International Forum on Aeroelasticity and Structural Dynamics
IFASD 2022
13-17 June 2022, Madrid, Spain

ADAPTIVE REDUCED ORDER MODEL OF AEROELASTIC SYSTEMS USING PROPER ORTHOGONAL DECOMPOSITION

Roberto Sánchez-Ortiz[1], **David Quero**[1], **Ruben Moreno-Ramos**[2]

[1]DLR (German Aerospace Center) – Institute of Aeroelasticity
Bunsenstraße 10, 37073 Göttingen, Germany
roberto.sanchezortiz@dlr.de
david.queromartin@dlr.de

[2]E.T.S.I. Aeronáutica y del Espacio, Universidad Politécnica de Madrid
Plaza del Cardenal Cisneros, 28040 Madrid, Spain
ruben.moreno@upm.es

Keywords: Adaptive Reduced Order Models, Proper Orthogonal Decomposition, Computational Aeroelasticity, CFD, Galerkin Projection

Abstract: An adaptive reduced order method is considered to simulate unsteady inviscid aeroelastic scenarios for the AGARD 445.6 wing benchmark case and turbulent simulations of a NACA 0012 airfoil. The method is based on Proper Orthogonal Decomposition (POD) and it is constructed using snapshots from a Computational Fluid Dynamics (CFD) solver, which is coupled with a modal structural solver. The method adaptively switches between the full order model (FOM) and the reduced order model (ROM), which is decided on the fly by monitoring a residual estimate. A least-squares minimisation and a Galerkin projection have been used to evolve the ROM, both applied to a residual calculated only on a subset of cells, which significantly further reduces the computational effort. A collocation method for cell selection is performed automatically via LU decomposition. Various implementations of the method are discussed, such as the inclusion of Dual Time-Stepping terms in the ROM iterations, QR reorthogonalisation of POD modes, and the creation of a separate POD for the turbulent equations. Results of forced motion cases and aeroelastic responses to subcritical flow conditions are presented. The outcome is an efficient and robust method with a reduction down to 11 % of the CPU time required to compute a full CFD solution.

1 INTRODUCTION

It is well known that the most demanding part in aeroelastic computations is the aerodynamic analysis. Currently, the most widely used method in industry for aerodynamic modelling is the Doublet Lattice Method (DLM) [1], which is a linear unsteady method based on potential theory. Since non-linear effects are neglected, there is an emerging demand in the industry for more accurate models such as Computational Fluid Dynamics (CFD).

CFD simulations are becoming standard practice for steady aerodynamics because they are now computationally affordable [2]. Meanwhile, unsteady simulations within the field of aeroelasticity still require substantial computational effort. Reduced order models (ROM) are therefore useful for accurately solving unsteady aerodynamics in a speedy manner. Most ROMs are currently based on some kind of linearisation, yet fully non-linear models have already been developed. These use Proper Orthogonal Decomposition (POD) to decompose the flow field in spatial functions or POD modes by means of Singular Value Decompositions (SVD) applied to a snapshot matrix, collected by the full order model (FOM). This is known as training and can be done previous to the simulation, *offline*, or during the simulation, *online*. This work studies an online method, also called adaptive, where both the FOM and ROM are simulated in an interspersed manner. At the beginning, a long FOM period is used to build a POD basis representative of the flow dynamics, and then shorter FOM periods update the POD basis if some new behaviour is encountered or if there is a loss of validity in the ROM solution. A schematic view of this process is given in Figure 1.

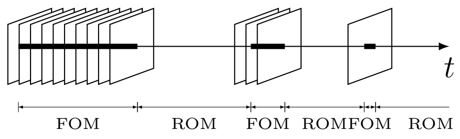


Figure 1: Sketch of the POD simulation, with snapshots (as planes) collected during the full order model (FOM)

Additionally, the computational cost can be further reduced by solving the ROM in a subset of cells which are representative of the total domain as in [3–5], so that non-linear terms are computed only on a few cells. This method contrasts with standard POD-Galerkin, which requires full evaluation of non-linear terms, and with other methods such as the Discrete Empirical Interpolation Method (DEIM) [6, 7] in simplicity, because the solution in this work is achieved by directly minimising the residual in the subset of cells. Moreover, the cell selection is made automatically as in [8] in what was called the LUPOD method.

In particular, an application of these methods to aeroelastic simulations were made in [9] for inviscid and viscous simulations of a NACA airfoil and in [10] for the automatic selection of cells in a NACA airfoil and the AGARD wing.

The present work continues the development of the LUPOD method of [9] and [10] applied to the AGARD 445.6 wing, particularly to the weakened-3 model of [11] as an inviscid case, and to the NACA 0012 airfoil as a turbulent case. Some innovations are also proposed, such as Dual Time-Stepping terms in the Jacobian, QR reorthogonalisation of POD modes and a Galerkin Projection. For the turbulent case, three different implementations of the method are discussed, in addition to the use of a Levenberg–Marquardt least-squares algorithm to converge the ROM solution.

2 FULL ORDER MODEL

The Full Order Model used in this work consists of a Computational Fluid Dynamics solver coupled with a structural solver, which were initially developed in [9, 10] with Python and using distributed-memory parallel computing. The CFD solver is an unstructured, cell-centred solver of the unsteady Euler and RANS equations where the convective terms are computed using the Jameson–Schmidt–Turkel (JST) dissipative central scheme [12, 13], the viscous terms use gradients evaluated as a combination of averages and directional derivatives [14], and the time integration is performed with a second-order backward method using a Dual Time-Stepping technique [15] with CFL-based pseudo-time steps. Regarding the turbulence modelling, the one-equation Spalart–Allmaras [16] model is used.

Some efficiency improvements to the CFD code have been additionally implemented in this work with the aim to bring performance closer to that of current industrial codes and hence make a fairer analysis in terms of computational cost reduction when comparing with the ROM. These improvements include the addition of the JST dissipative terms in the Jacobian, the Reverse

Cuthill-McKee algorithm for sorting linear systems, enhancements to MPI communications, and the use of precompiled code for CPU-intensive functions.

After every physical iteration of the CFD solver, the forces over the wing are integrated and projected into the generalised forces, F_i, of the structural model, which is solved providing a new state of the generalised coordinates, X_i. The problem is thus solved as a loosely coupled system since the characteristic time of the structural movement is much larger than that of the aerodynamics [9].

Finally, the mesh deformation is made with the VTK library [17]. The modes from the structural model are interpolated using a Gaussian kernel into the wing surface of the computational mesh used in CFD, and then extended to the volumetric mesh, giving a matrix ϕ_v of deformation modes. In this way, the displacements of the mesh deformation, $\Delta \boldsymbol{q}_v$, are directly proportional to the generalised coordinates of the structural model, $\boldsymbol{X}$, as $\Delta \boldsymbol{q}_v = \phi_v \boldsymbol{X}$.

3 REDUCED ORDER MODEL

The reduced order model is based on Proper Orthogonal Decomposition (POD). It is an online adaptive method, thus the name *POD on the fly*, because the POD modes are updated throughout the simulation. When the ROM is no longer accurate, the system reverts back to the full order solver in order to capture behaviours that were not previously encountered. The problem is further reduced by solving the equations on a subset of *master cells*, which will be automatically selected as those capturing relevant information from the fluid and allowing a reconstruction of the full domain solution.

3.1 LUPOD method

The state vector in master cells, indicated with subscript c, will be represented by a linear combination of POD modes as

$$\boldsymbol{W}_c = \boldsymbol{W}_c^0 + \Phi \boldsymbol{\xi}, \qquad \boldsymbol{\xi} \in \mathbb{R}^m \tag{1}$$

where $\boldsymbol{W}_c^0$ is the steady solution, Φ are the m POD modes, and $\boldsymbol{\xi}$ are the coefficients or generalised coordinates. To obtain the POD modes, the fluid state variables will be collected during the full order simulation followed by a singular value decomposition. The flowchart in Figure 2 provides a general overview of the LUPOD method.

Initially, the full order model is simulated over a timespan ΔT_{CFD} of n_t iterations, while the difference of state vectors of primitive variables $[\rho, \boldsymbol{u}, p]^T \in \mathbb{R}^N$ with respect to the steady state from each iteration are stored in a matrix of *snapshots* as

$$S = \begin{bmatrix} | & | & & | \\ \mathbf{\Delta W}^1 & \mathbf{\Delta W}^2 & \dots & \mathbf{\Delta W}^{n_t} \\ | & | & & | \end{bmatrix}_{N \times n_t} \tag{2}$$

3.1.1 *Automatic master cell selection*

Before applying the single value decomposition to the snapshot matrix, the computation domain is reduced to a subset of cells, namely the *master cells*. The method to select the subset of cells is automatic and it enables to perform 3D simulations without an initial manual selection of cells. It was already implemented in [10] and followed the work of [8], which consists in applying the LU decomposition, or Gauss elimination, to the snapshot matrix, S. A number of linearly independent rows and columns are selected, which can be used to reconstruct the complete snapshots as linear combinations. The cells associated to these selected rows are chosen as the master cells. Note that this method depends on a tunable truncation threshold defined as

$$\epsilon_{\mathrm{LU}} = \frac{\|S'_m\|_{\max}}{\|S\|_{\max}} \tag{3}$$

where S'_m is the state of the matrix in step m of the process. The value of ϵ_{LU} has been set to 10^{-6}. The method works in a parallel memory-distributed way, so that a complete snapshot matrix does not need to be stored in a single node.

