SELF ADAPTIVE POD BASED ROM AEROELASTIC SIMULATIONS

Rubén Moreno-Ramos[1], Fernando Varas[2], José M. Vega[2]

[1]Altran Innovación, Aeronautics Space and Defence Division
C/ Campezo, 1, E-28022, Madrid, Spain
ruben.morenoramos@altran.com

[2]E.T.S.I. Aeronáutica y del Espacio, Universidad Politécnica de Madrid
Pza. Cardenal Cisneros, E-28040, Madrid, Spain

Keywords: Reduced Order Models, Computational Aeroelasticity

Abstract: This paper presents the application of a self adaptive POD based ROM to unsteady aerodynamic prediction in a computational aeroelasticity framework. The method is first presented and then an application to aeroelastic simulations is performed for inviscid flows. In order to reduce the computational cost, while maintaining representativity to industrial environments, the simulations have been performed on an airfoil instead of a lifting surface. The results show very good potential for such simulations and even higher gains are expected when moving to three-dimensional simulations

1 INTRODUCTION

The prediction of steady forces over lifting surfaces and aerodynamic bodies by means of computational fluid dynamics (CFD) is becoming a standard in the aerospace industry. It allows for accurately predicting the vehicle lift, drag, loads and handling qualities, among other quantities. Moreover, this is done with an affordable computational cost and the support of a decreasing number of testing campaigns, needed to tune some solver parameters.

The use of CFD to predict unsteady aerodynamic forces is also intended as an alternative to current methods used in production tasks, such as the doublet lattice method (DLM), which is based on potential theory and does not capture nonlinear/viscous effects appearing in detached flows and shock waves. For dynamic aeroelastic simulations, however, unsteady CFD requires a large computational effort and its accuracy is not guaranteed by the large majority of current solvers given the complexity of the flow, as highlighted in recent meetings on the subject [1]. The research community is making a tremendous effort on improving the unsteady CFD solvers available to the aeroelastic engineers. In some cases, good correlation with wind tunnel tests has been reported [2], but the computational cost still prevents them from being implemented in industry due to the large number of simulations that are needed to account for different combinations of the parameters defining the vehicle configuration and flight conditions. Increasing attention is being paid to the use of *reduced order models* (ROMs) on aeroelastic calculations. However, current ROMs for aeroelasticity still rely on the linearisation of the aerodynamic forces. The resulting linearized equations are treated using various reduced order modeling methods, such as proper orthogonal decomposition (POD) [3], Volterra kernels [4] and frequency response functions [5]. Thus, these linearized models still exhibit some of the drawbacks mentioned above in connection with DLM.

Fully nonlinear reduced order models have already been developed in various fields, including aerodynamics. These models take advantage of the well-known fact that, in extended systems (modeled by partial differential equations and systems), the number of *physically relevant degrees of freedom* is much smaller than the number of *numerical degrees of freedom* (i.e. grid points or cells) required by standard CFD solvers. This is because of the redundancies associated with the underlying physical laws (e.g., mass and momentum conservation in the Navier-Stokes equations). A large class of ROMs first identify the physically relevant degrees of freedom by using POD and extracting the most energetic POD modes, and then project the governing equations (or the CFD numerical scheme) onto these modes. Most ROMs based on this idea are designed to simulate attractors (not transients) and could be called *pre-processed ROMs* because POD modes are calculated from a CFD run in a pre-process step. In principle, the CFD run should cover the relevant dynamics to be approximated by the ROM; this strategy can be improved using additional ingredients for fluid flows [6].

A different perspective has been tested for simpler equations such as the complex Ginzburg-Landau equation [7], a paradigm of a pattern forming system that exhibits quite complex dynamic behavior. The method consists in an adaptive strategy in which POD modes are calculated on demand, as the simulation proceeds. This strategy has proven to be very robust in both calculating particular solutions and constructing bifurcation diagrams [8], that require a large number of particular time-dependent runs. The method uses both a CFD solver and a reduced order model in interspersed time intervals and can thus be called a *POD on the fly* method. As it can be seen in Figure 1, the CFD solver is used to compute appropriate snapshots that in turn are used to first calculate (at the outset of the temporal run) and later update (when needed) the POD manifold. The ROM, which as mentioned is obtained by projecting the governing equations onto the POD modes, is much less computationally expensive than the CFD solver and is intended to be used for the majority of the time. In other words, the adaptive strategy is implemented in a numerical scheme capable of switching seamlessly between the full integration of the governing equations and an adaptive ROM that guarantees capturing the present simulation dynamics.

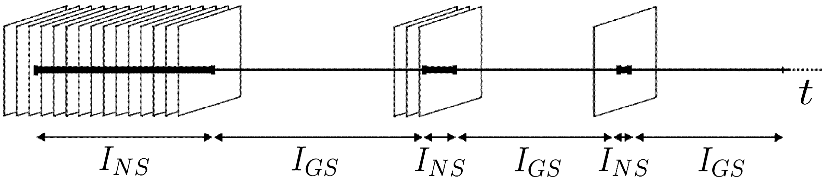

Figure 1: Sketch showing the ROM and full numerical solver time intervals

Note that such an adaptive strategy requires two main ingredients:

- The requirement for updating the POD manifold must be detected using only information from the ROM. Appropriate a priori error/residual estimates can be used to this end.
- A convenient updating strategy for the POD modes at each updating event.

Additionally, the computational cost required to project the governing equations (especially, nonlinear terms) can be further reduced if only a given set of cells, where the dominant dynamics of the problem reside, are utilized. Such a reduction of the computational cost has been used in the reduced modeling of several systems, arising in non-linear elasticity [9], steady aerodynamic forces [10, 11] as well as unsteady incompressible flows [12]. In particular, for the complex

Ginzburg-Landau equation, these ideas permit obtaining accelerating CPU factors in the range 10-15 and 300-350 for simulations in one and two space dimensions, respectively.

The extension of the POD on the fly method to the inviscid unsteady aerodynamic flows, including transonic flows, that are relevant in aeroelasticity has been presented in [13].

This paper extends the application to include the deformation of the numerical mesh, as well as the coupling with a structural model, in order to perform complete aeroelastic simulations in the time domain with increased computational efficiency.

2 FLUID STRUCTURE INTERACTION

In a *coupled fluid-structure interaction* problem, the position and speed of each point and cell in the computational domain Ω depend on the body motion. At the same time, the body motion depends on the internal elastic, viscous, inertial and control forces, as well as on the forces acting on the boundary shared with the fluid.

The equations governing the problem can be written as

$$\frac{\partial(\mathcal{V}_i(\tilde{x},t)\mathcal{Q}_i)}{\partial t} + \mathcal{R}_i(\mathcal{Q},\tilde{x}) = 0, \tag{1}$$

$$\mathcal{M}\frac{\partial^2\tilde{x}}{\partial t^2} + \mathcal{B}\frac{\partial\tilde{x}}{\partial t} + \mathcal{K}\tilde{x} + \phi(\tilde{x}) = \mathcal{F}(\mathcal{Q}), \tag{2}$$

where Ω is discretized in $i = 1,..,M$ cells of volume $\mathcal{V}_i$ and $\mathcal{Q}_i = \{\rho, u, v, w, p\}_i$ is the state vector joining the density, speeds and pressure of the fluid, respectively. A fluid state equation is additionally required to close the problem.

The coordinates corresponding to the body motion are represented by $\tilde{x}$, and $\mathcal{M}, \mathcal{B}$ and $\mathcal{K}$ represent the mass, damping and stiffness matrices of the body, respectively. Finally, $\phi\left(\tilde{x}\right)$ is a non-linear stiffness function, e.g., a free-play in a hinge, and $\mathcal{F}(\mathcal{Q})$ accounts for the generalized fluid forces on the body.

This problem is traditionally solved in a loosely coupled manner, since the characteristic time of the body motion is usually orders of magnitude higher than the aerodynamic counterpart one. In this way, the flow state is solved first, using the body state from the previous time step. The fluid forces acting on the body are computed and subsequently the body is evolved in time.

