

A method to generate computationally efficient reduced order models

D. Alonso, A. Velazquez, J.M. Vega

Aerospace Propulsion and Fluid Dynamics Department, School of Aeronautics, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros 3, 28040 Madrid, Spain
Applied Mathematics Department, School of Aeronautics, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros 3, 28040 Madrid, Spain

ARTICLE INFO

Keywords:

Reduced order model
Proper Orthogonal Decomposition
Incompressible nonisothermal flow

ABSTRACT

A new method is presented to generate reduced order models (ROMs) in Fluid Dynamics problems. The method is based on the expansion of the flow variables on a Proper Orthogonal Decomposition (POD) basis, calculated from a limited number of snapshots, which are obtained via Computational Fluid Dynamics (CFD). Then, the POD-mode amplitudes are calculated as minimizers of a properly defined overall residual of the equations and boundary conditions. The residual can be calculated using only a limited number of points in the flow field, which can be scattered either all over the whole computational domain or over a smaller projection window. This means that the process is both computationally efficient (reconstructed flow fields require less than 1% of the time needed to compute a full CFD solution) and flexible (the projection window can avoid regions of large localized CFD errors). Also, various definitions of the residual are briefly discussed, along with the number and distribution of snapshots, the number of retained modes, and the effect of CFD errors, to conclude that the method is numerically robust. This is because the results are largely insensitive to the definition of the residual, to CFD errors, and to the CFD method itself, which may contain artificial stabilizing terms. Thus, the method is amenable for practical engineering applications.

1. Introduction

The interest in the development of ROMs is twofold. The academic interest in solving complex nonlinear multi-parametric problems in a computationally efficient way, using a limited number of modes comes first. Second, the resulting ROM allows for the generation of fast methods to solve problems of practical engineering and academic interest. In fact, computational efficiency is becoming crucial nowadays to improve design cycles, saving both design cost and time to market. This is evident in the specialized literature where a continuously increasing number of methods and applications are published every year.

An example of the potential of reduced order approaches has been recently reported by Lieu et al. [1]. In this article, the authors apply a POD-based method to the aeroelastic modeling of a full F-16 fighter aircraft configuration in the transonic regime. In particular, they claim that their Mach-based flutter analysis is robust and accurate enough to think of future near real-time analysis of complete aircraft configurations. It is worth noting that the CPU time needed in this work for ROM calculations was of the order of 5% of the time needed to generate a new snapshot. This factor suggests the potential savings that could be achieved using this method. In a different work, Cizmas et al. [2] have explored several

acceleration techniques based on POD. Basically, the techniques include a splitting of the snapshots database, which only require solving quasi-symmetrical matrices, and a strategy to reduce the frequency of projection. The method was applied to the problem of a 2-D fluidized bed. Depending on the acceleration technique, the authors reported a ROM speed-up factor ranging from 20 to 100 when compared with the full model, while keeping similar accuracy. In the context of searching for computational efficiency, Shaverdi et al. [3] have developed a ROM approach based on the boundary element method for aerodynamic applications. The authors deal with several test cases including a wing-body combination in heaving motion and reported computational times of the order of 15% of those obtained with other ROMs. Some other examples of ROM development in various engineering fields have been considered by Valdes et al. [4] (hydraulics) and Bizon et al. [5] (chemical processes). In these two articles, the authors focused on the accuracy of their method but, unfortunately, no information was provided on the associated computational efficiencies.

Attempts to automatize the generation of ROMs are also being performed. The underlying idea is to provide means that simplify the formulation while retaining as much generality as feasible. In this context, Aquino [6] has presented an object-oriented design for reduced order modeling, and has investigated object-oriented design attributes and their capability to generate efficient models. Nie and Joshi [7] have presented a method based on a flow network modeling. Here, several ROMs that apply at different spatial scales

are assembled together intending an accurate description of a complex multiscale thermal system. Development of robust ROM formulation is also important in related control applications. Examples of reduced-order based control strategies have been reported, among others, by Ravindran [8] to study flow separation past a forward-facing step, and by Bergmann and Cordier [9] to minimize aerodynamic drag in the flow past a circular cylinder when the cylinder portion is taken as the control parameter.

Many industrial applications involve complex Fluid Dynamics problems whose direct numerical simulation is well beyond present computer capability. Turbulence models and other related simplifications still require spatial discretization with thousands to millions of mesh nodes and some artificial stabilizing terms to avoid numerical instability. Steady state situations are of interest either because the industrially relevant solutions are genuinely steady, as in the example considered below, or because of turbulence modeling simplifications. The objective of this paper is to develop a methodology, already anticipated in [10], to obtain ROMs to calculate steady states of multi-parametric fluid problems. The method itself is based on a flexible minimization strategy rather than on conventional Galerkin projection, which is more appropriate to deal with unsteady solutions. The main new ingredient results from the observation that the residual to be minimized can be calculated using a limited number of nodes in the fluid domain. The method is tested to conclude that results are largely insensitive to the definition of the residual, to the various parameters (number of snapshots and number of retained modes) that are present, to peculiarities of the CFD method (artificial stabilization terms can be ignored in our formulation) and to CFD errors (the method improves CFD results). Results are sensitive instead to the selected snapshots in the parameter plane, which is also illustrated. Thus, the method is both numerically robust and computationally inexpensive, and thus amenable to engineering applications. Regarding organization of the article, the method is presented in Section 2, a test problem is described in Section 3, and results are given and discussed in Section 4, followed by conclusions, in Section 5.

2. Method description

Galerkin projection is a standard method to obtain POD-based ROMs of time-dependent problems. In this case, the ROM is obtained projecting the exact equations over the previously computed POD manifold, obtained from a set of CFD-calculated snapshots. Also, nonhomogenous boundary conditions must be transformed into homogenous ones, in order that they can be assumed to be identically satisfied by both the snapshots and all POD modes. Thus, neither boundary conditions nor continuity equation, which is also homogenous in the incompressible case, need to be further considered. Still, POD modes are defined as *joint modes* for all state variables. After that, Galerkin projection is equivalent to obtaining the time derivatives of the POD-mode amplitudes as minimizers of a properly defined quadratic residual of the remaining equations. Several difficulties arise that are currently the object of active research in the literature: (1) Effective computations on the ROM require performing various integrals over the whole fluid domain that can be numerically expensive when a large amount of mesh nodes are present. (2) CFD calculated snapshots may exhibit large concentrated errors due to, for instance, singularities near the boundary (caused by, e.g., corners), whose elimination can be either numerically expensive or impossible when computability limits are reached. Such errors produce (unavoidable, in this setting) errors in the ROM. (3) CFD codes may exhibit artificial stabilizing terms that are introduced only for numerical reasons, and could need to be accounted for in the

ROM. Dependence of the ROM on these unphysical terms is undesirable. (4) Galerkin-based models are frequently unstable due to neglected higher order modes, which is a major weakness of the method.

The main object of this paper is to develop a new method, already anticipated in [10], which is intended to solve the above difficulties, all at a time, in multi-parametric steady state situations. This requires revising the basic concepts of reduced order modeling, seeking as much flexibility as possible. For the sake of clarity, we explain here the main ideas, with a reasonable generality, anticipating some of properties that will be quantitatively illustrated in Section 4.