3.1.2 Singular Value Decomposition

Since the master cell selection is performed before the mode extraction, the Singular Value Decomposition to extract the POD modes can now be applied to an already-reduced snapshot

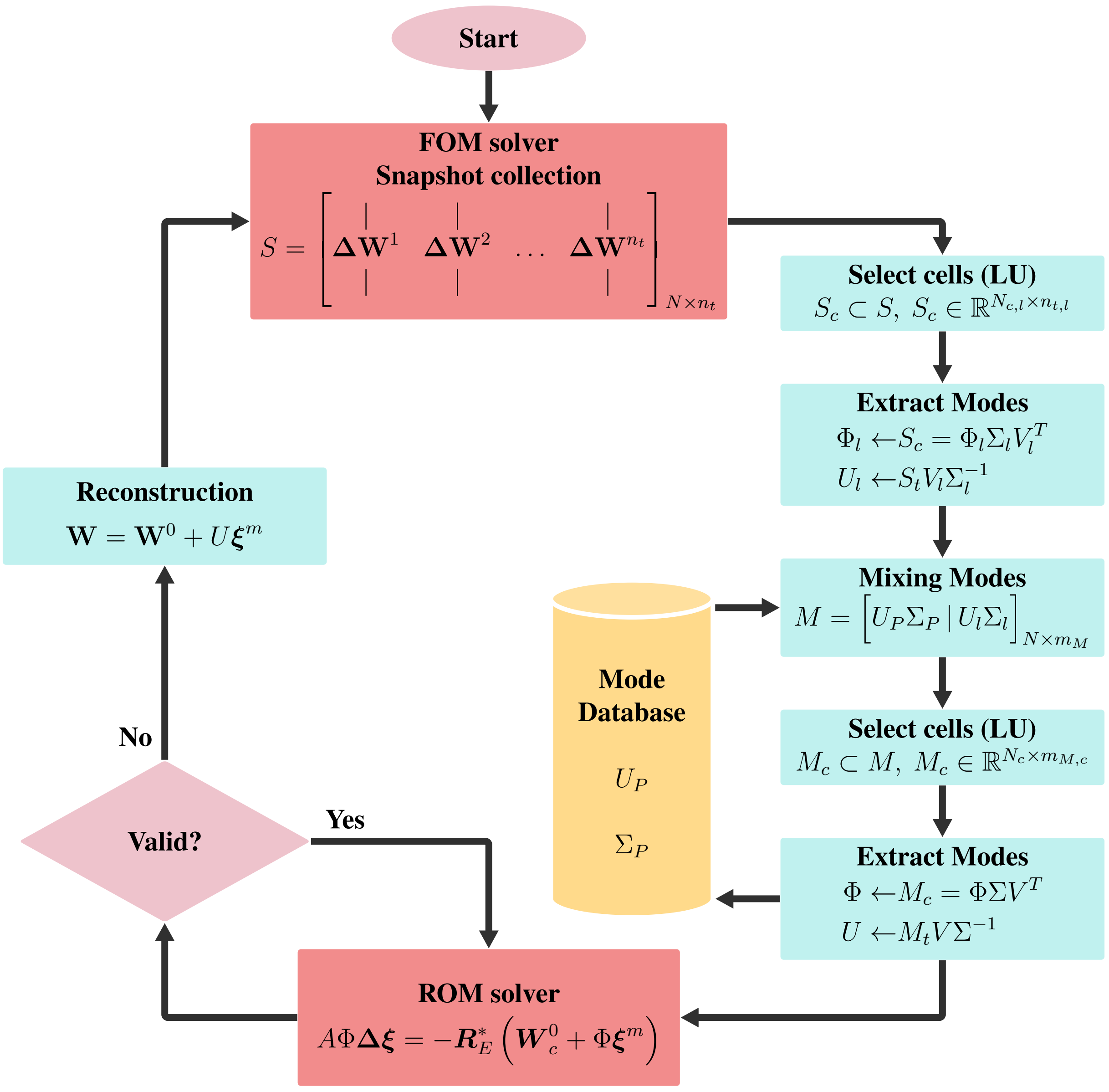

Start
FOM solver
Snapshot collection
$S = \begin{bmatrix} | & | & & | \\ \Delta\mathbf{W}^1 & \Delta\mathbf{W}^2 & \dots & \Delta\mathbf{W}^{n_t} \\ | & | & & | \end{bmatrix}_{N\times n_t}$
Select cells (LU)
$S_c \subset S,\ S_c \in \mathbb{R}^{N_{c,l}\times n_{t,l}}$
Extract Modes
$\Phi_l \leftarrow S_c = \Phi_l \Sigma_l V_l^T$
$U_l \leftarrow S_t V_l \Sigma_l^{-1}$
Mixing Modes
$M = \left[U_P\Sigma_P \,|\, U_l\Sigma_l\right]_{N\times m_M}$
Select cells (LU)
$M_c \subset M,\ M_c \in \mathbb{R}^{N_c\times m_{M,c}}$
Extract Modes
$\Phi \leftarrow M_c = \Phi\Sigma V^T$
$U \leftarrow M_t V \Sigma^{-1}$
Mode Database
U_P
Σ_P
Reconstruction
$\mathbf{W} = \mathbf{W}^0 + U\boldsymbol{\xi}^m$
No
Valid?
Yes
ROM solver
$A\Phi\boldsymbol{\Delta\xi} = -\boldsymbol{R}_E^*\left(\boldsymbol{W}_c^0 + \Phi\boldsymbol{\xi}^m\right)$

Figure 2: Flowchart of the LUPOD method

matrix $S_c \in \mathbb{R}^{N_c \times n_{tc}}$, where N_c is the total number of rows (which is n_c times the number of equations per cell), n_c is the number of master cells, and n_{tc} is the number of selected columns or snapshots. This matrix is build at the master rank after receiving the previously selected rows and columns of the snapshot matrix from each rank. The singular value decomposition is applied to this matrix S_c as

$$S_c \approx \Phi \Sigma V^T \tag{4}$$

which gives a matrix $\Phi \in \mathbb{R}^{N_c \times m}$, whose columns are the m left-singular vectors or *modes* of the SVD.

POD modes associated with singular values $\Sigma_i/\Sigma_0 < 10^{-8}$ are not valid in double precision [18]. Therefore, the matrix, Φ, is truncated to the modes such that $\Sigma_i/\Sigma_0 > \epsilon_{\text{POD}}$. A study of the influence of this parameter was already carried out in [9] and it was concluded that lower values of this threshold significantly increased the number of POD modes retained over time, but the efficiency of the ROM remained the same. In this work, the truncation value has been set to a fixed value of $\epsilon_{\text{POD}} = 10^{-6}$ for all cases. Figure 3 shows a usual distribution of singular values.

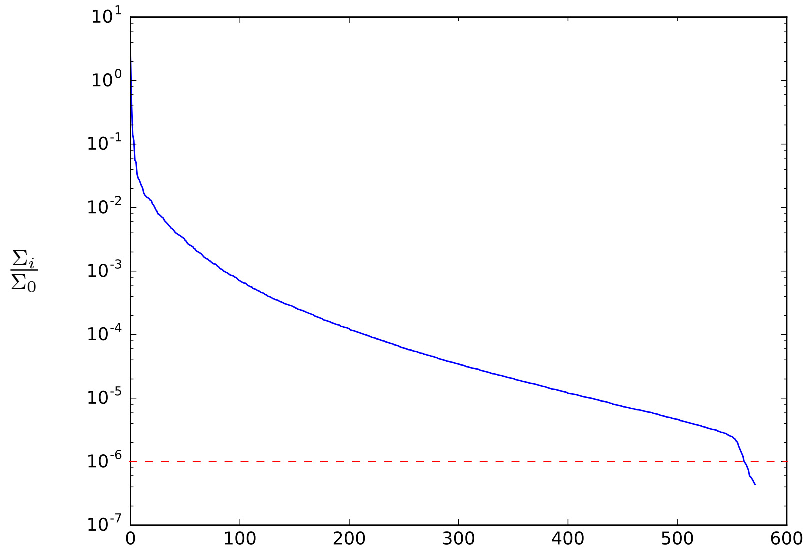


Figure 3: Relative Singular Values normalised by the first singular value as Σ_i/Σ_0 (solid line) and truncated by the parameter $\epsilon_{\text{POD}} = 10^{-6}$ (dashed line).

Reconstruction of the state vector in the rest of cells can be achieved by a linear combination of the *extended modes*, U, which are defined independently at each rank simply as

$$U = S_t V \Sigma^{-1}, \qquad U \in \mathbb{R}^{N \times m} \tag{5}$$

where S_t are the n_{tc} complete snapshots previously selected.

This is consistent with the way the master cells have been selected (details in section 3.1.1). Previous snapshots of flow variables of these cells are able to reconstruct the solution over the complete domain because variables of the rest of cells are linearly dependent of those of the master cells.

3.1.3 POD basis update

A two-step extraction of modes allows us to mix the new POD modes extracted from the snapshots of the last FOM period with the modes from previous ROMs without the need of accumulating all the snapshots since the very beginning of the simulation. The method requires also a double selection of master cells, one right before each mode extraction.