The fluid governing equations are evolved in time using an implicit, first order accurate, Euler algorithm, to which a pseudo-time sub-iteration step [14] has been added as

$$\begin{aligned}\left(\frac{\mathcal{V}_i^{n+1}\mathcal{Q}_i^{m+1,n+1} - \mathcal{V}_i^{n+1}\mathcal{Q}_i^{m,n+1}}{\Delta\tau} + \frac{\mathcal{V}_i^{n+1}\mathcal{Q}_i^{m+1,n+1} - \mathcal{V}_i^{n}\mathcal{Q}_i^{n}}{\Delta t}\right) + \mathcal{R}_i^m \\ + \frac{\partial\mathcal{R}_i^n}{\partial\mathcal{Q}}\left(\mathcal{Q}^{m+1,n+1} - \mathcal{Q}^{m,n+1}\right) = 0,\end{aligned} \tag{3}$$

where $\Delta\tau$ denotes the pseudo time-step, which is varied for each cell based on a constant CFL number $(u\Delta\tau/\Delta x)$ [15], and $\mathcal{Q}_i^{m,n+1}$ identifies the $m{-}th$ approximation to $\mathcal{Q}_i^{n+1}$. The Jacobian might be approximated by a lower order discretization, or even a diagonal approximation.

The solution is iterated in m until enough convergence has been achieved, as

$$\left(\frac{\mathcal{V}_i^{n+1}}{\Delta\tau}+\frac{\mathcal{V}_i^{n+1}}{\Delta t}+\frac{\partial\mathcal{R}_i^n}{\partial\mathcal{Q}_i}\right)\left(\mathcal{Q}_i^{m+1,n+1}-\mathcal{Q}_i^{m,n+1}\right)= \\ -\left(\mathcal{R}_i^m+\frac{\mathcal{V}_i^{n+1}\mathcal{Q}_i^{m,n+1}-\mathcal{V}_i^n\mathcal{Q}_i^n}{\Delta t}\right). \tag{4}$$

After reaching convergence, for example based on the relative change of the state vector and the reduction of the residuals a given order of magnitude, the forces acting on the body, $\mathcal{F}(\mathcal{Q})$, are calculated and the body state is then evolved in time. The evolution of the body changes the definition of the computational domain mesh, which might imply the update of the cell volumes, $\mathcal{V}_i$, that is used to further carry on the fluid integration in the next time step.

2.1 Mesh Deformation

The numerical fluid volume mesh needs to move in such a way that it is conformal to the body and the quality of the mesh is degraded as little as possible, since it could introduce errors in the solution

The explicit interpolation method of Luke and Blades [16] has been chosen in the present work due to its robustness, improved quality of the deformed mesh, efficiency and reduced numerical cost. In this method, the deformation of the volume mesh is given as a weighted sum of the rigid body displacements corresponding to each body-fluid interface surface point.

The displacement field $\overrightarrow{S}_i(\overrightarrow{r})$ of a node i on the deforming surface is given by

$$\overrightarrow{S}_i(\overrightarrow{r})=M_i+\overrightarrow{b}_i-\overrightarrow{r}, \tag{5}$$

where M_i is a rotation matrix, $\overrightarrow{b}_i$ is a displacement vector associated with the i^{th} node and $\overrightarrow{r}$ is a coordinate vector in the original mesh. The displacement field in the volume mesh is then described through a weighted average of all boundary node displacement fields:

$$\overrightarrow{S}(\overrightarrow{r})=\frac{\sum \mathrm{w}_i(\overrightarrow{r})\overrightarrow{S}_i(\overrightarrow{r})}{\sum \mathrm{w}_i(\overrightarrow{r})} \tag{6}$$

The interpolation weight function is chosen such that the weighting is a function of the reciprocal of distance. Near boundary deformations are preserved, while still providing a smooth transition, by using a two-exponent form. Finally, the area of the boundary face is used so that mesh refinement of a region (close to expected shocks for example) does not increase its influence in the interpolation method. The resulting weight function is given as

$$\mathrm{w}_i(\overrightarrow{r})=|\overline{S}_i|\left[\left(\frac{L_{def}}{|\overrightarrow{r}-\overrightarrow{r_i}|}\right)^a+\left(\frac{\alpha L_{def}}{|\overrightarrow{r}-\overrightarrow{r_i}|}\right)^b\right], \tag{7}$$

where a and b are user-defined tunable exponents, typically set to 3 and 5 respectively, L_{def} is an estimated length of the deformation region and α is an estimated size of the near body influence region, typically set to 0.2 to preserve viscous mesh quality in the boundary layer region.

3 POD BASED REDUCED ORDER MODEL

3.1 POD decomposition

Once the complete problem for the governing equations has been integrated for an adequate timespan, the flow state variables for each time step can be concatenated into a snapshot matrix, such as

$$S = \begin{bmatrix} \mathcal{Q}_1^1 & \mathcal{Q}_1^2 & \cdots & \cdots & \mathcal{Q}_1^T \\ \mathcal{Q}_2^1 & \mathcal{Q}_2^2 & \cdots & \cdots & \mathcal{Q}_2^T \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{Q}_M^1 & \mathcal{Q}_M^2 & \cdots & \cdots & \mathcal{Q}_M^T \end{bmatrix}. \tag{8}$$

The POD modes are calculated from this snapshot matrix. Traditionally, POD modes have been calculated by obtaining the covariance matrix of the snapshot matrix as $R_{ij} = \langle S_i, S_j \rangle$ for a given inner product $\langle \cdot, \cdot \rangle$. It is customary expressing the eigenvalue decomposition of R as

$$RA = AD^2, \tag{9}$$

where the columns of A are the orthonormal eigenvectors of R and D^2 is the diagonal matrix of the eigenvalues of R, whose square roots are the singular values. Then, the POD modes are the columns of the following matrix

$$\Phi = SAD^{-1}, \tag{10}$$

so that they are orthonormal with the considered inner product.

An additional important property of the POD modes is that, if the snapshot matrix is approximated as S^N making use of the first N POD modes, the relative root mean square error (RRMSE) can be estimated as

$$RRMSE_N^T = \sqrt{\frac{\sum\limits_{i=N+1}^{T} (D_{ii})^2}{\sum\limits_{i=1}^{T} (D_{ii})^2}}. \tag{11}$$

However, due to the fact that the covariance matrix requires the multiplication by the inner product of the snapshots, the POD modes associated with singular values such that $(D_{jj})/(D_{11}) < 10^{-8}$ are not valid [7] with the standard double precision binary floating point format.

For the particular, and typical, case in which the inner product can be represented as

$$\langle S_i, S_j \rangle = (S_i)^T \mathcal{G} S_j, \tag{12}$$

where $\mathcal{G}$ is a diagonal matrix, the accuracy of the POD modes is increased by avoiding the creation of the covariance matrix. In a first step, SVD decomposition is applied to

$$\tilde{S} = \sqrt{\mathcal{G}} S, \tag{13}$$

so that

$$\tilde{S} = U \Sigma V^T, \tag{14}$$

and therefore

$$S = \sqrt{\mathcal{G}^{-1}} U \Sigma V^T. \tag{15}$$

Substituting these into the definition of the covariance matrix, and taking into account that D and $\mathcal{G}$ are diagonal matrices, it follows that

$$R = S^T \mathcal{G} S = V \Sigma U^T \sqrt{\mathcal{G}^{-1}} \mathcal{G} \sqrt{\mathcal{G}^{-1}} U \Sigma V^T = V \Sigma^2 V^T. \tag{16}$$

Finally, comparing (9) and (16) it is trivial to show that

$$\Phi = S V \Sigma^{-1} = \sqrt{\mathcal{G}^{-1}} U. \tag{17}$$

3.2 Flow field as combination of POD modes

For a given set of N selected POD modes, the flow field can be expressed as a combination of the modes, where a reference value $\mathcal{Q}_o$ might potentially be used as

$$\mathcal{Q} = \mathcal{Q}_o + \Phi \xi, \tag{18}$$

so that with the substitution into (3), it follows

$$\sum_{j=1}^{N} \Phi_{ij} \left(\frac{\mathcal{V}_i^{n+1} \xi_j^{n+1} - \mathcal{V}_i^n \xi_j^n}{\Delta t} \right) + \sum_{k=1}^{M} \frac{\partial \mathcal{R}_i^n}{\partial \mathcal{Q}^k} \sum_{j=1}^{N} \Phi_{kj} \left(\xi_j^{n+1} - \xi_j^n \right) + \mathcal{R}_i^n = 0, \tag{19}$$

which could be used to evolve the fluid equations.

