots (i)–(ii). The trajectory is seen to exhibit quite complex (chaotic-like) dynamics in the interval $0.3 < t < 2.5$, before decaying to a simpler quasi-periodic attractor, which is stable. If instead a mesh with $M_0 = 2000$ points is used (plots (iii-iv)), then (a) the trajectory remains near the former trajectory (with $M_0 = 1000$) for $0 < t < 0.45$, but both trajectories diverge from each other quite fast for $t > 0.45$; in fact, the transient chaotic period shortens with this new mesh, and the trajectory already decays to the quasi-periodic attractor at $t = 1.5$. The strong instability of the transient dynamics is further illustrated in plots (v)–(vi), where the $M_0 = 1000$ points mesh is used, as in plots (i)–(ii), but with a quite small term, $0.00001\sin(2\pi x)$, added to the initial conditions (1.5). Note that, again, the trajectory remains close to that in plots (i)–(ii) for $0 < t < 0.45$, but both trajectories diverge quite fast for larger values of t. All these suggest that the chaotic transient behavior observed above might be what is known as *transient chaos* [10], which is due to the existence of a heteroclinic-like nearby orbit that converges to a slightly unstable (hyperbolic) chaotic invariant set. If this were true, a chaotic attractor (yielding dynamics that are similar to those in the chaotic transient in plot (ii)) should exist for nearby values of the parameters. This is checked in plots (vii) and (viii), where the case $\mu = 85,\ \alpha = -2$, and $\beta = 14$ is considered, and the chaotic attractor is clearly appreciated.

For illustration, some representative snapshots of the spatial structure of the solution, at $t = 0.1, 1$, and 4, are given in terms of both $|u|$ and $\mathrm{Re}(u)$ in Fig. 9. Note that spatial complexity is not quite high; spatio-temporal complexity instead is high enough as to require a fairly large number of spatial modes, see below.

Now, in order to illustrate the failure of the *local POD + Galerkin projection method* developed in Sections 3–5, we apply this method to the CGL equation (1.4) with initial conditions (1.5) and the parameter values indicated in the caption of Fig. 8, using $M_0 = 1000$ mesh points, $M = 100,\ N = 100,\ \delta_{\mathrm{snaps}} = 0.0005,\ \delta_{\mathrm{GS,min}} = 0.1$, and $\varepsilon = 10^{-3}$. The evolution of $|u|$ at $x = 1/4, 1/2$, and 3/4, as in Fig. 1 and the counterpart of Fig. 2(i), are plotted in Fig. 10(i)–(ii), where it is seen that the method (which requires two I_{NC} intervals, with $(n, n_1) = (29, 38)$ and (28,41)) fails at $t = 0.45$, precisely at that value of t where the various trajectories plotted in Fig. 8 diverge from each other. This suggests that the reason for the failure of the method is that at that value of t dynamics exhibit a highly unstable transition, with new features that were not present in the snapshots calculated in the first I_{NC} interval.

Fortunately, as anticipated in Section 3, such difficulty can be solved using the same ideas behind the method. As we shall see, these unexpected transitions can be detected comparing the behavior of higher order modes resulting from two Galerkin systems, namely (a) the one considered above and (b) a second GS with $n_2 > n_1$ modes defined such that (cf. (3.1))

$$\sqrt{\frac{1}{N}\sum_{j=n_2+1}^{N}(\sigma_j)^2} < \frac{\varepsilon_1}{100} = \frac{\varepsilon}{10000}, \tag{6.1}$$

and, in addition to condition (3.3), requiring (in the definition of each I_{GS}-interval) that the POD amplitudes of this new GS be such that (cf. (2.13) and (3.3))

t

(1) A POD manifold calculated by the method of snapshots in each I_{NC} interval contains the dynamics in the next I_{GS} interval within an error bound ε provided that a higher accuracy (say, $\varepsilon/100$) is required in the calculation of the POD manifold in I_{NC}. This means that higher order modes contain enough information on the dynamics of lower order modes in the next future, which is not surprising because of causality: the whole solution would be completely determined by the equation and initial conditions if infinite precision were possible. Note that we are only claiming that the POD manifold contains an approximation of the exact solution, not the exact solution itself.