The recombination is made by appending the new modes, U_l, with the modes from the previous iteration, U_P, both multiplied by their singular values, Σ, as

$$M = \left[U_P \Sigma_P \mid U_l \Sigma_l\right] \tag{6}$$

and then computing a second master cell selection and the singular value decomposition. These U_l and U_P modes are taken from a common set of rows of the reconstruction modes U of the new mode extraction and the previous ROM, respectively. The common rows are those corresponding to the accumulation of every master cell ever selected.

This process is equivalent to building a matrix by appending snapshots, if truncation errors are neglected,

$$\tilde{M} = \left[\, S_P \mid S \,\right] = \left[U_P \Sigma_P V_P^T \mid U_l \Sigma_l V_l^T\right] \tag{7}$$

It can be shown that there is no need to include the right-singular vectors, V, because they would cancel out in the computation of the left-singular vectors of $\tilde{M}$. By definition, the left-singular vectors of M are the eigenvectors of MM^T, therefore, proving that $MM^T = \tilde{M}\tilde{M}^T$ will suffice.

$$\tilde{M}\tilde{M}^T = U_P \Sigma_P V_P^T V_P \Sigma_P U_P^T + U_l \Sigma_l V_l^T V_l \Sigma_l U_l^T = U_P \Sigma_P^2 U_P^T + U_l \Sigma_l^2 U_l^T = MM^T \tag{8}$$

As already mentioned, it is also necessary to update the cell selection before applying the SVD to the matrix M, which is done with the LU decomposition as in section 3.1.1. Again a singular value decomposition is applied to a reduced matrix, M_c, extracted from the selected rows and columns of M, as

$$M_c = \Phi \Sigma V^T \tag{9}$$

Finally, the extended modes for reconstruction of the full fluid state are defined the same way as before using M_t, which is the $m_{M,c}$ columns of M selected by the LU method.

$$U = M_t V \Sigma^{-1} \tag{10}$$

3.2 Residual minimisation

The general solution at each time step will be a combination of POD modes given by the generalised coordinates $\boldsymbol{\xi}$, which will minimise the pseudo-steady residual $\boldsymbol{R}_{Ei}^*$ defined by the Dual-Time strategy [15]. It is evaluated only on master cells, which greatly reduces the computational cost as it is largely composed of non-linear functions. Therefore, the ROM can be exclusively computed by a single core.

There are two ways to approach the minimisation: directly minimising the least square residual on the master cells, or using a Galerkin projection.

3.2.1 Least squares

The least squares minimisation, $\min \left\| \boldsymbol{R}^{*}_{Ec} \right\|$, is achieved with a Gauss–Newton algorithm that solves the non-linear least squares problem on the master cells iteratively as

$$\frac{\partial \boldsymbol{R}^{*}_{Ei}}{\partial \boldsymbol{W}_{j}} \boldsymbol{\Delta W}_{c} = -\boldsymbol{R}^{*}_{E} \left(\boldsymbol{W}^{m}_{c} \right) \tag{11}$$

where $\boldsymbol{W}_c$ is replaced with the linear combination of POD modes, $\boldsymbol{W}^0_c + \Phi\boldsymbol{\xi}$. This linear system can also be expressed as $A \cdot \boldsymbol{\Delta\xi} = \boldsymbol{b}$, where A is a dense matrix. It is solved using the Moore–Penrose inverse, which is defined as

$$A^{\dagger} = \left(A^{T} \cdot A \right)^{-1} \cdot A^{T} \tag{12}$$

Finally, the solution is $\boldsymbol{\Delta\xi} = A^{\dagger} \cdot \boldsymbol{b}$.

The FOM solver uses a Dual Time-Stepping (DTS) method [15] for solving the unsteady aerodynamics, unlike the ROM solver in [9, 10], which uses a simple Gauss–Newton iteration scheme. This work also considers the inclusion of DTS-like terms in the ROM solver, which gives the following system of equations

$$\left[\frac{\boldsymbol{\Omega}^{m}_{i}}{\Delta\tau_{i}} \delta_{ij} + \frac{\partial \boldsymbol{R}^{*}_{Ei}}{\partial \boldsymbol{W}_{j}} \right] \Phi \boldsymbol{\Delta\xi} = -\boldsymbol{R}^{*}_{E} \left(\boldsymbol{W}^{0}_{c} + \Phi \boldsymbol{\xi}^{m} \right) \tag{13}$$

where $\Delta\tau_i$ are the pseudo-time steps computed with the same CFL condition as in the FOM. It might seem that these terms has little physical meaning because there is no CFL condition in the reduced model. Nevertheless, this approach will improve the convergence, allowing the POD to last more iterations without losing validity according to the criterion of equation (16).

3.2.2 Galerkin Projection

A Galerkin projection is an alternative approach where the residual is projected over the POD modes using an inner product as

$$\left\langle \, \Phi, \; \boldsymbol{R}^{*}_{Ec} \left(\boldsymbol{W}_{c} \right) \, \right\rangle = 0 \tag{14}$$

The projection is restricted to the master cells and the usual dot product is taken as the inner product, giving the following square system, which will be solved with a LU decomposition.

$$\Phi^{T} \frac{\partial \boldsymbol{R}^{*}_{Ei}}{\partial \boldsymbol{W}_{j}} \Phi \boldsymbol{\Delta\xi} = -\Phi^{T} \boldsymbol{R}^{*}_{E} \left(\boldsymbol{W}^{0}_{c} + \Phi \boldsymbol{\xi}^{m} \right) \tag{15}$$

It will be shown that this approach produced better results in terms of stability than the minimisation of least squares in inviscid cases. In RANS cases, however, Galerkin is highly unstable and only the least squares minimisation produced successful results.

3.3 Validity of solution

The ROM solver may lose its validity over time due to new challenges in the flow field not encountered before or truncation errors. The lost of validity is detected by monitoring the root mean square (RMS) of the residual [18] so that the ROM integration stops and the solver reverts to the FOM if the next condition is met.

$$\frac{\left\|\boldsymbol{R}^*_{Ec}\left(\boldsymbol{W}^0_c + \Phi\boldsymbol{\xi}^{n+1}\right)\right\|}{\left\|\boldsymbol{R}_{Ec}\left(\boldsymbol{W}^0_c + \Phi\boldsymbol{\xi}^{n}\right)\right\|} > \epsilon_K \tag{16}$$

where the threshold has been set to $\epsilon_K = 10^{-2}$.

3.4 QR Reorthogonalisation

There can be slightly non-orthogonal modes due to round-off errors, especially when small singular values are retained [10]. In this case, a reorthogonalisation might be helpful and can be easily accomplished by a QR decomposition of the modes as

$$\Phi = QR \tag{17}$$

where Q is an orthogonal matrix, and R is an upper triangular matrix. The reorthogonalisation is applied to both the POD modes Φ and reconstruction modes U by multiplying them with the inverse of R and reassigning the values as

$$\Phi := \Phi R^{-1} \quad \text{and} \quad U := UR^{-1} \tag{18}$$

It was observed that QR reorthogonalisation generally improved performance, while it did not deteriorate any case. Later investigations also showed that a similar behaviour can be obtained by using a more accurate algorithm to compute the singular value decomposition instead of a QR reorthogonalisation. The originally used LAPACK divide-and-conquer ‘`gesdd`’ algorithm followed by a QR reorthogonalisation performs as well as the rectangular general approach ‘`gesvd`’ algorithm alone. We can conclude that the accuracy of the decomposition is key to produce a POD basis stable over time.

3.5 Implementation in turbulent simulations

Turbulent cases present a challenge on how to apply the LUPOD method to the additional variables and equations for turbulence modelling. Three different approaches were studied and are detailed below.

- The easiest approach consists in including the turbulent variables in the snapshot matrix as just an additional variable, so that the snapshots are defined as $[\rho, \boldsymbol{u}, p, \tilde{\nu}]^T$, and the rest of the LUPOD method application remains the same. This led to having POD modes strongly influenced by the values of the turbulent variable, which tends to have much higher values than the other flow variables. The ROM was unable to provide a valid solution for any time, and so this approach was discarded.
- The second approach tries to model the complete physics by extracting modes only from the flow variables $[\rho, \boldsymbol{u}, p]^T$. The turbulent variable is made *slave* to the flow variables, so that only the flow equations are solved while the turbulence is just updated at each iteration by a linear combination of modes $U_{\tilde{\nu}}$. These modes are defined as an extension of the POD modes of the flow variables as

$$U_{\tilde{\nu}} = S_{\tilde{\nu}} V \Sigma^{-1} \tag{19}$$

where $S_{\tilde{\nu}}$ are the turbulent snapshots, and V and Σ come from the SVD of the flow snapshots. This resembles the way the reconstruction modes U were built, and assumes certain linearity between the turbulent and flow variables. The ROM in this case managed to simulate a few iterations before becoming unstable, but it never took over the simulation successfully. Therefore, this approach was eventually abandoned as well as the previous one.

- The last approach involves creating two sets of POD modes, Φ and $\Phi_{\tilde{\nu}}$, extracted independently with a SVD of flow snapshots $[\rho, \boldsymbol{u}, p]^T$ and turbulent snapshots $[\tilde{\nu}]^T$, respectively. The solution is achieved by minimising both the flow and turbulent residual independently, but the iterations are performed in a interspersed manner. Finally, this option delivered satisfactory results, as the ROM was active for significant periods of time. Results presented in section 5.1 are obtained with this approach.