Unfortunately, the number of cells in which the dominion is divided is usually orders of magnitude higher than the number of modes retained, making computationally expensive the minimization of the residual. Further reduction needs to be applied to the system, which can be done in any of the following ways:

- Minimize the least squares residual on a subset of M selected cells

$$min \left(\sum_{i=1}^{M} \left[\sum_{j=1}^{N} \Phi_{ij} \left(\frac{\mathcal{V}_i^{n+1} \xi_j^{n+1} - \mathcal{V}_i^n \xi_j^n}{\Delta t} \right) + \mathcal{R}_i(\mathcal{Q}_o + \Phi \xi^{n+1}) \right]^2 \right) \tag{20}$$

- Project the equation over the POD modes, using a defined inner product. This is known as Galerkin projection:

$$\left\langle \Phi, \sum_{j=1}^{N} \Phi_{ij} \left(\frac{\mathcal{V}_i^{n+1} \xi_j^{n+1} - \mathcal{V}_i^n \xi_j^n}{\Delta t} \right) + \mathcal{R}_i(\mathcal{Q}_o + \Phi \xi^{n+1}) \right\rangle = 0 \tag{21}$$

In this case, the definition of the inner product is essential, since it plays a key role in the characteristics of the equation above

4 ADAPTIVE POD ON THE FLY

4.1 POD based ROM drawbacks

Despite the use of the POD projection based ROMs, the unsteady simulations would still be prohibitively expensive for their use on an industrial environment. In other words, the overhead of performing the POD decomposition over a several millions (a typical number of grid points) times several thousands (a possible number of snapshots) snapshots matrix has to be added to the cost of solving the fluid equations to calculate the snapshots.

Besides, the snapshots to create the ROM have been traditionally obtained from complete simulations time history. This allows to simulate problems with similar dynamics to the previous solutions but, since the full equations are not completely integrated, it is not able to capture behaviors of the system not previously encountered

On another note, POD based ROMs are known to traditionally suffer from stability problems, mainly due to the fact that the numerical scheme used to obtain the solution can be stable, but the projection on a truncated basis that does not cover the whole solution subspace traditionally is not. In this way, some modes that initially had very low energy become dominant in the solution and the ROM validity is jeopardized.

4.2 POD on the fly

The new methodology, introduced here in the context of unsteady aerodynamics and computational aeroelasticity, known as *POD on the fly* and previously presented in [7, 8, 10–12], overcomes these drawbacks.

A flowchart summarizing the particularities of the method can be seen in Figure 2. Initially, the integration of the governing equations starts using the full numerical solver. Then, after a prescribed amount of time steps, for a time T_o^{CFD}, have been completed, a POD basis is obtained and truncated to a number N of modes in such a way that

$$RRMSE_N^T \leq \varepsilon_1, \tag{22}$$

where $RRMSE_N^T$ is defined in (11), and ε_1 is a tunable parameter.

At this point, the integration is switched to the ROM solver, and the system evolved in time, until the ROM loses its validity due to changes in system dynamics or truncation instabilities. It is therefore paramount to introduce in the time evolution the ability to detect this phenomenon. This is accomplished in this work by monitoring the residual reduction the ROM is able to achieve. In this way, the ROM integration will finish if

$$E_{res} = \frac{\left\| \sum_{j=1}^{N} \Phi_{ij} \left(\frac{\mathcal{V}_i^{n+1} \xi_j^{n+1} - \mathcal{V}_i^n \xi_j^n}{\Delta t} \right) + \mathcal{R}_i(\mathcal{Q}_o + \Phi \xi^{n+1}) \right\|}{\left\| \mathcal{R}_i(\mathcal{Q}_o + \Phi \xi^n) \right\|} > \frac{1}{K}, \tag{23}$$

where K is a tunable parameter and $\| \cdot \|$ is associated with a previously defined inner product.

After switching back to the numerical solver, the system is evolved for a time ΔT^{CFD}. It must be noted that the selection of ΔT^{CFD} needs to consider the balance between acquiring the new system dynamics and the efficiency of the overall integration scheme.

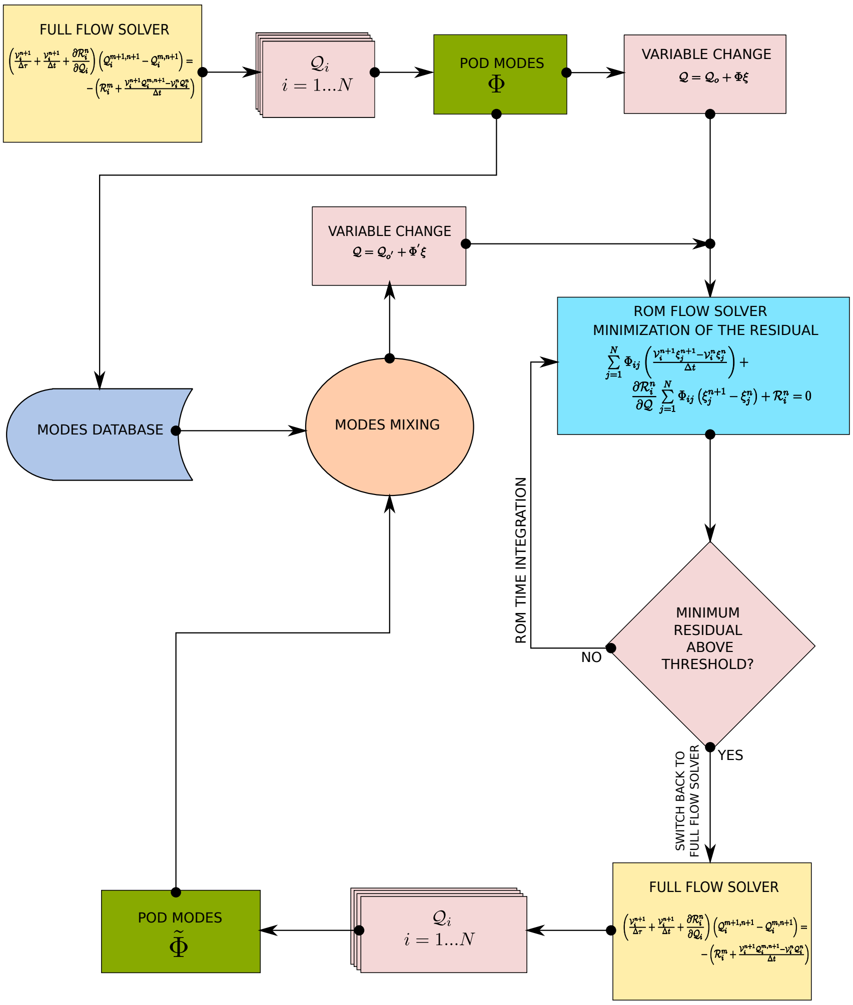
FULL FLOW SOLVER
$\left(\frac{\mathcal{V}_i^{n+1}}{\Delta\tau}+\frac{\mathcal{V}_i^{n+1}}{\Delta t}+\frac{\partial\mathcal{R}_i^n}{\partial\mathcal{Q}_i}\right)\left(\mathcal{Q}_i^{m+1,n+1}-\mathcal{Q}_i^{m,n+1}\right)=-\left(\mathcal{R}_i^m+\frac{\mathcal{V}_i^{n+1}\mathcal{Q}_i^{m,n+1}-\mathcal{V}_i^n\mathcal{Q}_i^n}{\Delta t}\right)$
$\mathcal{Q}_i$
$i=1...N$
POD MODES
Φ
VARIABLE CHANGE
$\mathcal{Q}=\mathcal{Q}_o+\Phi\xi$
VARIABLE CHANGE
$\mathcal{Q}=\mathcal{Q}_{o'}+\Phi'\xi$
ROM FLOW SOLVER
MINIMIZATION OF THE RESIDUAL
$\sum_{j=1}^{N}\Phi_{ij}\left(\frac{\mathcal{V}_i^{n+1}\xi_j^{n+1}-\mathcal{V}_i^n\xi_j^n}{\Delta t}\right)+\frac{\partial\mathcal{R}_i^n}{\partial\mathcal{Q}}\sum_{j=1}^{N}\Phi_{ij}\left(\xi_j^{n+1}-\xi_j^n\right)+\mathcal{R}_i^n=0$
MODES DATABASE
MODES MIXING
ROM TIME INTEGRATION
MINIMUM RESIDUAL ABOVE THRESHOLD?
NO
YES
SWITCH BACK TO FULL FLOW SOLVER
POD MODES
$\tilde{\Phi}$
$\mathcal{Q}_i$
$i=1...N$
FULL FLOW SOLVER
$\left(\frac{\mathcal{V}_i^{n+1}}{\Delta\tau}+\frac{\mathcal{V}_i^{n+1}}{\Delta t}+\frac{\partial\mathcal{R}_i^n}{\partial\mathcal{Q}_i}\right)\left(\mathcal{Q}_i^{m+1,n+1}-\mathcal{Q}_i^{m,n+1}\right)=-\left(\mathcal{R}_i^m+\frac{\mathcal{V}_i^{n+1}\mathcal{Q}_i^{m,n+1}-\mathcal{V}_i^n\mathcal{Q}_i^n}{\Delta t}\right)$