We consider a set of m partial differential equations and n boundary conditions, involving m state variables (such as velocity components and pressure)

$$E_j(q_1, \dots, q_m) = 0 \text{ in } \Omega, \quad BC_k(q_1, \dots, q_m) = 0 \text{ at } \partial\Omega \quad (1)$$

for $j = 1, \dots, m$ and $k = 1, \dots, n$, where $\partial\Omega$ is the boundary of the computational domain Ω , and both the equations and boundary conditions may depend on various parameters such as the Reynolds number or a velocity component at a portion of the boundary.

2.1. POD modes for each state variable

Now, we use a CFD code to calculate N_0 solutions of (1), with state variables q_{j1}, \dots, q_{jN_0} (calculated for various sets of parameter values), which will be called *snapshots*. Using these, we calculate for each state variable, the associated POD modes, denoted as Q_{j1}, \dots, Q_{jN_0} , with $Q_{jk} = \sum_{i=1}^{N_0} \alpha_{jk}^i q_{ji}$, where for each j , α_{jk}^i are the eigenvectors of the covariance matrix R^j , defined as $R_{kl}^j = \langle q_{jk}, q_{jl} \rangle$, where $\langle \cdot, \cdot \rangle$ is an appropriately defined inner product. Namely

$$\sum_{i=1}^{N_0} R_{kl}^j \alpha_{jm}^i = (\sigma_{jm})^2 \alpha_{jm}^k.$$

The square roots of the associated eigenvalues of R^j , $\sigma_{j1}, \dots, \sigma_{jN_0}$, are called *singular values*. Now, for each state variable, we define the number of retained modes, N_j , in such a way that

$$\sqrt{\frac{(\sigma_{j1})^2 + \dots + (\sigma_{jN_j})^2}{N_j}} < \varepsilon_0 \quad (2)$$

for some pre-determined error bound ε_0 . This means, invoking well known POD formulae, that after truncation to N_j modes the average error of reconstructing all snapshots in each state variable is bounded by ε_0 . Errors are defined here using the norm $\|q\| = \sqrt{\langle q, q \rangle}$. Now, we expand each state variable in terms of its retained modes, as

$$q_j = \sum_{k=1}^{N_j} A_{jk} Q_{jk}. \quad (3)$$

Using independent POD modes for each flow variable allows us to simultaneously impose all equations and boundary conditions, which cannot be done in standard Galerkin-type reduced order equations.

2.2. POD+Interpolation

Truncating the expansions (3) we can reconstruct any of the original snapshots. When the set of snapshots correspond to parameter values that conform a Cartesian mesh, a first approximation of the steady state for other parameter values could be obtained using either the *Proper Orthogonal Decomposition POD+Interpolation* [11] or *High Order Singular Value Decomposition (HOSVD)+Interpolation* method [12], which consist of applying either standard POD or HOSVD (an extension of SVD to tensors [13]) to the set of snaps

and then interpolate in the parameters. Here, we use a mixed strategy. When the snaps conform a Cartesian mesh in the parametric plane, we apply SVD to the POD-mode amplitudes in this two-parameter plane and perform one-dimensional interpolation on the SVD-modes associated with each parameter. When the parameter mesh is not Cartesian and SVD is not applicable to the POD-mode amplitudes, direct interpolation is made using the Akima's bivariate interpolation method for scattered data [14]. Since both SVD/POD and interpolation are numerically inexpensive, the method only requires about 6 CPU seconds in a desktop PC (3.20 GHz and 2 Gb RAM) to calculate each case.

The result of POD plus interpolation will be used as an initial guess in the residual minimization method that is considered next.

2.3. Overall residual

The POD-mode amplitudes are calculated minimizing an overall residual of the equations and boundary conditions, defined as

$$R = \sum_{j=1}^m \sqrt[p]{\frac{1}{N_E} \sum_{k=1}^{N_E} |E_j(x_k, y_k)|^p} + \sum_{j=1}^n \sqrt[p]{\frac{1}{N_{BC}} \sum_{k=1}^{N_{BC}} |BC_j(x_k, y_k)|^p}. \quad (4)$$

where the first sum is extended to the N_E mesh nodes in the computational domain, the second sum, to the N_{BC} mesh points in the boundary of the computational domain and $p > 0$. Computation of the residual in Eq. (4) requires to calculate partial derivatives of the state variables. In our case, these derivatives are readily obtained using second-order finite differences in the numerical scheme described in [15,16].

Minimizing the residual could in principle provide spurious solutions, which would be the counterparts of those obtained using Galerkin projection, as mentioned in Section 1. But such difficulty has been never encountered using the method proposed in this paper.

Now, these definitions involve all mesh nodes and can lead to fairly expensive calculations if N_E is large. In order to avoid calculating the whole residual in each step of the iterative optimization process we proceeded in [10] as it is usually done in Galerkin methods, namely using a preprocessing that allows to replace (4) by the polynomial expression in the unknown POD-amplitudes that is obtained replacing (3) into (4); the coefficients of such polynomial are integrals over the whole computational domain that are calculated from the outset. Obtaining standard polynomials means that absolute values must be avoided in (4), which requires that the exponent p must be even. The smallest possible value of p is 2, which means that in Fluid Dynamics problems (noting that momentum equations are quadratic) the preprocessed polynomial is fourth order; in fact, this requires some additional ingredients (see [10]) when momentum equations are cubic, as it happens when viscosity or thermal conductivity are temperature dependent [17]. This means that a large number of coefficients are present, which makes preprocessing numerically expensive. Larger values of p involve even more expensive calculations.

In order to avoid such difficulty, we note that the basic idea of the method is that all snaps are well approximated by a small number of modes, which suggests that the residual could be calculated using only information from an also small number of nodes, somewhat larger than the expected number of POD modes. This observation is new in the context of reduced order modeling and can be justified noting that if all solutions were exactly contained in the POD manifold and calculations were exact, then N_E could be taken (generically) equal to the dimension of the manifold, in order to have as many equations as the number of unknowns. Since calculations are not exact, N_E must be larger than the number of modes, but not necessarily equal to the number of mesh nodes,

which depends on the CFD method, not in the POD approximation itself. The required number of nodes should only depend on both the number of modes and the maximum distance of the snaps to the POD manifold. Of course, the selected nodes must bear enough information about the solution that is being approximated, which can be ensured checking that redundancy of information in the selected nodes be not too large; for instance, selecting all nodes in fluid regions where the flow is almost parallel is not expected to produce good results in problems that also exhibit nonparallel flow behavior. The idea could be illustrated assuming that the number of modes is one. The POD manifold is then a straight line in a N_E -dimensional space, whose determination should only require considering its projection on a low-dimensional subspace of the N_E -dimensional space.

This idea suggests taking only a limited number of nodes (points in the computational domain) in (4), scattered either over the whole domain Ω or over a portion of it, which will be called *projection window* below. As it will be shown, results are fairly independent of both the projection window and the number of selected nodes, with only weak limitations. And the computational cost for calculating, e.g., all results reported in Table 3, is low since this requires 42 CPU seconds when using, for instance, 51 nodes in the projection window.

In this context, the parameter N_{BC} could be also decreased as we did with N_E , but this would not reduce computational time because N_{BC} is usually quite small compared to N_E . In fact, in our test problem the boundary conditions are such that selecting one single point for each boundary condition is enough to impose the boundary condition.