When it comes to the cell selection, different ways of applying the LU method also arose, such as applying it to the flow snapshots alone, or to a complete snapshot matrix also containing the turbulent variables. The most successful solution was again obtained by applying the method to the flow and turbulent snapshot matrices separately. The master cells are then the union of both selections.

3.5.1 Levenberg–Marquardt Algorithm

The turbulent simulations presented problems with the stability of the ROM when using the minimisation of least-squares, and even more with the Galerkin projection. Therefore, an alternative is sought for the minimisation of the residual via the Levenberg–Marquardt algorithm [19,20]. This algorithm uses the following iterative formula to find the minimum,

$$\left(A^T \cdot A + \lambda I\right)^{-1} \cdot A^T \tag{20}$$

where A is the same dense matrix of the system in (11), λ is a damping factor which is adjusted at each iteration, and I is the identity matrix. This formula differs from the Gauss–Newton (12) only in the extra damping factor, which is why this method is also known as a damped least-squares. This method lies then between the Gauss–Newton algorithm ($\lambda = 0$) and the method of gradient descend ($\lambda \gg 1$).

The algorithm begins with an initial value of λ_0, which is increased or decreased depending on the change of the RMS residual after every iteration. The parameter is lowered by ν_{down} when the iteration provides a lower residual, or increased by ν_{up} when the iteration provides a higher residual. In this latter case, the solution is also discarded. Results of this algorithm are provided in section 5.1.2.

4 INVISCID CASE

The method has been applied to the AGARD 445.6 wing, in particular, to the weakened-3 model of [11], which considers four structural modes.

The computational mesh, plotted in Figure 4, consists of a box that extends 66 c_r in front of the wing and 100 c_r behind it, with a height of 133 c_r and a width of 50 c_r in the span direction, where c_r is the root chord (0.5587 m). It is composed of $40\,000$ triangles on the wing surface and a total of $200\,000$ tetrahedra.

Two categories of cases have been proposed: coupled aeroelastic cases and forced motion cases, which are presented in the next sections.

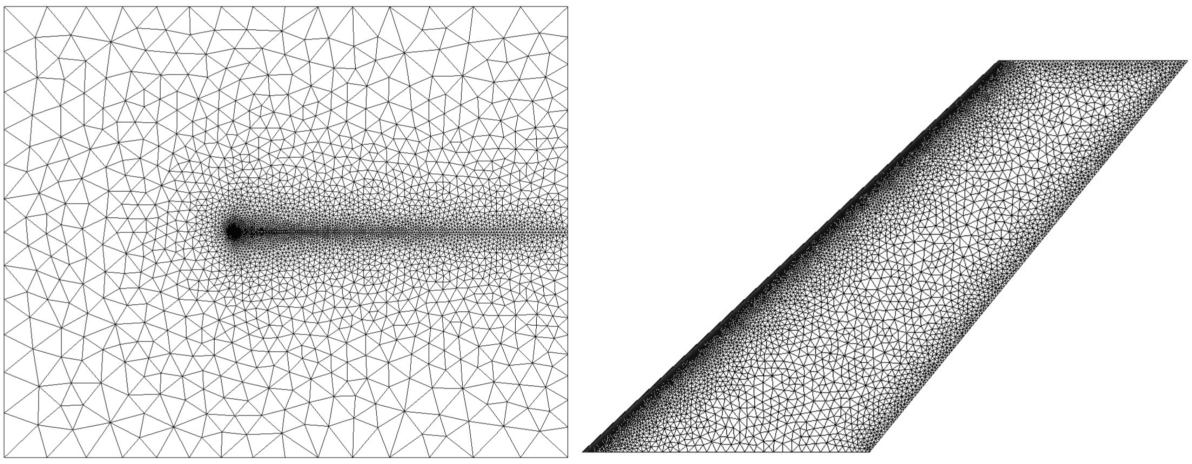

(a) symmetry section (b) wing surface

Figure 4: AGARD 445.6 computational mesh

4.1 Coupled Aeroelastic Cases

In coupled aeroelastic cases, the fluid solver is coupled with a structural solver. The aeroelastic response is computed after an initial impulse of value 0.01 on the first torsion mode speed with the flow conditions below the flutter point at $q_\infty = 5540$ Pa shown in Table 1.

Table 1: Subcritical conditions

M_∞	a_∞	q_∞
0.678	341.3 $\mathrm{m\,s^{-1}}$	4117.7 Pa

4.2 Forced Motion Cases

The purpose of these forced motion simulations is to test the behaviour of the POD method under harmonic motions and whether it will become self-sufficient. If the POD correctly extract the modes out of the unsteady CFD simulation and once the transient has damped out, the ROM is expected to resolve the flow indefinitely without reverting back to the full CFD solver, since it will not encounter new behaviour under a completely periodic response.

The analyses consist of a sinusoidal motion in the generalised coordinates of the first bending mode (X_0) or the first torsion mode (X_1) or both, defined as $X_i(t) = A_i \sin(2\pi f_i(t - T_i))$, whose parameters are defined in Table 2.

Table 2: Forced Motion conditions

	A_0	f_0[Hz]	T_0[s]	A_1	f_1[Hz]	T_1[s]
FM1	0.00065	9.6	0	0	0	0
FMp2	0.00065	9.6	0	$0.0004\Theta(t-0.5)$	38.17	0.5

4.3 Inviscid results

Every case simulation has been carried out with the following parameters of the LUPOD method so that the results can be easily compared.

$$\Delta T_{\mathrm{CFD}} = 0.0025 \text{ s}, \quad \epsilon_{\mathrm{POD}} = 10^{-6}, \quad \epsilon_{\mathrm{LU}} = 10^{-6}, \quad \epsilon_{\mathrm{K}} = 10^{-2} \tag{21}$$

For this configuration, a typical number of master cells selected by the LU method was found to be around 500 once the ROM took over the simulation. The same number was also reported for the mode extraction.

The results are represented as the evolution over time of the first bending, X_0, and first torsion, X_1, generalised coordinates in the case of the coupled aeroelastic simulation, and as the evolution of the two first generalised forces, F_0 and F_1, associated also with the first bending and first torsion modes, in the case of a forced motion simulation. The time spans when the ROM is active are indicated as shaded areas.

Additionally, two ratios are given for every simulation: an activation ratio, defined as the quotient of the simulation time that the ROM has been active over the total simulation time ($\Delta t_{\mathrm{ROM}}/\Delta t_{\mathrm{TOTAL}}$), and a CPU reduction ratio, defined as the quotient of the CPU time of a hybrid POD simulation (FOM and ROM) over the CPU time of a standard CFD simulation ($\Delta t_{\mathrm{CPU,POD}}/\Delta t_{\mathrm{CPU,CFD}}$). A summary of these ratios of the subcritical and forced motion cases is given in section 6.

4.3.1 Simulation with Least-Squares Minimisation

The first simulations were carried out with the same schemes and options as in [10], namely, least-squares minimisation of the ROM residual, albeit with the new dissipative terms included in the Jacobian computation. These conditions will be referred as the base options hereinafter. The results of the subcritical aeroelastic response and the forced motion cases showed activation ratios around 45 %, which led to the CPU time being halved.

With the goal of increasing the activation range of the ROM, a modification to the Jacobian as in equation (13) has been tried out. It involves the inclusion of dual-time-stepping-like terms, motivated by the way the equations are integrated in the FOM. In Figure 5, the subcritical case shows an improvement over the base simulation. In particular, the activation ratio increased

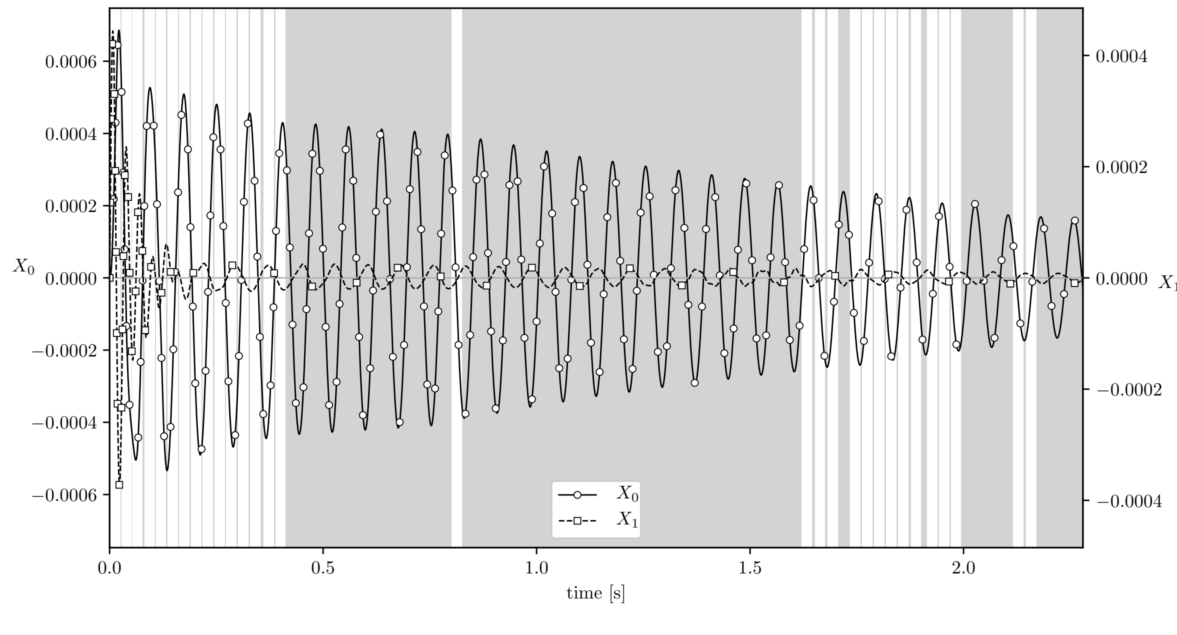


Figure 5: Time evolution of generalised coordinates, X, of the subcritical inviscid AGARD case with DTS terms and least-squares minimisation. Shaded areas indicate when the ROM is active.