Figure 2: Adaptive ROM Flowchart

At the end of the period ΔT^{CFD}, the new system dynamics have been acquired, and a new POD basis (Φ) has been calculated from the corresponding matrix of snapshots. Since some of the previous system dynamics are most likely part of the present system evolution, both Φ and $\hat{\Phi}$ (the previous set of modes) need to be combined to create the new updated ROM. This is done by applying POD to the set of vectors

$$\hat{\nu}_1\hat{\Phi}_1, \cdots, \hat{\nu}_{N_{old}}\hat{\Phi}_{N_{old}}, \nu_1\Phi_1 \cdots, \nu_{N_{new}}\Phi_{N_{new}}, \tag{24}$$

where the weights $\hat{\nu}_j$ and ν_j are defined as

$$\hat{\nu}_j = min\left\{\frac{\hat{D}_{jj}}{\sqrt{\sum_{i=1}^{N_{old}}(D_{ii})^2}}, \frac{\xi^j}{\sqrt{\sum_{i=1}^{N_{old}}(\xi^j)^2}}\right\}, \nu_j = \frac{D_{jj}}{\sqrt{\sum_{i=1}^{N_{new}}(D_{ii})^2}}. \tag{25}$$

In this way, those modes with higher relative "energy" (based on the inner product), and with higher relative contribution to the flow field, are retained in the newly created ROM, and the process is ready to start over again.

4.3 Selection based inner product

The main drawback of the traditional POD based ROMs is the computational overhead associated to the extraction of the POD modes. As previously highlighted, this cost is intrinsically associated to the definition of the inner product.

One way to overcome this drawback is [10, 11] to define the inner product only in a subset of the cells that form the computational domain in which the fluid is divided. By doing so, the evolution of the fluid variables for the non-selected cells is made slave to the ones of the selected domain.

Recalling the definition of inner product in matrix form

$$\langle S_i, S_j \rangle = (S_i)^T \mathcal{G} S_j \tag{26}$$

and rearranging the vectors such that the M selected subset cells are at the top, the $\mathcal{G}$ matrix is created such that

$$\mathcal{G} = diag\left(\mathcal{G}_1, \ldots, \mathcal{G}_M, 0, \ldots, 0\right). \tag{27}$$

Since all the snapshots are used for the creation of the POD modes, a full reconstruction of the flow field is possible, while reducing significantly the cost associated to the extraction of the POD modes. Besides, the spatial derivatives corresponding to each of the POD modes, in the selected and slave cells, can be calculated and stored in order to further reduce the cost. Additionally, the residual equations for the rows corresponding to the non selected cells do not play any role in the minimization, so the cost can be further reduced by avoiding its calculation.

Due to the fact that the rows of the residual corresponding to the time evolution of the selected cells are calculated, but the residual depends on values from its neighbor cells, special care has to be put in choosing the cell selection, so that it allows for adequate imposition of the boundary conditions. Also, due to the fact that the behavior of the non-selected cells is subordinated to the evolution of the selected cells, it is important to include in this selection those areas were significant and unique dynamics exist. Lastly, it is typical of CFD solvers to accumulate errors in specific zones of the computational domain, like for example the trailing edges of lifting surfaces. By not selecting cells in this regions, the robustness and significance of the extracted modes is increased.

4.4 Residual minimization

For the results in this paper, the ROM residuals are minimized in a least square sense (see (20)) making use of a Gauss-Newton algorithm [17, pp. 67-68], such that if $\mathfrak{J}$ is the Jacobian of the ROM residual $\mathfrak{R}$ with respect to the POD modes coordinates ξ, defined as

$$\mathfrak{R}_i = \sum_{j=1}^{N} \Phi_{ij} \left(\frac{\mathcal{V}_i^{n+1} \xi_j^{n+1} - \mathcal{V}_i^{n} \xi_j^{n}}{\Delta t} \right) + \sum_k \frac{\partial \mathcal{R}_i^n}{\partial \mathcal{Q}^k} \sum_{j=1}^{N} \Phi_{kj} \left(\xi_j^{n+1} - \xi_j^n \right) + \mathcal{R}_i^n = 0, \tag{28}$$

$$\mathfrak{J}_{ij} = \Phi_{ij} \frac{\mathcal{V}^{i,n+1}}{\Delta t} + \sum_k \frac{\partial \mathcal{R}^{i,n}}{\partial \mathcal{Q}^k} \Phi_{kj}, \tag{29}$$

the incremental step to the values of ξ that would reduce the residual norm is given as

$$\Delta \xi^{n+1} = -\left(\mathfrak{J}^T \mathfrak{J} \right)^{-1} \left(\mathfrak{J}^T \mathfrak{R} \right). \tag{30}$$

The Gauss-Newton algorithm is applied until convergence in the forces applied over the body and in the norm of the residuals has been achieved, or until a maximum number of iterations has been reached.

5 PRACTICAL APPLICATION

Let us consider the aeroelastic response of a three degrees of freedom (DoF) airfoil. For this, the airfoil equations of motions are loosely coupled to an in-house inviscid CFD solver, developed for the purpose of this study and previously presented in [13]. The coupled system is excited at speeds above and below the critical flutter speed both in the subsonic (M=0.2) and high transonic (M=0.8) range.

5.1 Structural model

The airfoil, sketched in Figure 3, can heave and rotate around its center of rotation, located at a distance x_{rot} from the nose. A control surface, attached to it, can rotate around a hinge placed at a distance x_c from the center of rotation. Its motion can subsequently be described with three generalized coordinates, namely, the vertical deflection h, the rotation around the center of rotation θ and the control surface deflection angle β.

The system mass properties can be represented by the box mass (M_B), the position of the box center of gravity (CG) relative to the rotation center (x_B), the mass inertia of the box around its CG (I_B), the control surface mass (M_{cs}), the position of the control surface CG relative to the hinge (x_{cs}), and the mass inertia of the control surface around its CG (I_{cs}).