Now, the residual defined above can be minimized using various methods. Genetic Algorithms exhibit the advantage of being robust, which is convenient in the present paper, where the GA described in [10] is used. Of course, gradient based methods such as steepest descend would provide much faster versions of our ROMs. However, since we focus on the ROM methodology it was decided to use a robust, conventional approach for the minimization algorithm, with a conservative selection of GA specifications. Improvement of the minimization method is the subject of on going research.

3. Problem description

To illustrate our method, we have selected as a test problem the *nonisothermal (heated from the lower wall) flow past a backwards facing step* in the steady regime (see Fig. 1); see [10,15,16] for a more detailed description of the problem and the CFD method. Here, we just mention that the mesh is Cartesian, with 32,051 nodes; dimensionless horizontal and vertical distances between nodes are both equal to 0.02.

This problem is characterized by a *recirculation bubble* that appears behind the step. We consider temperature dependent viscosity and thermal conductivity to account for the fact that large variations (of the order of 300% in the former, see reference [17]) occur in the temperature range considered here, from 293 K to 353 K; Reynolds number varies from 50 to 250.

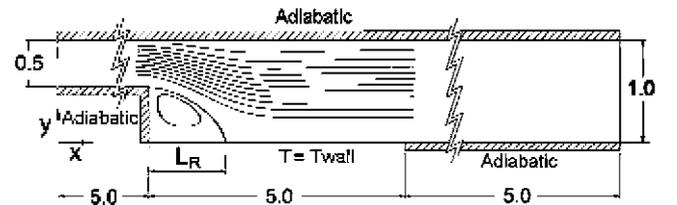


Fig. 1. Sketch of the computational domain.

If the width of the channel is large compared to depth, then the flow topology can be considered 2-D because, as shown in Refs. [18,19], the onset of 3-D instabilities starts at a Reynolds number in the range from 700 to 1000, depending on the remaining parameters. Dimensionless continuity, momentum, and energy equations are

$$\partial_x u + \partial_y v = 0, \quad (5)$$

$$u\partial_x u + v\partial_y u + \partial_x p - \text{Re}^{-1}[\mu\Delta u + 2\partial_x \mu \partial_x u + \partial_y \mu (\partial_y u + \partial_x v)] = 0, \quad (6)$$

$$u\partial_x v + v\partial_y v + \partial_y p - \text{Re}^{-1}[\mu\Delta v + 2\partial_y \mu \partial_y v + \partial_x \mu (\partial_y u + \partial_x v)] = 0, \quad (7)$$

$$u\partial_x T + v\partial_y T - (\text{RePr})^{-1}[\kappa\Delta T + \partial_x \kappa \partial_x T + \partial_y \kappa \partial_y T] = 0. \quad (8)$$

These are the exact equations that will be considered below. The CFD method instead considers a pseudo-evolution version of these equations (which consists of adding new terms, $\varepsilon_1 \partial_t p$, $\partial_t u$, $\partial_t v$, and $-\partial_t T$, with $\varepsilon_1 > 0$ small, in the left hand sides of Eqs. (5)–(8), respectively). In addition, in order to avoid numerical instabilities, four stabilizing terms are added in the left hand sides Eqs. (5)–(8), namely $\varepsilon_2 \Delta p - \varepsilon_3 \Delta^2 p$, $\varepsilon_4 \text{Re}^{-1} \Delta^2 u$, $\varepsilon_4 \text{Re}^{-1} \Delta^2 v$, and $\varepsilon_5 (\text{RePr})^{-1} \Delta^2 T$, respectively, with $\varepsilon_2, \dots, \varepsilon_5 > 0$ small [20]. Each CFD run, to obtain each steady state, typically (depending on the flow parameters) requires a CPU time of the order of 6 h on a desktop PC.

The spatial variables, velocity, and pressure are rendered dimensionless with the hydraulic diameter of the inlet section, the inlet average velocity, and the dynamic pressure, $2\tilde{h}^{\text{inlet}}$, \tilde{U}^{inlet} , and $\rho^{\text{inlet}}(\tilde{U}^{\text{inlet}})^2$, respectively. Nondimensional temperature is defined as $T = (T - \tilde{T}^{\text{inlet}})/(\tilde{T}^{\text{max}} - \tilde{T}^{\text{inlet}})$, where the superscript max stands for the maximum temperature imposed at the nonadiabatic part of the lower wall (353 K). Δ stands for the Laplacian operator and Reynolds and Prandtl numbers are defined as $\text{Re} = 2\rho^{\text{inlet}}\tilde{h}^{\text{inlet}}\tilde{U}^{\text{inlet}}/\tilde{\mu}(\tilde{T}^{\text{inlet}})$ and $\text{Pr} = \tilde{c}_p\tilde{\mu}(\tilde{T}^{\text{inlet}})/\tilde{\kappa}(\tilde{T}^{\text{inlet}})$. Functional relations for the dimensionless viscosity and thermal conductivity are obtained by polynomial fitting of the experimental data provided in Ref. [17], as

$$\mu = \tilde{\mu}(\tilde{T})/\tilde{\mu}(\tilde{T}^{\text{inlet}}) = 1 - \mu_1 T + \mu_2 T^2, \quad (9)$$

$$\kappa = \tilde{\kappa}(\tilde{T})/\tilde{\kappa}(\tilde{T}^{\text{inlet}}) = 1 + \kappa_1 T - \kappa_2 T^2, \quad (10)$$

where $\mu_1 = 1.1292$, $\mu_2 = 0.4904$, $\kappa_1 = 0.1572$, and $\kappa_2 = 0.0470$. At the inlet and outlet sections we impose a Pouseuille-like flow and a stress-free condition, respectively. Namely

$$u(y) = -24\left(y^2 - \frac{y}{2}\right), \quad v(y) = 0, \quad \partial_x p = \frac{-48}{\text{Re}}, \quad T = 0, \quad (11)$$

$$\partial_x u - \partial_x v - \partial_x^2 p - \partial_x T = 0. \quad (12)$$

No-slip ($u = v = 0$) is imposed at solid walls and pressure is computed solving an approximate formulation of momentum equations with one sided derivatives (into the flow domain). Wall temperature is prescribed

$$T = T^{\text{wall}}. \quad (13)$$

in the region $y = 0, 5 < x < 10$ (see Fig. 1), while the remaining part of the lower wall and the upper wall are both considered to be adiabatic.

As formulated, the test problem above depends on two nondimensional parameters, the wall temperature and the Reynolds number, which are assumed to vary in the range $T^{\text{wall}} \in [0, 1]$, $\text{Re} \in [50, 250]$. On the other hand, three parameters have been selected to compare ROM and CFD solutions. Heat transfer from the nonadiabatic part of the wall is measured by the *Nusselt number*, which is defined in terms of the dimensional heat flux, \tilde{Q}' , as

$$\text{Nu} = \frac{\tilde{Q}'}{2(\tilde{T}^{\text{max}} - \tilde{T}^{\text{inlet}})\tilde{\kappa}(\tilde{T}^{\text{wall}})\tilde{h}^{\text{inlet}}} = \int_5^{10} \partial_y T(x, 0) dx. \quad (14)$$

The *reattachment length*, L_R , of the separated flow region behind the step (see Fig. 1) gives an idea of the flow topology and is defined using that point over the horizontal line of the mesh just above the lower wall in which the horizontal velocity is zero. The pressure drop, P_D , is a measure of the required pump power, which is defined as the difference between the average pressures at the inlet and outlet sections.