from 0.45 to 0.67, while the CPU reduction ratio dropped from 0.5 to 0.32, with respect to the base simulation. Very narrow strips can be seen every ΔT_{CFD} from the beginning, which correspond to every attempt of the ROM to overtake the simulation. It does not activate successfully until $t \simeq 0.4$ s, which is approximately 1.8 times the time it takes for a fluid particle to go down the wake until reaching the mesh border. It presented a continuous ROM span of 1.8 s. Nonetheless, it eventually switched back to the full order model, which suggests that the ROM was not self-sufficient.

On the other hand, the forced motion case behaves perfectly in Figure 6. As might be expected, the ROM is able to capture the complete fluid physics of the simulation in a few cycles, after which it becomes autonomous without the need to switch back to the FOM solver. Its activation and CPU ratios are 0.9 and 0.11, respectively.

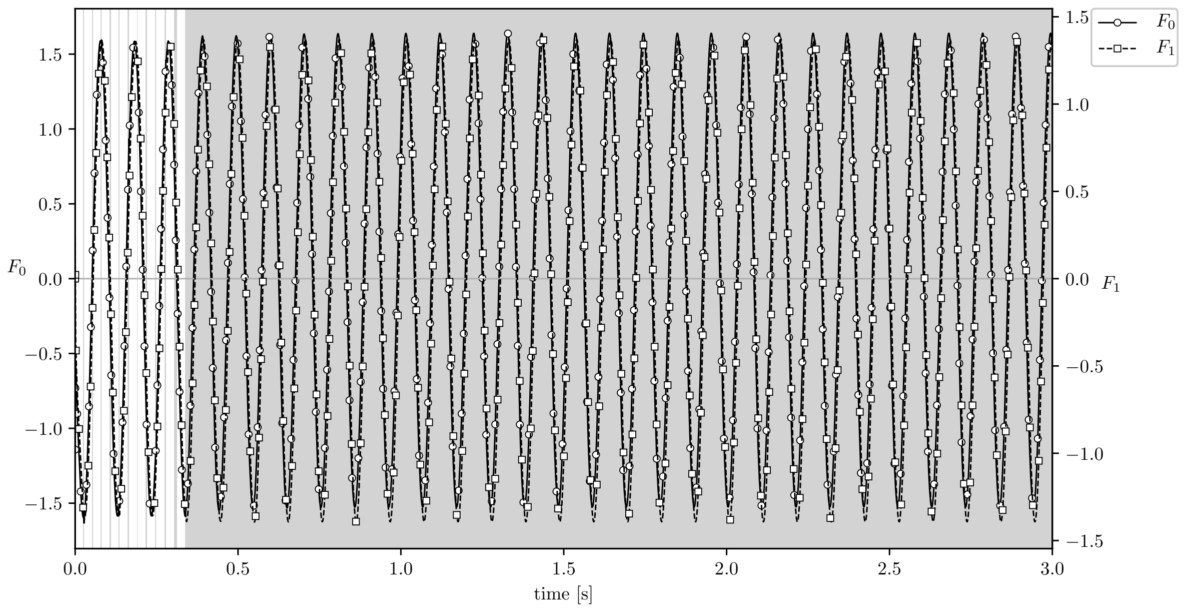


Figure 6: Time evolution of generalised forces, F, of the forced motion inviscid AGARD case FM1 with DTS terms. Shaded areas indicate when the ROM is active.

It is additionally concluded that, although the DTS-like terms seem non-physical, they make the Jacobian more diagonally dominant, improving the convergence.

Given the success of the previous forced motion case, a more complex case has been considered. It is referred as FMp2 in Table 2 and combines two harmonic oscillations in the first bending mode and the first torsion mode. The torsion oscillation is present since the beginning of the simulation, but the bending oscillation starts once the ROM is active at $t = 0.5$ s. Figure 7 shows that after the second oscillation begins, the ROM switches back to the FOM, as expected. Then, it takes 0.25 s to collect new POD modes and work properly again. Once the ROM is active again, the performance is not as good as in the previous case FM1. The activation ratio is 0.5 and the CPU ratio is 0.55, which is more than satisfactory given the complexity of the case.

4.3.2 QR Reorthogonalisation

As stated in section 3.4, there might be slightly non-orthogonal POD modes, which can be corrected with a QR decomposition. Its application to the subcritical case greatly improved the results, as shown in Figure 8 where there is an endless active ROM from $t \approx 0.4$ s with an astonishing activation ratio of 0.88 and a CPU ratio of 0.14.

However, the forced motion case FM1 in Figure 9, deteriorated with respect to the previous section after the inclusion of the QR decomposition. The activation ratio is 0.67 compared to

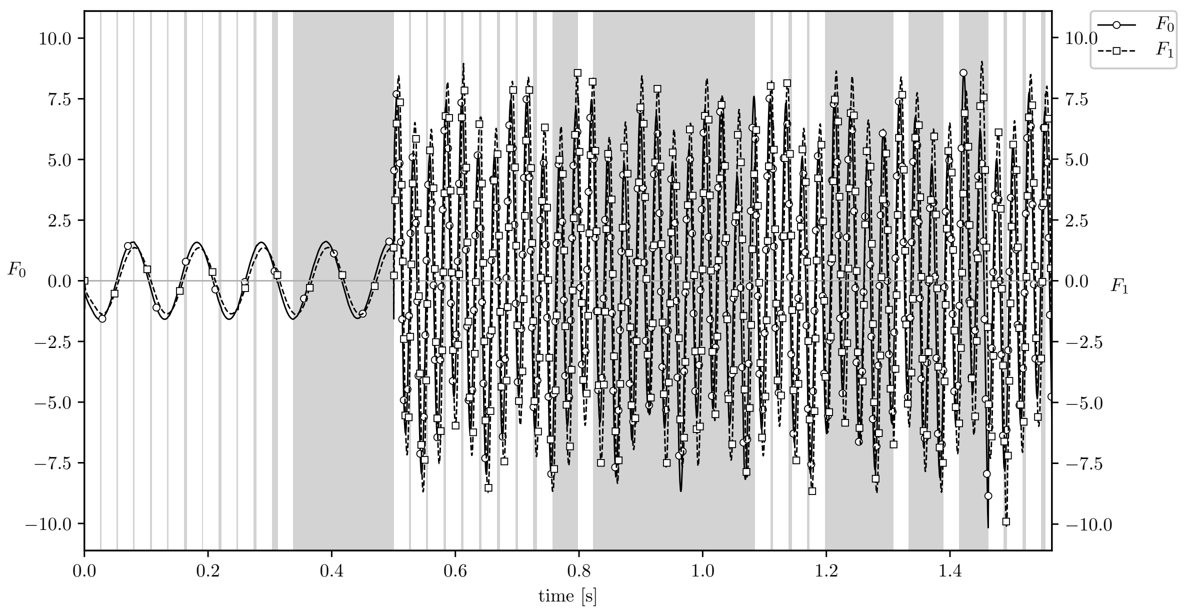


Figure 7: Time evolution of generalised forces, F, of the forced motion inviscid AGARD case FMp2 with DTS terms and least-squares minimisation. Shaded areas indicate when the ROM is active.