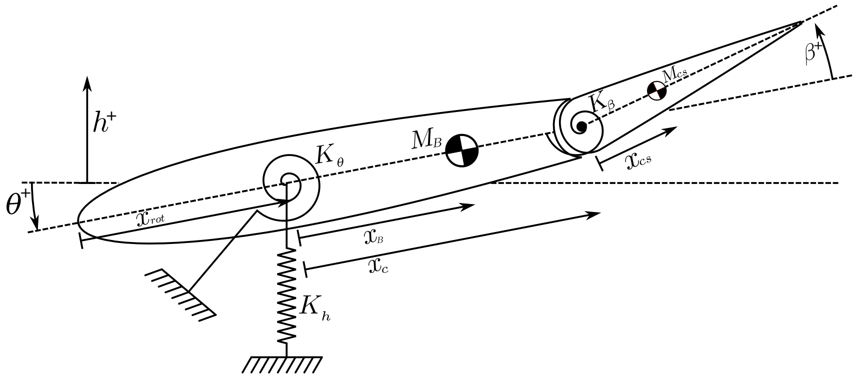

Figure 3: Airfoil description

The governing equations of the airfoil motion are given by:

$$\ddot{h} + x_\theta\ddot{\theta} + \frac{M_{cs}}{M}x_{cs}\ddot{\beta} + g_h\omega_h^2\dot{h} + \omega_h^2 h = \frac{\mathcal{F}_h}{M} = -\frac{1}{M}\int_{\partial\Omega} pS_{fz}ds, \quad (31)$$

$$x_\theta\ddot{h} + \frac{I_\theta}{M}\ddot{\theta} + \left[\frac{M_{cs}}{M}x_c x_{cs} + \frac{I_\beta}{M}\right]\ddot{\beta} + \frac{I_\theta}{M}g_\theta\omega_\theta^2\dot{\theta} + \frac{I_\theta}{M}\omega_\theta^2\theta = \frac{\mathcal{F}_\theta}{M} = -\frac{1}{M}\int_{\partial\Omega} pxS_{fz}ds, \quad (32)$$

$$\begin{aligned}\frac{M_{cs}}{M}x_{cs}\ddot{h} + \left[\frac{M_{cs}}{M}x_c x_{cs} + \frac{I_\beta}{M}\right]\ddot{\theta} + \frac{I_\beta}{M}\ddot{\beta} + \frac{I_\beta}{M}g_\beta\omega_\beta^2\dot{\beta} + \frac{I_\beta}{M}\omega_\beta^2\beta = \frac{\mathcal{F}_\beta}{M} = \\ = -\frac{1}{M}\int_{\partial\Omega_{cs}} p(x - x_c)S_{fz}ds.\end{aligned} \quad (33)$$

5.2 Computational fluid domain and mesh

The aerodynamic shape has been chosen as a 4 digit symmetric, uncambered, NACA airfoil with a 10% thickness to chord length ratio, known as NACA-0010 [18]. This type of airfoil has been commonly used in computational and experimental aeroelastic problems [19–21].

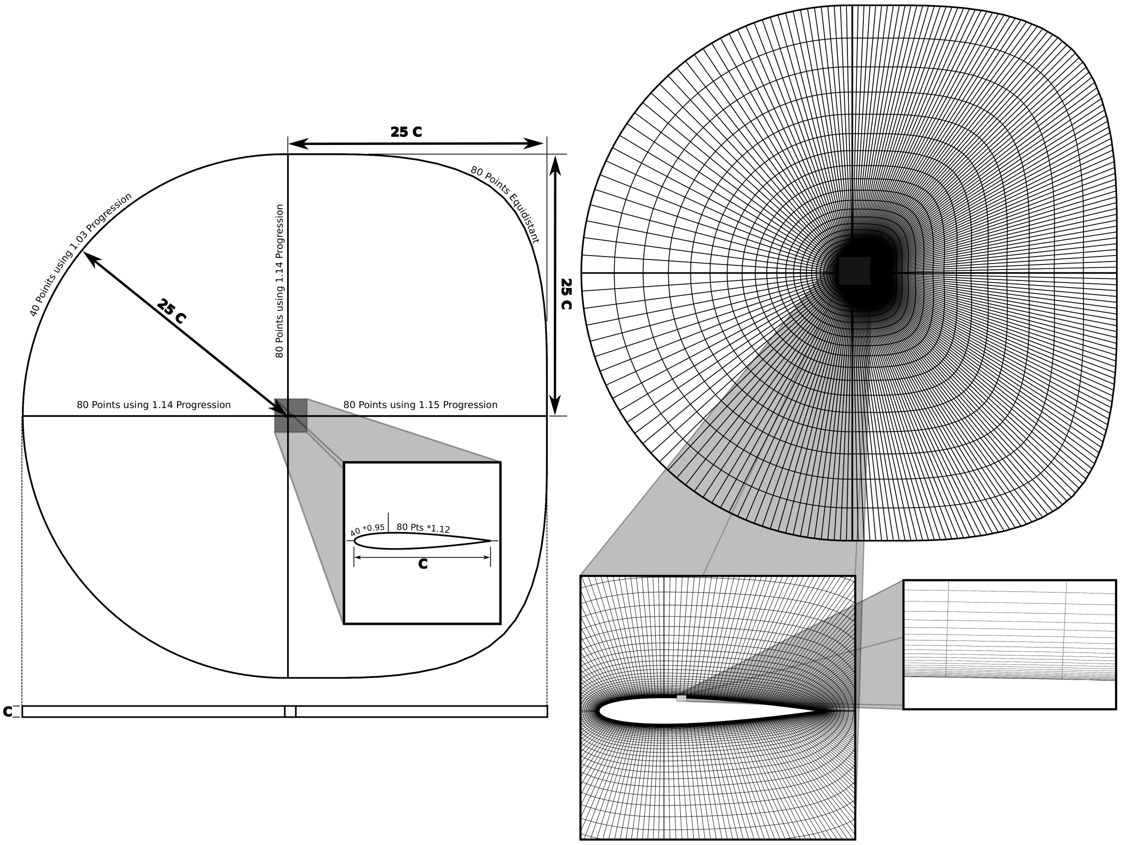
25 C
80 Points Equidistant
40 Points using 1.03 Progression
25 C
80 Points using 1.14 Progression
25 C
80 Points using 1.14 Progression
80 Points using 1.15 Progression
40 *0.95
80 Pts *1.12
C
C

Figure 4: Computational domain (left) and numerical mesh (right)

The computational mesh, see Figure 4(right), has been created in the domain presented in Figure 4(left). It consists of 18802 cells and has been generated using Gmsh, a three-dimensional mesh generator with built-in pre- and post-processing utilities [22].

5.3 Sub-mesh containing the master cells

A substantial and critical part of the novel adaptive ROM developed here relies on the use of a limited number of *master cells* that are used to apply POD and project the governing equations.

We insist that, in some sense, the remaining cells just act as ‘slaves’ of the master cells through the POD modes. Thus, the master cells must be distributed in the computational domain such that they allow the enforcement of the boundary conditions and the transmission of information between the different regions. They should also contain the interesting dynamics of the problem. Regions with known numerical noise, like the trailing edge of lifting surfaces, should be avoided.

Several methods have been proposed in literature, such as *missing point estimation* [23], *hyper-reduction* [9] and LUPOD [24]. These methods optimize the sub-mesh selection according to the expected dynamics and/or the nature of the operators acting on the POD modes. However, since the goal is to adapt to unknown possible dynamics, a more general procedure is used here.

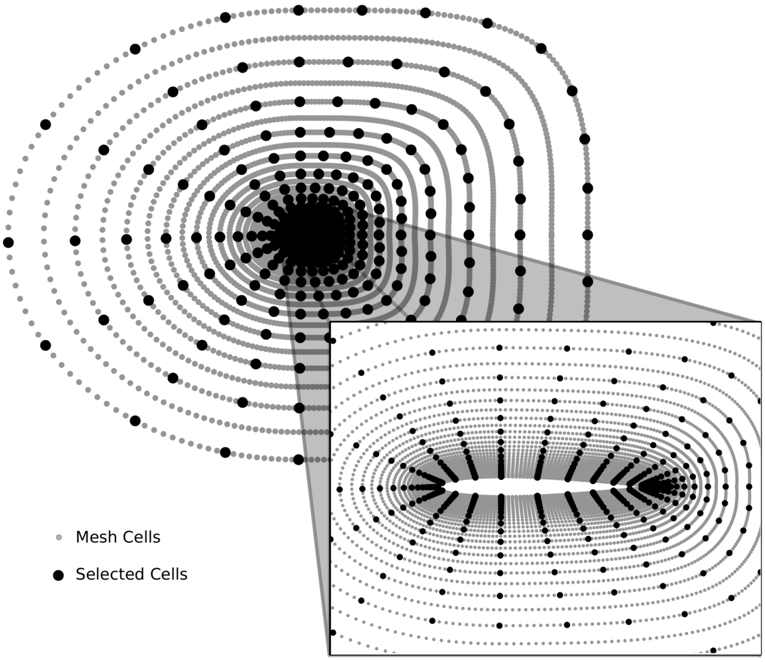

Figure 5: Sub-mesh containing the master cells

A series of (radial) lines are traced at regular intervals from the airfoil wall to far-field faces and 960 cells (see Figure 5) in total are selected along these lines. No additional points are included to capture the shock that is present in the high transonic flow considered below. This is because the position of the shock is unknown, and the sub-mesh is selected beforehand here. Of course, an adaptive selection would be more convenient, but this is well ahead of the scope of this paper.