4. Results

To assess the capability of the method, we have calculated 25 snapshots using CFD and have selected five test points, PT1, ..., PT5, where velocity, pressure and temperature fields will be reconstructed. The first four test points are located near the corners of the parametric plane (to ensure including the most problematic points), namely at $(T^{\text{wall}}, \text{Re}) = (0.875, 75)$, $(0.875, 225)$, $(0.125, 75)$, and $(0.125, 225)$. The fifth point, at $(T^{\text{wall}}, \text{Re}) = (0.375, 175)$, is located somewhat near to the center, to include a representative point of the interior of the parameter plane. A visual impression of the parametric plane, the computed snapshots, and the selected test points is presented in Fig. 2.

In order to have a reference of the results obtained below, we give in Fig. 3 contours of the local residual of the CFD solution on continuity and x-momentum equations, at point PT3; residuals at the remaining points and the remaining equations are similar. Note that large errors are concentrated near the upper part of the step, which are due to the singularity associated with the step-corner.

We have applied to this test problem both CFD and the *POD+Interpolation* method described in Section 2.2. Results are shown in Table 1 where differences in percentage between the two methods are provided. Errors lower than 0.1% are rounded hereafter to 0.0. Note that, when using 25 snapshots, *POD+Interpolation* predict the reattachment length with errors smaller than 2% all over the parametric plane. Pressure drop and Nusselt number relative errors deteriorate (relative errors of the order of 10%) at low Re, at points PT1 and PT3, which is due to the fact that dependence of pressure drop on Re is strongly nonlinear at low Re. In fact, the ability of the method to detect CFD errors would be a surplus, which will be pursued below. When using only nine snapshots, the results (including the reattachment length) degrade consistently. Errors are of the order of 10% except in the case of the pressure drop, which can be as high as 22%. Also, it seems that instead of using a Cartesian grid of snapshots in the parametric space, it would be better to concentrate them in a more efficient way, using, e.g., an a priori error estimate.

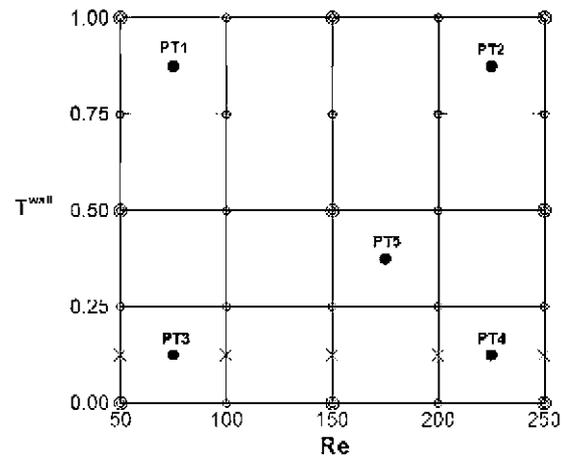


Fig. 2. Sketch of the parametric domain.

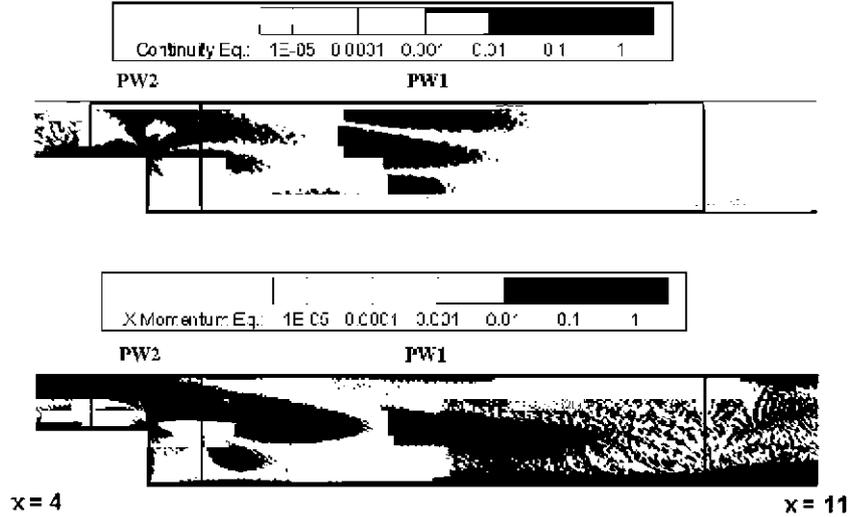


Fig. 3. Contours of the local residuals (namely, the absolute value of the left hand side of the equation) of CFD solution at point PT3 in continuity (upper figure) and x -momentum (lower) equations.

With these results in mind, the object of the remaining of this section is to illustrate the effect on the results of the various parameters appearing in the method described in Section 2. Summarizing the method and the initial selection of the parameters, which has been made after some preliminary calibration, we retain six POD modes in each flow variable (which leads to a total number of $6 \times 4 = 24$ mode amplitudes) from the set of 25 snapshots, introduce the expansion (3) into Eqs. (5)–(8) and boundary conditions (11)–(13), and minimize the overall residual (4) (with $p = 1$ except in Section 4.3) using a Genetic Algorithm, whose specifications are (see [10] for more details): 5000 individuals with a discretization of 10 bits per individual, 200 generations, 1.33% of elite individuals going into the next generation, crossover probability equal to 0.8, mutation probability at bit level equal to 0.042, and span allowed around the POD initial solution equal to 50%.

According to our description of the method in Section 2, the following specifications of the method are investigated in the following sections, 4.1–4.6: The position of the projection window, the number of points inside the window, the value of p used in the formula (4) to calculate the residual, the number of retained POD modes, the number of snapshots, and the specifications of the Genetic Algorithm. In addition, the ability of the method to improve CFD results that show numerical errors will also be checked.

4.1. The projection window

In principle, we should apply POD and calculate the residual using Eq. (4) in the whole computational domain, which contains 32,051 points. Since the residual must be calculated a large number of times by the Genetic Algorithm, a good question is whether

information from a smaller part of the computational domain would be enough. In order to answer this question, two projection windows PW1 and PW2 are considered (see Fig. 3) both to calculate the POD modes and the overall residual. PW1 is a rectangular window located downstream of the step, excluding a corner region, where CFD errors are expected to be largest and including that region that is heated from below. The window sides are $(x,y) \in [5.5, 10.0] \times [0.02, 0.98]$. This window contains 11,074 mesh nodes. PW2 is a window, $(x,y) \in [4.5, 5.5] \times [0.02, 0.98]$, located right on the corner region, which contains 1849 mesh nodes.

Results using these two projection windows are given in Table 2 and show that POD+Interpolation results in Table 1 are clearly improved by the ROM. In fact using the whole computational domain provides errors that are within 3% in all cases except for the Nusselt number at point PT3, which is calculated with a 11.1% error. The reason for such larger error at this point will be analyzed in Section 4.6 below. Also, accuracy degrades only slightly when the projection window PW1 is used: errors are within 3% except at point PT3. Computational time using the projection window PW1 decreases since only 11,074 points are involved, instead of the 32,051 points of the whole domain. Using PW1, each computation only requires 10 min in a desktop personal computer (3.20 GHz and 2 Gb RAM). Results are consistently better in case PW1 than in case PW2. This is due to the fact that the projection window PW1 contains at least a part of the recirculation bubble and thus bears more information about the flow topology than window PW2; also, PW1 excludes the region where CFD errors are larger, see Fig. 3. This must be taken into account when selecting the projection window. Finally, it is remarkable that using a projection window, we are obtaining the solution in the whole computational

Table 1

CFD-results and relative errors resulting from POD+Interpolation over the six first modes.