(2) Each I_{GS} interval finishes when the Galerkin system ceases to approximate the exact solution within an error bound ε, which is decided (without the need of calculating the exact solution) using an a priori error estimate based on higher order modes. This error estimate is based on the only (consistency) assumption that POD modes describe well the solution provided that sufficiently many modes are kept. Comparing with the exact error, the error estimate has been checked to provide quite good results, namely both the exact and the estimated errors coincide within plot accuracy; see Figs. 2(i), 3(ii), 4, and 5.

A basic method builded upon these principles was described in Section 3 that turned out to work well for the NSNSF equation, and to be quite robust against perturbations of the various calibration parameters, with one exception: small changes in the length of the I_{NC} intervals, δ_{NC}, produce large changes in the results. A better selection of δ_{NC} results from the various improvements that are recalled now:

(3) In the simplest method above, the POD manifold is completely calculated in each I_{NC} interval, ignoring any previous information. On the other hand, the expected continuous dependence of the POD manifold on time suggests that the POD manifold will suffer only a rotation between two consecutive I_{GS} intervals, and it is only such rotation (and not the whole POD manifold) that needs to be calculated in the next I_{NC} interval. Based on these ideas, a method has been constructed in Section 4 to use (after some preprocessing) the information of the POD manifold we already have in conjunction with the information in the snapshots calculated in the new I_{NC} interval, to calculate the new POD manifold in a computationally efficient way. In this way, all I_{NC} intervals, except for the first one, become very short; see Fig. 6(i)–(ii). In fact, the length of the I_{NC} intervals can be chosen in an automatic way, as explained in Section 4; see Fig. 6(iii)–(iv).

(4) In principle, the right hand side of the Galerkin system used in the I_{GS} intervals involve integrals that must be calculated in each time step of the numerical integration scheme, which can be a quite expensive process. This is usually solved in classical POD + Galerkin methods applied to equations exhibiting polynomial nonlinearities by some preprocessing to reduce the Galerkin system to a polynomial system whose coefficients can be calculated from the outset. But this restricts generality and can still be fairly expensive. Instead, we note that projecting on an n-dimensional POD manifold could be done using information on a number of points that is somewhat larger than n (say, $2n$), which in general is much smaller than the number of points of the computational mesh. This improvement has been introduced in Section 5, where it has been shown that leads to similar results as when all points in the computational mesh are used; see Fig. 7.

(5) The basic method, with the improvements outlined above is quite robust and computationally efficient when applied to the NSNSF equation (1.1), in which temporal complexity only results from time dependence of the nonlinear forcing term. In particular, our estimate $E_n^{n_1}$ of the error $E_{L_2}^n$ was quite good (both plots are undistinguishable in Figs. 2–6) except of course when these errors are much smaller than the error bound ε.

(6) Application to the CGL equation (1.4) instead produces no so good results. This is because of the inherent temporal complexity of this equation, whose dynamics can exhibit spontaneous transitions in some I_{GS} intervals associated with features that could be not present in the former I_{NC} intervals that were used to calculate the POD manifold. Such transitions can be detected using a second Galerkin system. The resulting method works quite well for the CGL equation; see Figs. 11(iii)–(iv) and 12. Again, our estimate $E_n^{n_1}$ of the error $E_{L_2}^n$ was quite good, see Figs. 11, 12.

The method has been developed somewhat empirically, from the basic ideas above, and checked in two one-dimensional parabolic problems, the NSNSF equation (1.1), which exhibits somewhat simple dynamics and the CGL equation (1.4), whose dynamics exhibit spontaneous instabilities. In fact, in the latter case, a set of values of the parameters was chosen in which the equation exhibited transient chaos, which is the most demanding situation to check the method. The results of the paper indicate that the method is both accurate and robust, and thus amenable to be applied to many related parabolic equations and systems, which is the object of current research, as is pursuing various improvements of the method, which are well ahead of the scope of this paper.