5.4 Results for the subsonic flow at M=0.2

As a first case to test the adaptive POD on the fly methodology for aeroelastic simulations, we consider the impulse response of the structure in the low subsonic regime, M=0.2.

The POD modes have been calculated using the steady state as reference, and the ROM residuals have been minimized using a Gauss-Newton algorithm. The tunable parameters of the problem appearing in §4.2 are selected for the subsonic case considered as

$$T_0^{\mathrm{CFD}} = 0.05 \text{ s}, \quad \Delta T^{\mathrm{CFD}} = 0.05 \text{ s}, \quad \epsilon_1 = 10^{-10}, \quad K = 100.$$

The aeroelastic simulations are run at a speed below (Figure 6) and above (Figure 7) the critical flutter speed. The unshaded areas in both figures represent time spans at which the ROM has lost its validity, under the criteria presented above, and the full numerical solver has been run for a time ΔT^{CFD} to update the modal basis. As it can be seen, the agreement is excellent for the initial phases of the simulation but the control surface rotation displacement deviates slowly from the numerical solver results as the simulation evolves, indicating that the update criteria would ideally need to be made more stringent. Additional runs show that just by doubling the value of K, the agreement is excellent for the whole simulation (see Figure 8).

One additional interesting remark is the fact that for both simulation points very few updates are deemed necessary once the ROM has acquired a sufficient number of modes.

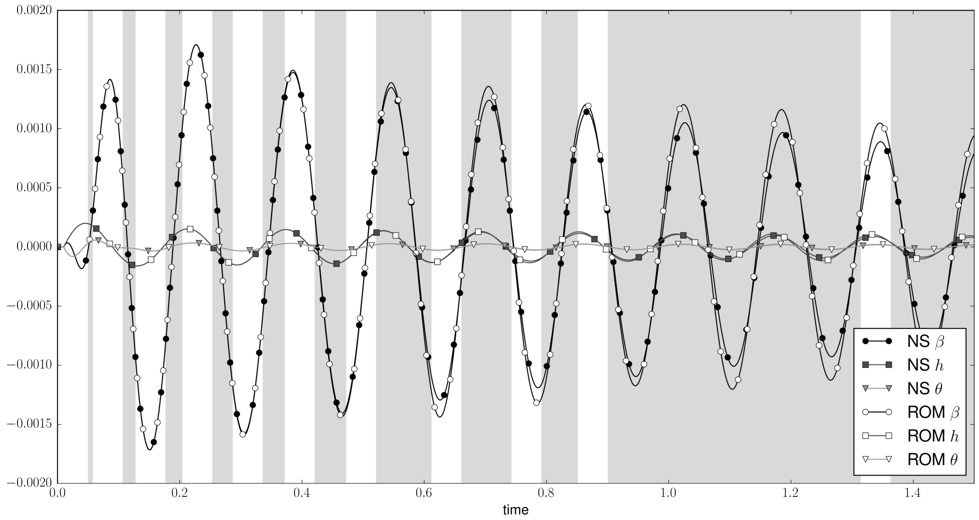

Figure 6: POD on the fly vs. the CFD solver. M=0.2 at a speed below flutter point

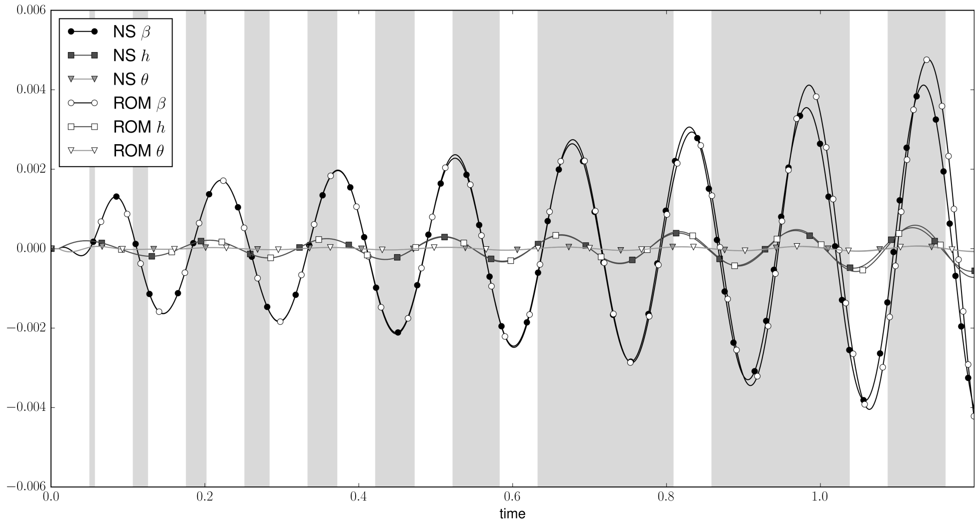

Figure 7: POD on the fly vs. the CFD solver. M=0.2 at a speed above flutter point

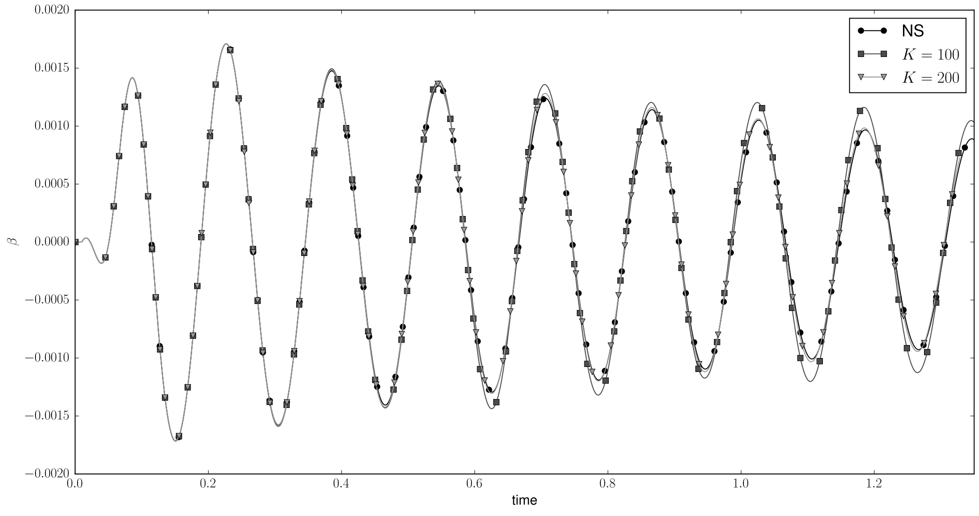

Figure 8: K effect on control surface rotation, for M=0.2 at a speed below flutter point

5.5 Results for the transonic flow at M=0.8

Let us now increase the Mach number to M=0.8. At this conditions, the NACA0010 airfoil exhibits strong shocks in the pressure and suction sides of the airfoil, which need to be captured by the ROM (see for example Figure 11)

The POD modes have been calculated using the steady state as reference and the ROM residuals have been minimized using a Gauss-Newton algorithm. In order to capture the shock motion, the ROM updates have been made more frequent, by reducing the value of ΔT^{CFD}. The tunable parameters of the problem appearing in §4.2 are selected for the transonic case considered here as

$$T_0^{\text{CFD}} = 0.05 \text{ s}, \quad \Delta T^{\text{CFD}} = 0.005 \text{ s}, \quad \epsilon_1 = 10^{-10}, \quad K = 100.$$

The aeroelastic simulations are run at a speed below (Figure 9) and above (Figure 10) the critical flutter speed. Overall, the agreement is excellent, with both simulations practically indiscernible from each other. This comes, however, at the cost of more frequent ROM updates needed to capture the significant shock movement.