	CFD results			POD+Interpolation error (in%)					
	L_R	P_D	Nu	25 snapshots			9 snapshots		
				L_R	P_D	Nu	L_R	P_D	Nu
PT1	1.26	3.59	4.48	1.5	8.6	0.6	12.7	22.2	7.1
PT2	2.76	0.98	6.15	0.0	1.0	0.1	0.0	11.2	0.1
PT3	1.16	3.71	0.54	0.0	8.6	16.7	10.3	22.3	13.0
PT4	2.62	1.01	0.90	0.8	0.0	1.1	0.0	11.9	2.2
PT5	2.24	1.38	2.47	0.0	2.2	0.8	3.5	13.0	0.0

Table 2

Relative errors (in%) at test points obtained using all mesh nodes in the computational domain and in the projection windows PW1 and PW2. We retain six POD modes in each flow variable obtained from 25 snapshots.

	Whole domain			PW1			PW2		
	L_R	P_D	Nu	L_R	P_D	Nu	L_R	P_D	Nu
PT2	0.7	1.0	0.1	0.0	1.0	0.0	2.2	13.3	0.2
PT3	0.0	0.8	11.1	1.8	1.1	20.3	5.2	1.9	18.5
PT4	2.3	0.0	1.1	2.3	1.0	3.3	2.3	4.0	1.1
PT5	0.0	2.1	0.8	0.0	1.4	0.8	5.4	8.7	0.4

domain minimizing the residual only in a (small) part of it. This opens the possibility of obtaining the modes from a sparse set of data, which has obvious consequences in various aspects of engineering applications. Thus, after having shown the advantages of using a limited projection window, all calculations below are made in projection window PW1.

4.2. Number of nodes inside the projection window

As an additional step to save computational time, we now check whether the number of nodes in the projection window can still be decreased, considering the following numbers of equispaced nodes in projection window PW1: 84, 51, and 26, reducing the original number of nodes by a factor of 121, 169, and 289, respectively. Selecting equispaced nodes makes an essential difference with collocation methods, in which nodes position is selected to minimize the error. Here, instead, error minimization is implicit in both POD and the residual minimization, and the selection of the nodes position is unessential, provided that they involve nonredundant information. Results are shown in Table 3. Comparison with Table 2 indicates that, excluding again Nusselt number at point PT3, no loss of accuracy results from decreasing the number of nodes in the projection window. Moreover, the number of nodes needed by our ROM is really small. Actually, it suffices that this number be somewhat larger than the number of unknowns in the problem, namely the number of POD mode amplitudes (i.e., $6 \times 4 = 24$ in the present case). The computational cost is consistently decreased, since, e.g., 53, 42, and 34 seconds are enough to compute each case using 84, 51, and 26 nodes, respectively. A conservative criterion has been used to assess convergence of the Genetic Algorithm. In particular, all cases have been run for a fixed number of generations that overshoots the one needed for the worst case. This means that information provided about computational cost is conservative but, on the other hand, data about the different cases can be compared on the same basis.

4.3. The formula used to calculate the residual

Now, we consider the effect of calculating the residual using Eq. (4) with $p = 2$ and 4, and the same ROM parameters as in Table 3, where the value $p = 1$ was used; the latter results are also included to facilitate comparison. Relative errors are given in Table 4, where it is seen that results degrade as p increases. This is because at large p the contribution to the residual of that part of the domain where the equations are better satisfied is masked by the contribution of those zones where the residual is larger. For instance, if the residual shows a maximum value of 0.01 in (say) zone A, and values of the order of 0.001 in zone B, when the fourth power of all these is taken and the contributions of all nodes are added, contributions from zone B are just too small, and are ignored. Thus, the Genetic Algorithm improves the residual only in zone A and the resulting minimum of the residual provides a poor approximation. This happens with pressure drop at point PT1, where large errors already appeared associated to the use of POD+Interpolation (Table 1). For-

Table 3
As in Table 2, but minimizing the residual in the indicated number of nodes in the projection window PW1.

	84 nodes			51 nodes			26 nodes		
	L_R	P_D	Nu	L_R	P_D	Nu	L_R	P_D	Nu
PT1	0.0	0.3	0.9	0.0	1.9	0.9	1.6	2.5	0.7
PT2	0.7	1.0	0.0	0.7	1.0	0.0	0.0	2.0	0.0
PT3	1.7	1.0	18.5	1.7	0.3	16.7	1.8	0.3	13.0
PT4	2.3	1.0	1.1	2.3	1.0	0.0	2.3	1.0	2.2
PT5	0.0	1.4	0.8	0.0	2.2	1.2	0.0	1.4	0.8

Table 4
Relative errors (in%) obtained calculating the residual using formula (5), with $p = 1, 2$ and 4, using 84 points in projection window PW1.

	$p = 1$			$p = 2$			$p = 4$		
	L_R	P_D	Nu	L_R	P_D	Nu	L_R	P_D	Nu
PT1	0.0	0.3	0.9	0.0	1.7	1.1	1.6	8.9	0.7
PT2	0.7	1.0	0.0	0.7	2.0	0.0	0.7	2.0	0.0
PT3	1.7	1.0	18.5	1.7	0.5	14.8	0.0	0.3	18.5
PT4	2.3	1.0	1.1	2.3	1.0	1.1	2.3	2.0	2.2
PT5	0.0	1.4	0.8	0.0	1.4	0.8	0.0	1.4	0.8

unately, results with $p = 2$ are acceptable, which is good to apply minimization methods (such as steepest descend) that require that the residual be a smooth function of the mode amplitudes, which does not occur at $p = 1$. The main conclusion is that the exponent $p = 1$ is a good selection when using a Genetic Algorithm, and will be maintained below.

4.4. The number of modes in each flow variable

Let us now consider the influence of the number of modes in each flow variable (recall that the number of POD-mode amplitudes is four times larger). As above, projection window PW1 is used with 84 projection points. Results are shown in Table 5, where comparison with results obtained using POD+Interpolation is also made. These results show that the ROM is always better than POD+Interpolation, especially in connection with pressure drop. As expected, our ROM results deteriorate when the number of modes decreases. On the other hand, some of the results (e.g., Nusselt number at PT1) deteriorate and some other ones improve only slightly when the number of modes increases from 6 to 7. Excluding points PT1 and Point PT3, which exhibit an anomalous behavior to be analyzed in Section 4.6, we conclude that six modes

Table 5
As in Table 4 ($p = 1$), retaining the indicated number of modes.