Acknowledgments

This research has been supported by the Spanish Ministry of Education, under Grants TRA2007-65699 and MTM2007-63204. We are also indebted to two anonymous referees for a careful reading of a previous version of the paper, and for some suggestions that helped to improve presentation of the results.

References

[1] D. Alonso, A. Velazquez, J.M. Vega, Robust reduced order modeling of heat transfer in a back step flow, Int. J. Heat Mass Transfer 52 (2009) 1149–1157.

[2] D. Alonso, A. Velazquez, J.M. Vega, A method to generate computationally efficient reduced order models, Comput. Methods Appl. Mech. Eng. 198 (2009) 2683–2691.

[3] I.S. Aranson, L. Kramer, The world of the complex Ginzburg–Landau equation, Rev. Mod. Phys. 74 (2002) 100–142.

[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] K. Bizon, G. Continillo, L. Russo, J. Smula, On POD reduced models of tubular reactor with periodic regimes, Comput. Chem. Eng. 32 (2008) 1305–1315.

[6] P.G.A. Cizmas, B.R. Richardson, T.A. Brenner, T.J. O'Brien, R.W. Breault, Acceleration techniques for reduced-order models based on proper orthogonal decomposition, J. Comput. Phys. 227 (2008) 7791–7812.

[7] M. Couplet, C. Basdevant, P. Sagaut, Calibrated reduced-order POD-Galerkin system for fluid flow modelling, J. Comput. Phys. 207 (2005) 192–220.

[8] E.H. Dowell, K.C. Hall, Modeling of fluid–structure interaction, Ann. Rev. Fluid Mech. 33 (2001) 445–490.

[9] D. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, SIAM, 1977.

[10] C. Grebogi, E. Ott, J.A. Yorke, Chaotic attractors in crisis, Phys. Rev. Lett. 48 (1982) 1507–1510.

[11] V. Kalb, A. Deane, An intrinsic stabilization scheme for proper orthogonal decomposition based on low dimensional models, Phys. Fluids 19 (2007) 054106-1–054106-18.

[12] T. Lieu, C. Farhat, M. Lesoinne, Reduced-order fluid/structure modeling of a complete aircraft configuration, Comput. Methods Appl. Mech. Eng. 195 (2006) 5730–5742.

[13] L.S. Lorente, J.M. Vega, A. Velazquez, Generation of aerodynamics databases using singular value decomposition, J. Aircraft 45 (2008) 1779–1788.

[14] D.J. Lucia, P.S. Beran, W.A. Silva, Reduced-order modelling: new approaches for computations physics, Prog. Aerospace Sci. 40 (2004) 51–117.

[15] D. Rempfer, Low-dimensional modeling and numerical simulation of transition in simple shear flows, Ann. Rev. Fluid Mech. 35 (2003) 229–265.

[16] H. Shahverdi, A.S. Nobari, M. Behbahani-Nejad, H. Haddadpour, An efficient reduced-order modelling approach based of fluid eigenmodes and boundary element method, J. Fluids Struct. 23 (2007) 143–153.

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

[18] S. Sirisup, G.E. Karniadakis, Stability and accuracy of periodic flow solutions obtained by a POD-penalty method, Physica D 202 (2005) 218–237.

[19] S. Sirisup, G.E. Karniadakis, D. Xiu, I.G. Kevrekidis, Equations-free/Galerkin-free POD assisted computation of incompressible flows, J. Comput. Phys. 207 (2005) 568–587.

[20] L. Sirovich, Turbulence and the dynamics of coherent structures, Quart. Appl. Math. XLV (1987) 561–590.

[21] J.R. Valdes, M.J. Miana, J.L. Nuñez, T. Pütz, Reduced order models for estimation of fluid flow and flow forces in hydraulic proportional valves, Energy Convers. Manage. 49 (2008) 1517–1529.

[22] D. Venturi, X. Wan, G.E. Karniadakis, Stochastic low dimensional modelling of a random laminar wake past a circular cylinder, J. Fluid Mech. 606 (2008) 339–367.