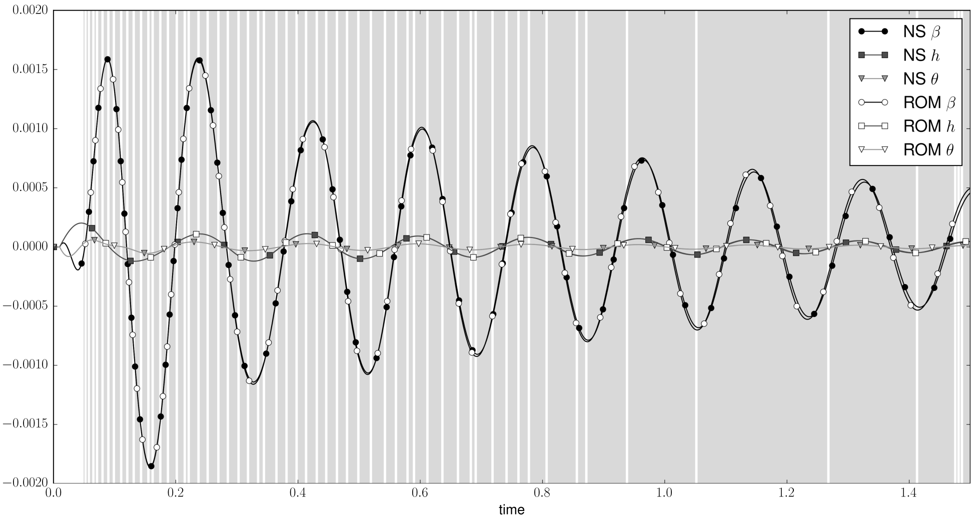

Figure 9: POD on the fly vs. the CFD solver. M=0.8 at a speed below flutter point

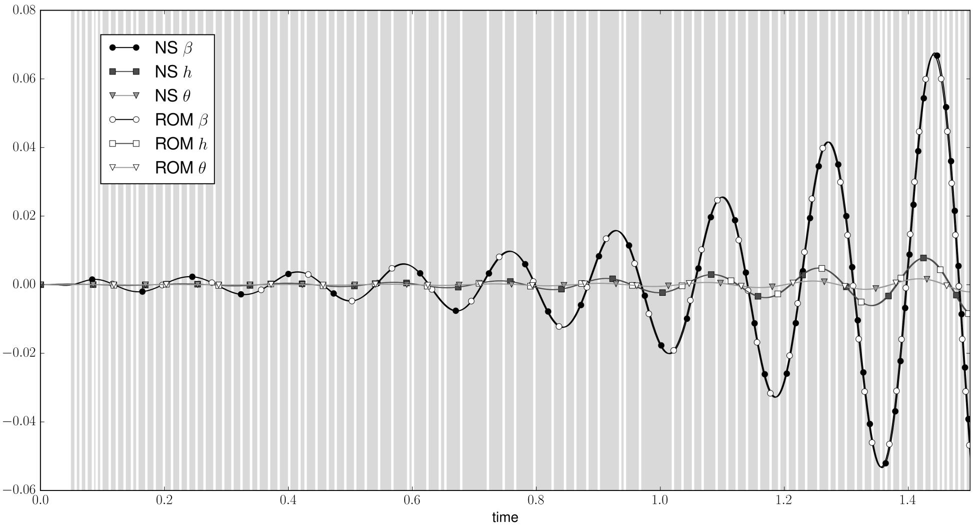

Figure 10: POD on the fly vs. the CFD solver. M=0.8 at a speed above flutter point

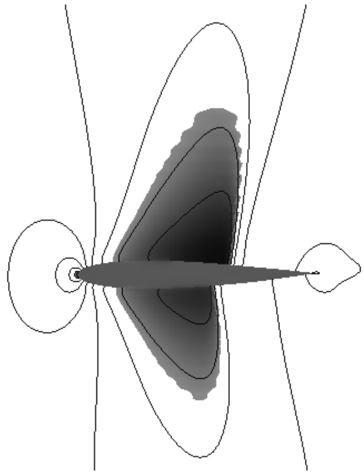

Figure 11: Iso Mach contours during simulation at M=0.8

5.6 ROM efficiency and number of modes

The ratio between the time the POD on the fly ROM is run and the time the full solver is run, in order to update the modal basis, is presented here as a first measure of the ROM efficiency.

For the subsonic case (Figure 12), this ratio increases as the simulation time (and the number of restarts) increases, up to the point that almost no more restarts are needed. In the transonic case (Figure 13) we have two distinct effects. For the stable and unstable cases, the ratio starts growing (up to very high values for the stable case) but then the ratio decreases mainly due to the fact that for the unstable case the amplitude keeps growing, and the shock movement increases.

The previous indicator neglects the fact that the computational cost of the ROM is not negligible. Instead, it increases as the number of retained modes increases. Thus, it is convenient to also take into account the required number of POD modes that are retained. As shown, the number of modes increases with the simulation time, converging to a given number for each of the considered conditions. For the unstable transonic case, since the distance the shock travels increases with each oscillation, the number of modes needed keeps increasing.

6 CONCLUSIONS AND FUTURE WORK

The extension of the novel POD on the fly method to the simulation of unsteady aerodynamics, and its application to aeroelastic simulations, has been successfully demonstrated in the cases where fluid viscosity is neglected. The lack of viscosity in the simulation has posed some additional challenges in the extension of the method, which have been resolved to obtain a robust method with high application potential. Several new ingredients have been incorporated additionally to the method, such as the ability to deal with deforming meshes and shock motion.

In order to demonstrate the application of the method, a two-dimensional airfoil has been considered. Since the number of modes necessary to represent the solution do not grow as quickly as the number of numerical degrees of freedom when increasing the number of spatial dimensions, the three-dimensional lifting surface case is expected to have higher gains.

The method has shown significant accelerations for the subsonic case, to the point that after a given point the ROM is capable of taking over the full simulation, reverting seldom to the full

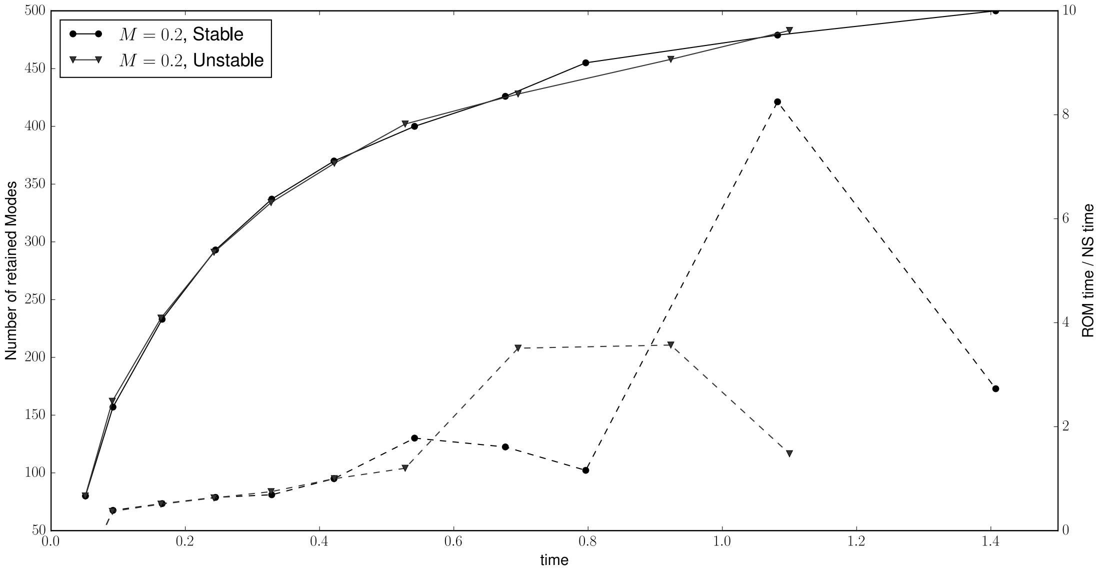

Figure 12: Number of retained modes (——) and ROM vs NS times ratio (- - -) for the M=0.2 aeroelastic cases