	ROM (seven modes)			POD+Interpolation (seven modes)		
	L_R	P_D	Nu	L_R	P_D	Nu
PT1	0.0	2.2	6.7	0.0	8.7	0.9
PT2	0.7	1.0	0.2	0.0	1.0	0.2
PT3	1.7	1.3	25.9	0.0	8.6	17.7
PT4	0.8	1.0	2.2	0.8	1.0	2.2
PT5	0.0	2.2	0.9	0.0	2.2	0.9
	ROM (six modes)			POD+Interpolation (six modes)		
PT1	0.0	0.3	0.9	1.6	8.7	1.1
PT2	0.7	1.0	0.0	0.0	2.0	0.0
PT3	1.7	1.0	18.5	0.0	8.6	20.4
PT4	2.3	1.0	1.1	0.8	1.0	0.0
PT5	0.0	1.4	0.8	0.0	1.4	0.8
	ROM (five modes)			POD+Interpolation (five modes)		
PT1	0.0	2.5	0.7	1.6	8.7	1.1
PT2	1.4	1.0	0.0	1.4	2.0	0.0
PT3	0.0	1.3	24.1	1.7	8.6	20.3
PT4	3.8	1.0	1.1	3.8	1.0	0.0
PT5	0.9	2.2	0.9	0.9	2.2	0.9
	ROM (four modes)			POD+Interpolation (four modes)		
PT1	1.6	1.1	0.4	3.2	9.2	0.2
PT2	1.4	1.0	6.9	1.4	1.0	0.7
PT3	3.4	1.1	25.9	1.7	7.8	11.1
PT4	2.3	1.0	13.3	4.2	1.0	20.0
PT5	0.9	2.2	2.8	0.9	2.2	7.7
	ROM (three modes)			POD+Interpolation (three modes)		
PT1	17.5	3.1	1.3	0.0	9.7	0.4
PT2	0.7	2.0	7.3	0.7	1.0	0.7
PT3	0.0	0.8	25.9	5.2	7.8	11.1
PT4	2.3	1.0	10.0	3.8	1.0	20.0
PT5	0.0	0.7	3.6	0.0	0.7	7.3

yield errors comparable to CFD errors. Thus, we can say that the six modes case is converged with regard to this parameter, and that increasing the number of modes may worsen results. This could be surprising at first sight but is readily explained noting that errors are usually relegated by POD to higher order modes, which is the underlying idea on using POD to cleaning noisy databases and images.

Instead of selecting the same number of modes for all flow variables, we can proceed as indicated in Section 2.1. Namely, the number of modes in each flow variable is selected such that condition (2) is fulfilled, with $\varepsilon_0 = 10^{-4}$. Such condition is met taking 6, 8, 4, and 6 modes for the horizontal velocity, vertical velocity, pressure, and temperature, respectively. Results with these numbers of modes using both our ROM and POD+Interpolation as shown in Table 6, where it is seen that the former are exact within 2%, except for the Nusselt number at point PT3. The advantage now is that the number of modes is automatically selected once the error bound ε_0 has been selected.

4.5. The number of snapshots used to compute POD modes

Let us now vary the number of snapshots. This parameter is very important when devising practical engineering applications because, after the improvements introduced above, the time needed to reconstruct a flow field using the ROM is but a tiny fraction (less than 1%) of the time required to compute each snapshot. In order to analyze this effect, we compare in Table 7 the results shown above (using 25 equispaced snapshots) with those obtained using nine equispaced snaps (marked with a double circle in Fig. 2). Comparison with Table 6 shows that accuracy deteriorates as the number of snapshots is reduced, as expected. But it is note worthy that the ROM still proves to be superior to the POD plus interpolation approach with nine snapshots. This can be improved with a small number of additional snapshots, as shown in Table 7, where calculations with nine snaps are repeated using either (i) two additional snapshots, at points (100,0.25) and (100,0.75) and (ii) four additional snapshots at points (100,0.25), (100,0.75), (200,0.25) and (200,0.75). Note (see Fig. 2) that the additional snaps have been selected in the vicinity of the problematic points, PT1 and PT3.

It is quite remarkable how adding just two conveniently selected snaps improves the results so much (except for Nusselt calculation at point PT3, which is considered in the next section). Moreover, this opens the possibility of trying to design a method of selection of snaps that allows obtaining results comparable to those in Table 3 using a much smaller number of snaps, which is crucial to dramatically save computational time. But this is outside the scope of this paper.

4.6. Improving CFD results with errors: analysis of points PT1 and PT3

Let us now concentrate on the ROM calculation of points PT1 and PT3. A first attempt to improve these results consists of adding five new snapshots, at those points denoted with crosses in Fig. 2,

Table 6
As in Table 5, but using 6, 8, 4, and 6 modes for u , v , p , and T .

	ROM			POD+Interpolation		
	L_R	P_D	Nu	L_R	P_D	Nu
PT1	1.6	2.2	0.7	1.6	9.2	1.0
PT2	0.0	1.0	0.0	0.0	1.0	0.0
PT3	0.0	0.8	16.7	3.4	1.1	9.3
PT4	0.8	1.0	0.0	0.0	2.0	0.0
PT5	0.0	2.2	0.8	0.0	2.2	0.8

Table 7

Relative errors (in%) resulting from three sets of snapshots, using six POD-modes in each flow variable in the projection window PW1 with 84 points.

	L_R	P_D	Nu	L_R	P_D	Nu
	ROM (9 snaps)			POD+Interpolation (9 snaps)		
PT1	12.7	2.5	3.8	11.1	18.9	3.1
PT2	0.0	3.2	0.2	0.0	11.2	0.2
PT3	6.9	1.6	29.6	5.2	19.7	24.1
PT4	2.3	1.0	3.3	2.3	22.8	3.3
PT5	3.6	0.7	2.8	3.6	13.0	0.0
	ROM (11 snaps)			POD+Interpolation (11 snaps)		
PT1	4.8	0.8	1.8	1.6	10.3	1.8
PT2	0.0	1.0	0.0	0.7	23.4	0.0
PT3	0.0	0.3	13.0	1.7	10.8	18.5
PT4	2.3	3.0	1.1	2.3	23.8	3.3
PT5	4.5	0.0	0.0	3.6	12.3	0.4
	ROM (13 snaps)			POD+Interpolation (13 snaps)		
PT1	3.2	0.6	1.8	3.2	0.6	1.8
PT2	0.7	2.0	0.0	0.7	2.0	0.0
PT3	0.0	0.0	13.0	0.0	0.0	18.5
PT4	0.0	1.0	1.1	0.0	1.0	3.3
PT5	0.9	1.5	0.0	0.9	1.5	0.4

namely at $T^{wall} = 0.125$ and $Re = 50, 100, 150, 200,$ and 250 . Modes are selected as in Table 6 and results are given in Table 8. Comparison with Table 6 shows that prediction of Nusselt number at point PT3 has only slightly improved. At this point, a doubt arises on the precision of CFD results at PT3, which leads us to plot in Fig. 4 (solid line) the CFD values of Nusselt versus Reynolds number at $T^{wall} = 0.125$. Such plot shows an oscillatory slope, which is suspicious. CFD errors may be due to the fourth order stabilizing term

Table 8

Relative errors (in%) resulting from two sets of snapshots, using (6,8,4,8)-POD-modes in each flow variable in the projection window PW1 with 84 points.