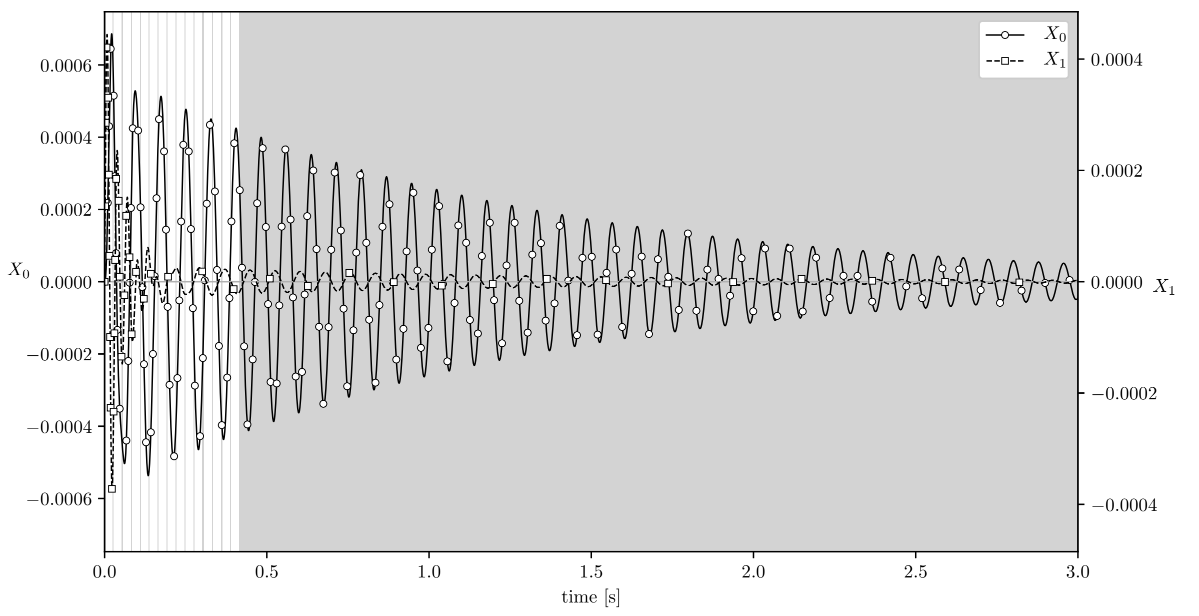


Figure 8: Time evolution of generalised coordinates, X, of the subcritical case with DTS terms, QR reorthogonalisation, and least-squares minimisation. Shaded areas indicate when the ROM is active.

the previous 0.9, and the CPU ratio is 0.3 compared to the previous 0.11. This was found to be caused by the emergence of noise in low-energy modes, which appeared after many iterations in the residual minimisation if the ROM solver was not stopped promptly.

It is highly advisable to establish a convergence criterion that is triggered when the RMS residual stalls, similarly to a Cauchy criterion. Otherwise, additional iterations would only cause noise in low-energy modes without further convergence in the solution. Unlike in the least-squares minimisation, these convergence issues were not observed when using a Galerkin pro-

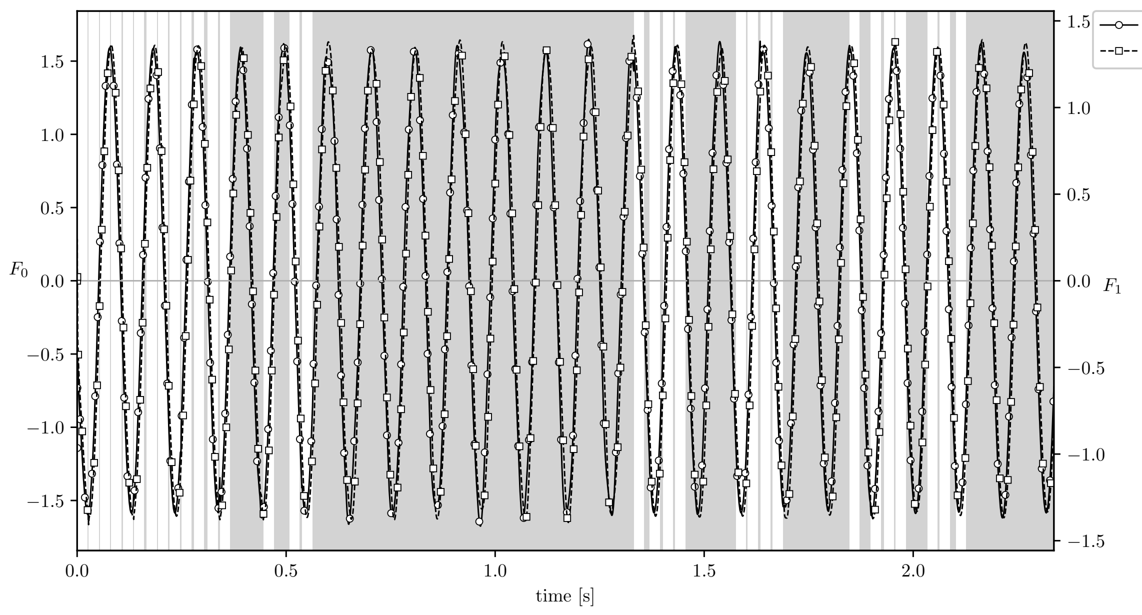


Figure 9: Time evolution of generalised forces, F, of the forced motion inviscid AGARD case FM1 with DTS terms, QR reorthogonalisation and least-squares minimisation. Shaded areas indicate when the ROM is active.

jection, see the following section.

4.3.3 Simulations with Galerkin Projection

Simulations with Galerkin Projection gave as good results as the best cases with least-square minimisation. Again, QR reorthogonalisation also improved the overall ROM behaviour. This can be seen in Figure 10 for the subcritical inviscid case with a remarkable activation ratio of 0.89 and a CPU reduction ratio of 0.12.

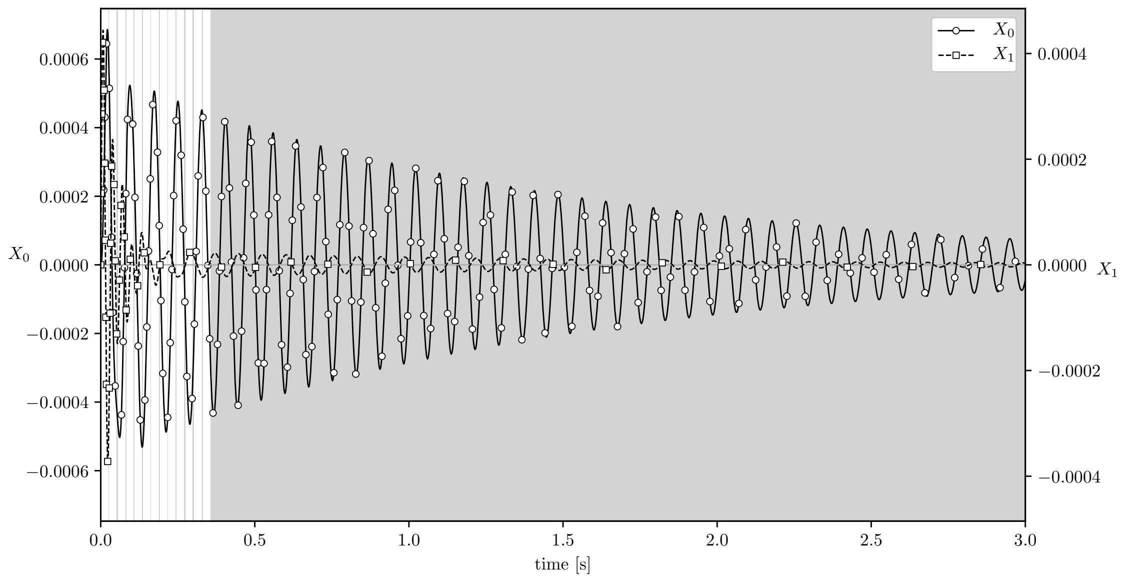


Figure 10: Time evolution of generalised coordinates, X, of the subcritical inviscid AGARD case with Galerkin projection and QR reorthogonalisation. Shaded areas indicate when the ROM is active.

There is an additional advantage over the least-squares minimisation. Simulations with Galerkin

Projection do not present the problem with convergence criteria related to noise mentioned in the previous section. The generalised coordinates of the Galerkin cases, including those associated with low-energy modes, showed a very smooth trajectory throughout the simulation, in contrast to some of the least-squares minimisation cases.

5 TURBULENT CASE

The LUPOD method is now applied to the turbulent simulation of the NACA 0012 airfoil. The computational mesh, plotted in Figure 11, is a hybrid C-shape mesh with a 150-chord radius and 200 chords in the wake direction. The mesh extrusion satisfies the $y+ \sim 1$ condition at the wall while having a progression rate of 1.2 for the first 15 layers and of 1.3 for the following 15 layers. It is composed of $70\,000$ elements.

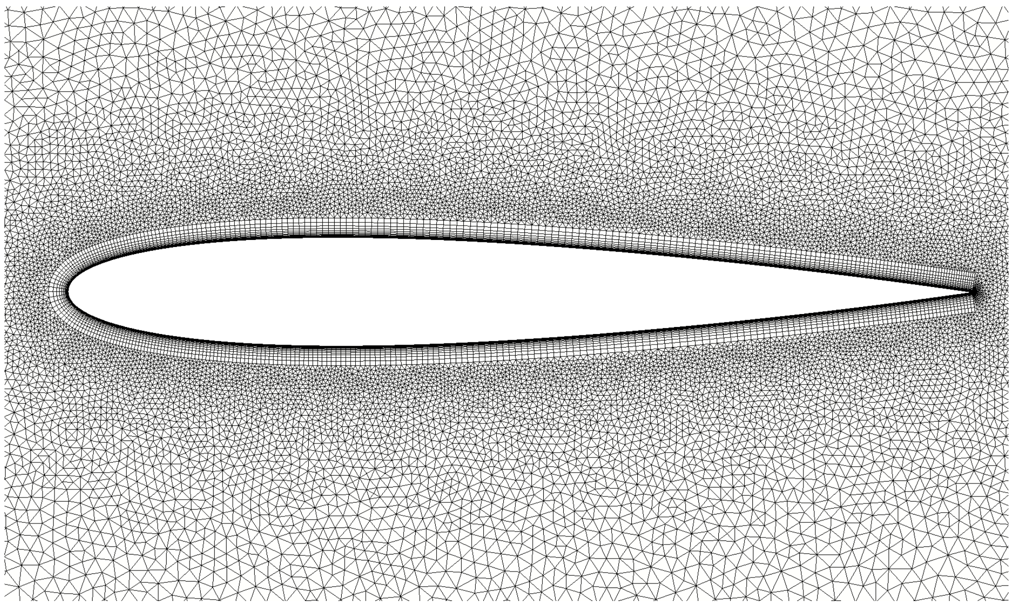


Figure 11: NACA 0012 computational mesh

The following simulations consist of a forced harmonic oscillation in pitch with an amplitude of 0.001 rad and a frequency of 10 Hz, which can be expressed as $\theta(t) = 0.001 \sin(2\pi \times 10 \times t)$. The flow conditions are a Reynolds number of 6 millions and a subsonic Mach number of 0.3.