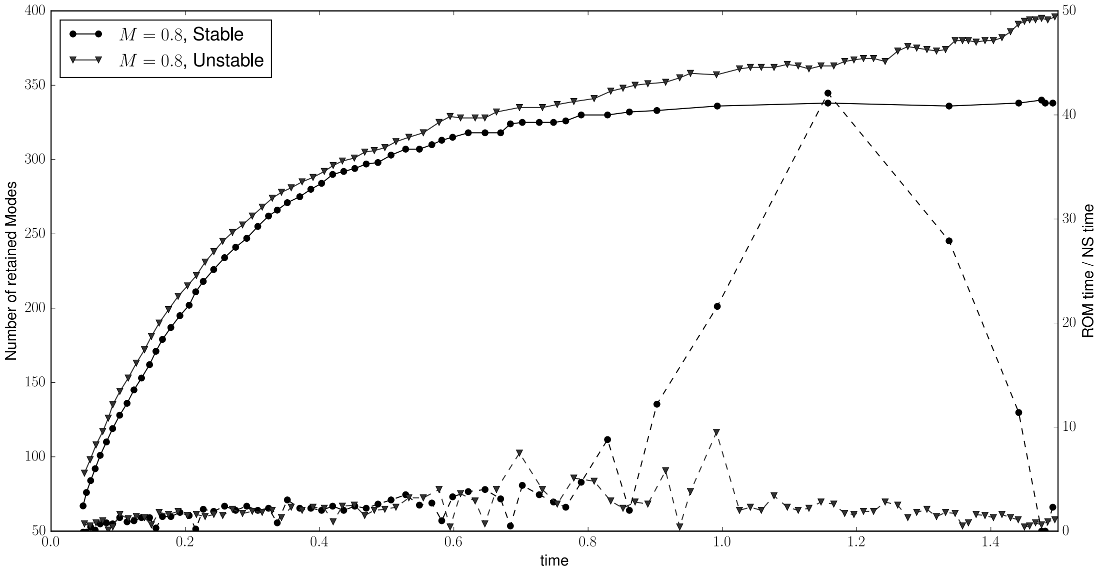

Figure 13: Number of retained modes (——) and ROM vs NS times ratio (- - -) for the M=0.8 aeroelastic cases

solver. In the transonic case the method has to deal with the motion of the strong shocks present on the airfoil, which imposes a shorter validity of the ROM, but the ROM is still capable of significant performance increases.

A further increase in the computational efficiency may result by using generic mode libraries [7], which highly reduces the required value of ΔT^{CFD} (thus, increasing the overal efficiency of the method) when simulating very complex dynamics in the Complex Ginzburg-Landau equation.

Initial tests show very high potential of the method in the case of turbulent viscous simulations. In this case, the turbulence variable evolution ($\tilde{\nu}$ if the Spalart-Allmaras [25] turbulence model is used) can be made dependent of a subset of the primitive variables, so that its residual does not need to be calculated or minimized.

7 REFERENCES

[1] Heeg, J., Chwalowski, P., Florance, J. P., et al. (2013). Overview of the aeroelastic prediction workshop. In *51st AIAA Aerospace Sciences Meeting*. Grapevine, TX, United States.

[2] Moreno, R., Taylor, P., and Newsom, J. (2012). A rigid horizontal tail wind tunnel test for high transonic mach and high frequency unsteady pressure acquisition. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. Honolulu, HI, United States.

[3] Dowell, E., Thomas, J., and Hall, K. (2001). Transonic limit cycle oscillation analysis using reduced order aerodynamic models. In *42nd AIAA ASME ASCE AHS ASC Structures, Structural Dynamics and Materials Conference and Exhibit, Seattle, WA*.

[4] Silva, W. and Bartels, R. (2004). Development of reduced-order models for aeroelastic analysis and flutter prediction using the cfl3dv6.0 code. *Journal of Fluids and Structures*, 19(6), 729 – 745.

[5] Raveh, D. E. (2004). Identification of computational-fluid-dynamics based unsteady aerodynamic models for aeroelastic analysis. *Journal of aircraft*, 41(3), 620–632.

[6] Balajewicz, M. J., Dowell, E. H., and Noack, B. R. (2013). Low-dimensional modelling of high-reynolds-number shear flows incorporating constraints from the navier-stokes equation. *Journal of Fluid Mechanics*, 729, 285–308.

[7] Rapún, M.-L., Terragni, F., and Vega, J. (2015). Adaptive pod-based low-dimensional modeling supported by residual estimates. *International Journal for Numerical Methods in Engineering*, 104(9), 844–868.

[8] Terragni, F. and Vega, J. (2014). Construction of bifurcation diagrams using pod on the fly. *SIAM Journal on Applied Dynamical Systems*, 13, 330–365.

[9] Ryckelynck, D. (2009). Hyper–reduction of mechanical models involving internal variables. *International Journal for Numerical Methods in Engineering*, 77(1), 75–89.

[10] Alonso, D., Vega, J. M., and Velazquez, A. (2010). Reduced-order model for viscous aerodynamic flow past an airfoil. *AIAA Journal*, 48, 1946–1958.

[11] Alonso, D., Velazquez, A., and Vega, J. M. (2009). A method to generate computationally efficient reduced order models. *Computer Methods in Applied Mechanics and Engineering*, 198, 2683–2691.

[12] Terragni, F., Valero, E., and Vega, J. (2011). Local pod plus galerkin projection in the unsteady lid-driven cavity problem. *SIAM Journal on Scientific Computing*, 33, 3538–3561.

[13] Moreno-Ramos, R., Vega, J. M., and Varas, F. (2016). Computationally efficient simulation of unsteady aerodynamics using pod on the fly. *Fluid Dynamics Research*, 48(6), 061424.

[14] Rumsey, C. L., Sanetrik, M. D., Biedron, R. T., et al. (1996). Efficiency and accuracy of time-accurate turbulent navier-stokes computations. *Computers & Fluids*, 25(2), 217–236.

[15] Courant, R., Friedrichs, K., and Lewy, H. (1967). On the partial difference equations of mathematical physics. *IBM journal of Research and Development*, 11(2), 215–234.

[16] Luke, E., Collins, E., and Blades, E. (2012). A fast mesh deformation method using explicit interpolation. *Journal of Computational Physics*, 231(2), 586–601.

[17] Wright, S. and Nocedal, J. (1999). Numerical optimization. *Springer Science*, 35.

[18] Garrick, I. E. (1933). Determination of the theoretical pressure distribution for twenty airfoils. Technical Report 465, NACA.

[19] Batina, J. T. (1985). Effects of airfoil shape, thickness, camber, and angle of sttack on calculated transonic unsteady airloads. Technical Report NASA-TM-86320, NASA.

[20] Rivera, J. J., Dansberry, B., Bennet, R., et al. (1992). Naca 0012 benchmark model experimental flutter results with unsteady pressure distributions. In *Structures, Structural Dynamics, and Materials and Co-located Conferences*. AIAA.

[21] Conner, M., Tang, D., Dowell, E., et al. (1997). Nonlinear behavior of a typical airfoil section with control surface freeplay: a numerical and experimental study. *Journal of Fluids and Structures*, 11(1), 89–109.

[22] Geuzaine, C. and Remacle, J.-F. (2009). Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11), 1309–1331.

[23] Astrid, P., Weiland, S., Willcox, K., et al. (2008). Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control*, 53(10), 2237–2251.

[24] Rapún, M.-L., Terragni, F., and Vega, J. (2017). Lupod: Collocation in pod via lu decomposition. *Journal of Computational Physics*, 335, 1–20.

[25] Spalart, P. and Allmaras, S. (1992). A one-equation turbulence model for aerodynamic flows. *AIAA Journal*, 94.

COPYRIGHT STATEMENT