	ROM (30 snaps)			POD+Interpolation (30 snaps)		
	L_R	P_D	Nu	L_R	P_D	Nu
PT1	0.0	1.1	0.9	1.6	7.8	0.9
PT2	0.7	0.0	0.2	0.7	0.0	0.2
PT3	0.0	0.8	11.3	1.7	6.2	11.3
PT4	0.0	1.0	3.3	0.0	1.0	3.3
PT5	1.1	1.0	1.7	3.3	0.5	1.5

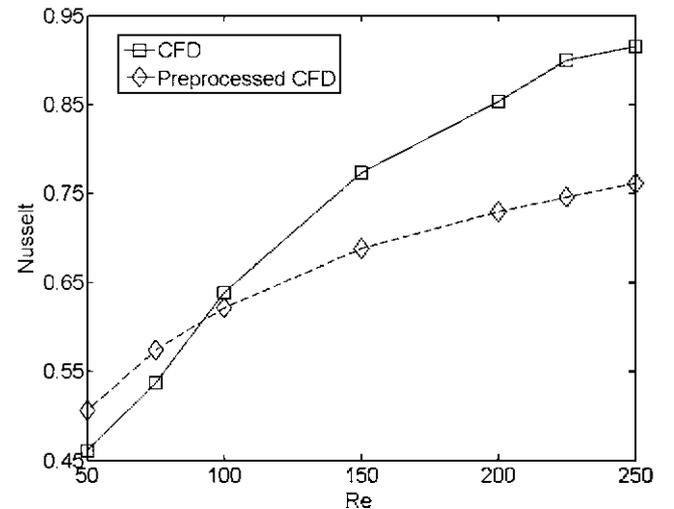


Fig. 4. Nusselt number versus Reynolds number resulting from CFD calculations, both with the code used in the remaining of the paper (solid) and with post-processing on the energy equation (dashed).

in the energy Eq. (9), which is especially dangerous at this low temperature. Thus, we recalculate the temperature profiles at $T^{wall} = 0.125$ integrating only the energy equation, without any stabilizing term, using the velocity field provided by the former CFD calculation. Results are plotted with dashed lines in Fig. 4 and confirm that former CFD results were not accurate enough, as suspected. Note that now the plot is concave.

The new CFD values of Nusselt number at points PT3 and PT4 are 0.57 and 0.90 respectively, which means that the errors of the ROM results at points PT3 and PT4 must be recalculated. It turns out that the large error at point PT3 has disappeared, but a new large error on Nusselt number is encountered at point PT4, as could have been anticipated noting the large deviations between both curves in Fig. 4 at large Re.

The reason for that large error at point PT4 is only due to the fact that the window span for the genetic algorithm (namely, the window where the genetic algorithm is allowed to search) is centered at POD amplitude values that are too far away from the true minimum. In order to ascertain that, we repeat the ROM calculation at point PT4, with the same 30 snaps as in Table 8 but with the span window centered at the projection on the POD manifold of the corrected CFD solution, which gives the results in Table 9. Note that these are quite good. Thus, we conclude that the manifold also contains the solution at PT4 but, due to the constrictions of the minimization method, we were not able to find the correct values for the amplitudes. For completeness, such new window is also applied using the 25 snaps already used in Table 6 (namely, ignoring the new five snaps at $T^{wall} = 0.125$), which provides the fairly reasonable results given in Table 9.

All this means that the ROM exhibits unexpected advantages. Namely, all results on the ROM in Table 9 have been obtained using the original CFD-calculated snaps, which exhibit errors at low temperature due to the unphysical stabilization terms. In spite of that, ROM results are quite good. This means that the ROM could improve the CFD results for practical engineering problems.

As it happened when using a few points in a small projection window to minimize the residual, this unexpected advantage of the ROM is new to our knowledge in this context and could be surprising at first sight. But in fact, the advantage becomes more natural when realizing that POD modes are only used to (i) obtain a good POD manifold; the exact equations themselves are imposed again when (ii) the residual is minimized. Thus, errors in the snaps only affect step (i), while the actual approximation is calculated in step (ii). One can also think that Fourier modes do not bear any information about the solution that is being approximated (they only bear information about the fact that the solution is periodic); the equations themselves are imposed when calculating the actual Fourier expansion, which is the counterpart of our step (ii).

4.7. Genetic Algorithm specifications

Finally, we test the behavior of the genetic algorithm itself and run a series of cases, G2 in Table 10, with the following specifications: 15,000 individuals with a discretization of 10 bits per indi-

Table 9

Relative errors (in %) resulting from using the ROM with the span allowed to genetic algorithm centered at the amplitudes of projection of CFD over the POD manifold, with the indicated number of snaps using the same parameters as in Table 8.

	L_R	P_D	Nu	L_R	P_D	Nu
PT4	ROM (30 snaps)			Projection of CFD on POD manifold		
	1.5	2.0	2.7	0.0	2.0	9.5
PT3	ROM (25 snaps)			Projection of CFD on POD manifold		
	1.7	1.1	5.3	3.4	1.1	5.3
PT4	0.0	1.0	2.7	0.0	2.0	10.8

Table 10

Influence of the Genetic Algorithm specifications on relative errors (in%). The ROM parameters are the same as in Table 5 with 6 modes.

	Genetic Algorithm G1			Genetic Algorithm G2		
	L_R	P_D	Nu	L_R	P_D	Nu
PT1	0.0	0.3	0.9	0.0	1.9	1.1
PT2	0.7	1.0	0.0	0.7	1.0	0.0
PT3	1.7	1.0	18.5	1.7	1.0	20.3
PT4	2.3	1.0	1.1	2.3	1.0	1.1
PT5	0.0	1.4	0.8	0.0	1.4	0.8

Table 11

Standard deviation results obtained from running the Genetic Algorithm 50 times. ROM parameters are as in Table 5 with six modes.

	Standard deviation		
	L_R	P_D	Nu
PT1	0.020	0.005	0.014
PT2	0.004	0.006	0.001
PT3	0.011	0.005	0.025
PT4	0.006	0.000	0.013
PT5	0.003	0.001	0.001

vidual, 200 generations, 1.33% of elite individuals going straight into the next generation, crossover probability equal to 0.8, mutation probability at bit level equal to 0.014, span allowed around the POD initial solution equal to 50%.

These new cases have been computed using the same ROM specifications as in Table 5, with six modes. Results obtained using the new genetic algorithm specifications (column G2) are compared to those already presented in Table 5 (column G1). Note that the more strict specifications of case G2 do not generate significant changes in the results, which means that results were converged with the former GA specifications used along the paper. In addition, we have performed one more test that consists of repeating 50 times the computations presented in column G1 of Table 10. Each of these new 50 computations has been started using a different seed for the random number generator needed for the Genetic Algorithm. The standard deviation around the mean value obtained for the various flow parameters and tests points is given in Table 11.

5. Conclusions

We have presented a new method for constructing ROMs to calculate steady states of complex fluid flows in a multi-parametric setting. The main idea behind the method was given in Section 2, and developed in Section 4 using a test problem. The following overall conclusions/remarks are in order:

- Alternative methods can be constructed that are more appropriate than standard Galerkin to calculate steady states. The main weaknesses of the latter can be overcome, revising the ideas behind the concept of POD-based ROMs. In particular:
 - More flexibility is convenient in the definition of the POD modes themselves, treating the various flow variables independently, to naturally increase the number of free POD-mode amplitudes and allow improving both continuity equation and boundary conditions.
 - Minimizing a properly defined, overall residual is a good way to treat steady problems (and, we believe, unsteady problems as well, which might allow better methods than Galerkin's).
 - The definition of the residual is crucial to improve both precision and computational cost. Also, localized CFD errors, which are frequently present in industrial calculations, can be dealt with in an efficient way.