5.1 Turbulent results

Every case simulation has been carried out with the following parameters of the LUPOD method so that the results can be easily compared.

$$\Delta T_{\text{CFD}} = 0.0025 \text{ s}, \quad \epsilon_{\text{POD}} = 10^{-6}, \quad \epsilon_{\text{LU}} = 10^{-6}, \quad \epsilon_{\text{K}} = 10^{-2} \tag{22}$$

For most simulations, around 900 master cells were selected, while the number of POD modes was around 500 for the flow variables and 400 for the turbulent variables.

The results are represented as the evolution over time of the vertical and horizontal forces, F_y and F_x, respectively. The time spans when the ROM is active are indicated again as shaded areas. Finally, the same ratios defined in section 4.3 are also provided.

5.1.1 Simulation of Least-Squares Minimisation with Gauss–Newton algorithm

The first simulations were carried out with the same schemes as the most successful cases of the inviscid simulations in an attempt to replicate the good results obtained before.

Unfortunately, this was not the case. As can be seen in Figure 12, the ROM using a least-squares minimisation together with a QR reorthogonalisation showed narrower intervals than in inviscid cases and did not fully succeed in taking over the simulation after more than 0.7 seconds. Additionally, after ending every ROM interval and reverting back to the FOM, there is quite a discrepancy in the integrated forces. This was found to be caused by a progressive accumulation of small errors with respect to the FOM, which ended up making the FOM diverge at 0.77 s. The activation ratio was 0.3 and the CPU reduction ratio was around 0.87.

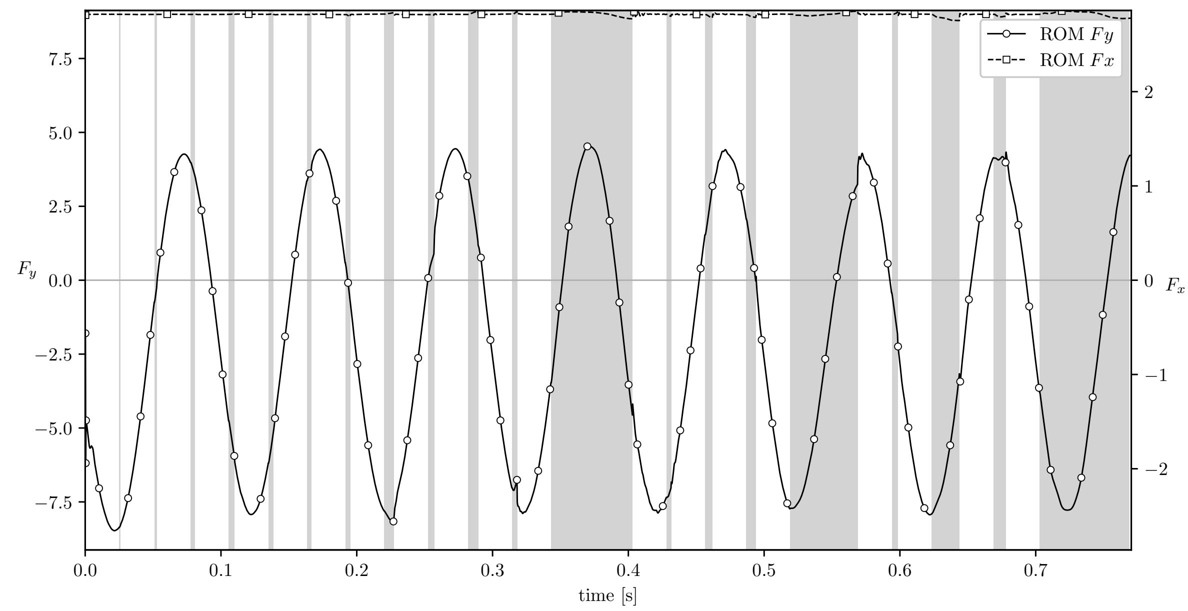


Figure 12: Time evolution of integrated forces, F, of the turbulent NACA0012 case with least-squares minimisation and QR reorthogonalisation. Shaded areas indicate when the ROM is active.

On the other hand, the Galerkin projection made the ROM much more unstable and did not give any results worth reporting.

5.1.2 Simulation with Levenberg–Marquardt algorithm

This method is also known as damped least-squares and is an attempt to obtain a more robust solution than the standard least-squares. The parameters defined in section 3.5.1 were taken as

$$\lambda_0 = 1.0, \quad \nu_{\text{up}} = 10.0, \quad \text{and} \quad \nu_{\text{down}} = 3.0 \tag{23}$$

The first thing noticed was that this method tended to be slower during residual minimisation, requiring twice as many inner iterations and each iteration also being more costly since the system matrix changes every time. It is nonetheless worthwhile, because the ROM stability improved, as Figure 13 shows. Moreover, this case did not diverged, unlike the one of the previous section, achieving a longer simulation time. The activation ratio was around 0.5 and the CPU reduction ratio was 0.7.

Finally, a comparison of the pressure field between the solution of FOM and ROM at $t = 0.6$ s is provided in Figure 14, where it can be seen that the two contour plots are indistinguishable.

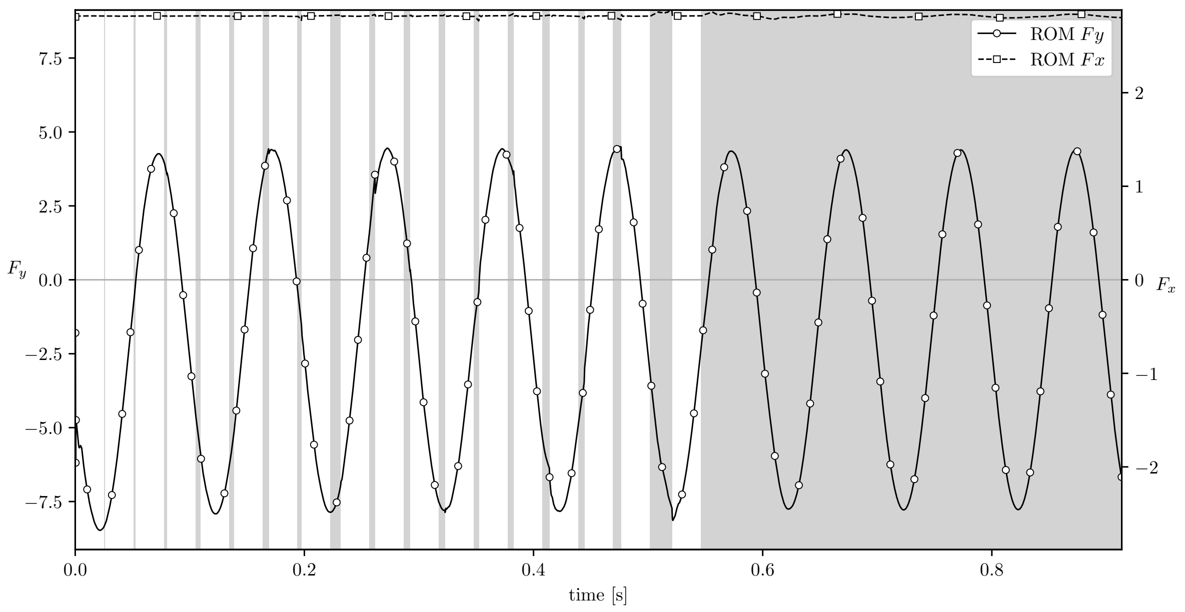
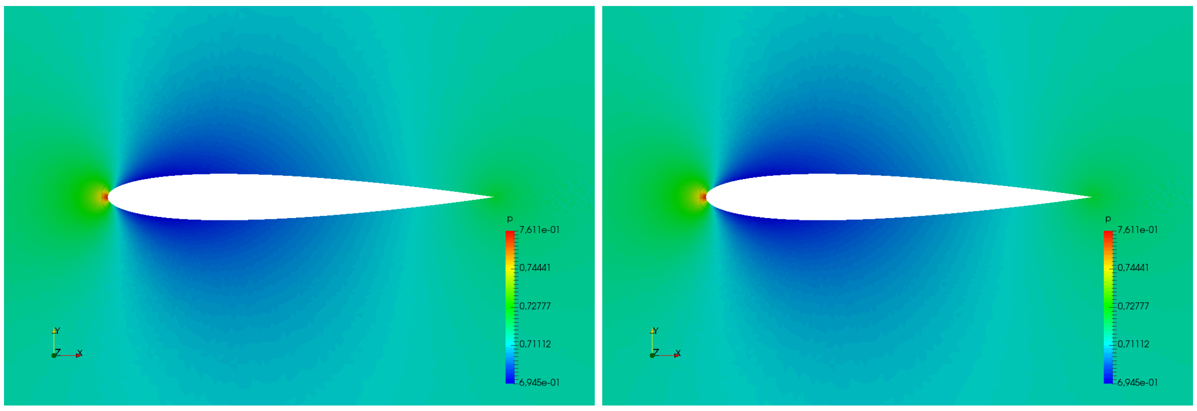


Figure 13: Time evolution of integrated forces, F, of the subcritical turbulent NACA0012 case with Levenberg–Marquardt algorithm. Shaded areas indicate when the ROM is active.



(a) FOM (b) ROM

Figure 14: Non-dimensional pressure field contour at $t = 0.6$ s.

6 CONCLUSIONS

The LUPOD method has been used to create a ROM for solving unsteady aeroelastic systems following the work of [9]. As described in section 3, it is an online adaptive method that extracts POD modes from snapshots collected during FOM periods. The model is further reduced by minimising the residual on a subset of cells, namely the *master cells*, which are automatically selected with the LU method, or using a Galerkin projection with a suited inner product.

In successful cases where the ROM was able to take over the simulation, the number of cells and modes selected was around 500 for inviscid cases and 500+400 for turbulent cases. This proves to be an efficient method in reducing the computational cost of high-fidelity aeroelastic calculations.

The method has been applied to solve high-fidelity aeroelastic calculations, in particular the inviscid aeroelastic response of the AGARD 445.6 wing. Both aeroelastic motion and forced motion simulations have been carried out using the following modifications to the original method.

- In the least-squares minimisation cases, the inclusion of Dual Time-Stepping terms produced a more endurable and stable ROM. As already mentioned, these terms have little physical meaning because there is no Courant condition in the ROM problem. Anyhow, it considerably improved the results.
- Another way of improving the ROM effectiveness was to perform a reorthogonalisation of the POD modes. It was done through a QR decomposition, and it improved the amount of time the ROM was active by 20 %.
- It was also shown that there was an issue with the generation of noise in low-energy modes, which resulted in a loss of accuracy and stability. It can be avoided if the iterations are stopped sufficiently early once the RMS residual has stalled.
- Lastly, it has been analysed the use of a Galerkin Projection to solve the ROM. It performed greatly with the ROM becoming active 90 % of the time and consuming 12 % of CPU time with respect to the full CFD simulation. In addition, it did not show noise-related issues. Again, the QR reorthogonalisation improved even more the results.

Some more challenging cases have been tested, such as a forced motion in two different structural modes beginning at different times. They gave a successful outcome, proving the method to be extremely robust.

A summary of the ROM activation ratios, defined as the ratio of time the ROM is active over the total time ($\Delta t_{\text{ROM}}/\Delta t_{\text{TOTAL}}$), is given in Table 3. A summary of the CPU reduction ratios, defined as the ratio of the CPU time of a POD simulation over the CPU time of a CFD simulation ($\Delta t_{\text{CPU,POD}}/\Delta t_{\text{CPU,CFD}}$), is given in Table 4.

Table 3: Summary of ROM activation ratios

	Subcritical	Forced Motion (FM1)
Base options	0.46	0.44
Dual Time-Stepping Terms	0.67	0.90
DTS + QR reorth.	0.88	0.67
Galerkin	0.69	–
Galerkin + QR reorth.	0.89	0.89

Table 4: Summary of CPU reduction ratios

	Subcritical	Forced Motion (FM1)
Base options	0.50	0.48
Dual Time-Stepping Terms	0.32	0.33
DTS + QR reorth.	0.50	0.30
Galerkin	0.30	–
Galerkin + QR reorth.	0.12	0.12

The method was then applied to the turbulent case of a forced harmonic oscillation of a NACA 0012 airfoil. Several options emerged for the application of the LUPOD method to turbulent cases, and it was concluded that the most successful one was applying the SVD and cell selection independently at the flow and turbulent snapshots. The reduction in CPU time of turbulent cases was not as great as in the inviscid cases due to a more costly ROM, but the final outcome was positive as the cost was reduced compared to the FOM. It is also concluded that the Levenberg–Marquardt algorithm improves the stability of the ROM making it last longer and avoiding divergence as opposed to the Gauss–Newton algorithm. Nevertheless, the discrepancy in the forces is left unresolved and should be studied more thoroughly in the future. A final observation common to all turbulent cases is that the turbulence equation was the most difficult to converge, being responsible for the ROM to exit and revert back to the FOM.

Future work to improve the ROM stability may include the use of weights on the snapshots to redefine the POD projections, and studying methods for residual minimisation in the ROM as alternatives to the least-squares. In addition, the turbulent method may be applied to transonic cases and to 3D flows.

7 REFERENCES

[1] Bisplinghoff, R. L., Ashley, H., and Halfman, R. L. (1996). *Aeroelasticity*. New York: Dover Publications Inc., new edition ed. ISBN 978-0-486-69189-3.

[2] Weigold, W., Stickan, B., Travieso-Alvarez, I., et al. (2017). Linearized Unsteady CFD for Gust Loads with TAU. In *IFASD 2017 - International Forum on Aeroelasticity and Structural Dynamics*. Como, Italien. ISBN 978-88-97576-28-0.

[3] Alonso, D., Velazquez, A., and Vega, J. (2009). A method to generate computationally efficient reduced order models. *Computer Methods in Applied Mechanics and Engineering*, 198. doi:10.1016/j.cma.2009.03.012.

[4] Alonso, D., Vega, J. M., and Velazquez, A. (2010). Reduced-Order Model for Viscous Aerodynamic Flow Past an Airfoil. *AIAA Journal*, 48(9), 1946–1958. ISSN 0001-1452. doi:10.2514/1.J050153. Publisher: American Institute of Aeronautics and Astronautics.

[5] Terragni, F., Valero, E., and Vega, J. M. (2011). Local POD Plus Galerkin Projection in the Unsteady Lid-Driven Cavity Problem. *SIAM Journal on Scientific Computing*, 33(6), 3538–3561. ISSN 1064-8275. doi:10.1137/100816006. Publisher: Society for Industrial and Applied Mathematics.

[6] Chaturantabut, S. and Sorensen, D. C. (2010). Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM Journal on Scientific Computing*, 32(5), 2737–2764. ISSN 1064-8275. doi:10.1137/090766498. Publisher: Society for Industrial and Applied Mathematics.

[7] Peherstorfer, B. and Willcox, K. (2015). Online Adaptive Model Reduction for Nonlinear Systems via Low-Rank Updates. *Society for Industrial and Applied Mathematics*. ISSN 1064-8275.

[8] Rapun, M.-L., Terragni, F., and Vega, J. (2017). LUPOD: Collocation in POD via LU decomposition. *Journal of Computational Physics*, 335. doi:10.1016/j.jcp.2017.01.005.

[9] Moreno Ramos, R. (2017). *Adaptive Reduced Order Modeling of Aeroelastic Flows Based on Proper Orthogonal Decomposition*. Ph.D. thesis, E.T.S.I. Aeronáuticos (UPM).

[10] Moreno Ramos, R., Varas, F., and Vega, J. M. (2019). Self Adaptive POD based ROM 3D Aeroelastic Simulations. Savannah, Georgia, USA: International Forum on Aeroelasticity and Structural Dynamics, IFASD.

[11] Yates, E. and Carson, J. (1987). AGARD standard aeroelastic configurations for dynamic response. Candidate configuration I.-wing 445.6. Tech. Rep. TM-100492, NASA Langley Research Center, Hampton, VA, USA.

[12] Jameson, A., Schmidt, W., and Turkel, E. (1981). Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes. In *14th Fluid and Plasma Dynamics Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics. doi:10.2514/6.1981-1259.

[13] Jameson, A. (2017). Origins and Further Development of the Jameson–Schmidt–Turkel Scheme. *AIAA Journal*, 55(5), 1487–1510. ISSN 0001-1452. doi:10.2514/1.J055493. Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.J055493.

[14] Blazek, J. (2005). *Computational Fluid Dynamics: Principles and Applications*. Amsterdam Heidelberg: Elsevier Science, 2nd edition ed. ISBN 978-0-08-044506-9.

[15] Jameson, A. (1991). Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. In *10th Computational Fluid Dynamics Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics. doi:10.2514/6.1991-1596.

[16] Spalart, P. and Allmaras, S. (1992). A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting and Exhibit*, Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics. doi:10.2514/6.1992-439.

[17] Schroeder, W., Martin, K., and Lorensen, W. (2006). *The Visualization Toolkit*. 4 ed. ISBN 978-1-930934-19-1.

[18] Rapún, M.-L., Terragni, F., and Vega, J. M. (2015). Adaptive POD-based low-dimensional modeling supported by residual estimates. *International Journal for Numerical Methods in Engineering*, 104(9), 844–868. ISSN 1097-0207. doi:10.1002/nme.4947.

[19] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2), 164–168. ISSN 0033-569X, 1552-4485. doi:10.1090/qam/10666.

[20] Marquardt, D. W. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2), 431–441. ISSN 0368-4245. doi:10.1137/0111030. Publisher: Society for Industrial and Applied Mathematics.

COPYRIGHT STATEMENT