- Artificial stabilizing terms, required by CFD computations, have been completely ignored above. Instead, the exact equations and boundary conditions have been considered to calculate the ROM. Thus, our ROM is robust, namely it is independent of the way snapshots have been calculated.
- The following argument is based on simple Linear Algebra concepts and helps to understand some of the simplifications introduced in the paper. Namely, that even though the full system is governed by partial differential equations, which involve infinite dimensional functional spaces, the set of relevant solutions is well approximated by a small number of modes. This means, in particular, that calculation of the approximate solution for a given set of parameter values does not need information from all mesh nodes resulting from CFD discretization; instead, information from a few, properly selected nodes is enough. Such number of nodes is only a few times larger than the number of total POD-mode amplitudes, as expected. This principle has been checked in this paper, and has allowed both a good precision of the ROM and inexpensive computations. In fact, this is the main conclusion of the paper, namely that *it is feasible to compute the modes using only information from a limited flow field region*. Reconstruction of the full flow field using this has an error comparable to that obtained using complete information. Now, both the projection window and the specific nodes that are used to calculate the residual are subject only to weak limitations, which are good news in connection with application of the method to other fluid problems. For instance, projection windows near the entrance and the exit of the computational domain, where the flow is almost parallel, have been avoided; the region of the recirculation bubble (projection window 2, see Fig. 3) instead would be preferred. And a limited number of equispaced nodes (without further requirements) give good results. Thus, thinking of applications to other fluid problems, we can expect that placing the projection window in that part of the fluid domain that show more structure (which can be usually decided a priori, with only a qualitative knowledge of the expected solution) and selecting equispaced nodes should be enough to obtain good results.
- Selection of the snapshots is a critical issue since CFD calculation of these is the most expensive part of the process. Here, we have shown that appropriate selection of the snaps, away from a standard Cartesian mesh in the parameter space, could reduce the number of required snaps. Furthermore, we believe that it should be possible to design a method to select the snapshots in such a way that only a few of them are enough, if properly selected. Its number should be just somewhat larger than the number of POD modes. The method would provide a dramatic reduction in computational time, since this is essentially associated with CFD; the remaining calculations in our method are quite inexpensive after the improvements introduced above. Such method is well outside the scope of this paper, and the object of our current research.
- The results above also open the possibility of deriving the whole flow field using information from a small part of it. This means, for instance, that databases of practical engineering interest could be reconstructed out of gappy experimental data.
- Some preliminary results on related problems suggest that the efficiency of the method increases as the number of parameter increase, which is quite promising in connection with industrial applications.
- Finally, and somewhat unexpectedly, our analysis (in Section 4.6) of the anomalous behavior of point PT3 has shown that our ROM could improve the precision of the CFD results for prac-

tical engineering problems. Namely, the ROM produces quite reasonable results even though POD modes are calculated using contaminated CFD-computed snapshots. This could have been expected noticing that precision in snapshots is not crucial since, at the end, the true equations are used by our ROM to obtain the approximated solution. Such unexpected advantage of our ROM is of great interest in industrial applications, since industrial CFD codes frequently exhibit large errors due to time and cost constraints. But this is ahead of the scope of the paper, and again the object of our current research.

Acknowledgement

This research was supported by the Spanish Ministry of Education and Science (MEC) under Projects DPI2005-05572 and TRA2007-65699. We are indebted to two anonymous referees for their suggestions on an earlier version of the paper that helped to improve presentation of the results.

References

- [1] T. Lieu, C. Farhat, M. Lesoinne, Reduced-order fluid/structure modeling of a complete aircraft configuration, *Comput. Methods Appl. Mech. Engrg.* 195 (2006) 5730–5742.
- [2] P.G.A. Cizmas, B.R. Richardson, T.A. Brenner, T.J. O'Brien, R.W. Breaull, Acceleration techniques for reduced-order models based on Proper Orthogonal Decomposition, *J. Comput. Phys.* 227 (2008) 7791–7812.
- [3] H. Shaverdi, A.S. Nobari, M. Behbahani-Nejad, H. Haddapour, An efficient reducer-order modelling approach based of fluid eigenmodes and boundary element method, *J. Fluid Struct.* 23 (2007) 143–153.
- [4] J.R. Valdes, M.J. Miana, J.L. Nuñez, T. Putz, Reduced order models for estimation of fluid flow and flow forces in hydraulic proportional valves, *Energy Conv. Manag.* 49 (2008) 1517–1529.
- [5] K. Bizon, G. Continillo, L. Russo, J. Smula, On POD reduced models of tubular reactor with periodic regimes, *Comput. Chem. Engrg.* 32 (2008) 1305–1315.
- [6] W. Aquino, An object-orientated framework for reduced-order models using Proper Orthogonal Decomposition (POD), *Comput. Methods Appl. Mech. Engrg.* 196 (2007) 4375–4390.
- [7] Q. Nie, Y. Joshi, Reduced order modeling and experimental validation of steady turbulent convection in connected domains, *Int. J. Heat Mass Transfer* 51 (2008) 6063–6076.
- [8] S.S. Ravindran, Control of flow separation over a forward facing step by model reduction, *Comput. Methods Appl. Mech. Engrg.* 191 (2002) 4599–4616.
- [9] M. Bergmann, L. Cordier, Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reducer-order models, *J. Comput. Phys.* 227 (2008) 7813–7840.
- [10] D. Alonso, A. Velazquez, J.M. Vega, Robust reduced order modeling of heat transfer in a back step flow, *Int. J. Heat Mass Transfer* (2009) 1149–1157.
- [11] T. Bui-Thanh, Proper Orthogonal Decomposition extensions and their applications in steady aerodynamics, Master Thesis, Singapore-MIT Alliance, 2003.
- [12] L.S. Lorente, J.M. Vega, A. Velazquez, Generation of aerodynamics databases using Singular Value Decomposition, *J. Aircraft* 45 (2008) 1779–1788.
- [13] L. de Oathauer, B. de Moor, J. Vandewalle, On the best rank-one and rank- (R_1, R_2, \dots, R_N) approximation of higher order tensors, *SIAM J. Matrix Anal. Appl.* 21–24 (2000) 1324–1342.
- [14] H. Akima, A method of bivariate interpolation and smooth surface fitting for irregular distributed data points, *ACM Trans. Math. Softw.* 4 (1978) 148–164.
- [15] B. Mendez, A. Velazquez, Finite point solver for the simulation of a 2-D laminar incompressible unsteady flows, *Comput. Methods Appl. Mech. Engrg.* 193 (2004) 825–848.
- [16] A. Velazquez, J.R. Arias, B. Mendez, Laminar heat transfer enhancement downstream of a backward facing step using a pulsating flow, *Int. J. Heat Mass Transfer* 51 (2008) 2075–2089.
- [17] F.P. Incropera, D.P. Dewitt, *Introduction to Heat Transfer*, John Wiley and Sons, 1996. Table A-6, p. 764.
- [18] L. Kaiktsis, G.E. Karniadakis, S.A. Orzag, Onset of the three dimensionality, equilibria and early transition in flow over a backward facing step, *J. Fluid Mech.* 231 (1991) 501–558.
- [19] D. Barkley, M.G.M. Gomes, R. Henderson, Three dimensional instability in flow over a backward facing step, *J. Fluid Mech.* 473 (2002) 167–190.
- [20] B. Mendez, A. Velazquez, Finite point based numerical study on the unsteady laminar wake behind square cylinders, *Int. J. Numer. Methods Heat Fluid Flow* 17 (2007) 108–132.