Chapter 15
Accelerating Unsteady CFD Simulations Using a Minimum Residual Based Nonlinear Reduced Order Modeling Approach

Matteo Ripepi and Stefan Görtz

Abstract Reduced-order modeling is evaluated as a means to speed up unsteady computational fluid dynamics (CFD) simulations while maintaining the desired level of accuracy. In the reduced order modeling approach, proper orthogonal decomposition (POD) is applied to some computed response time history from a compressible, unsteady CFD solver to compute a set of orthogonal basis vectors. An approximate flow solution for the next time step is predicted by minimizing the unsteady flow solver residual in the space spanned by the POD basis. This is done by solving a non-linear least-squares problem. This approximate flow solution is then used to initialize the flow solver at this time step, aiming to reduce the number of inner iterations of the dual time stepping loop to convergence compared to the conventional choice of initializing with the previous time step solution or an extrapolation in time. This procedure is repeated for all following time steps. Results for the pitching LANN wing at transonic flow conditions show a more than twofold reduction in the number of inner iterations of the flow solver to convergence. Despite the overhead caused by evaluating the reduced-order model (ROM) at every time step, the method results in a 38% savings in computational time without compromising accuracy, thus improving the overall efficiency for unsteady aerodynamics applications. Finally, several means to further improve the performance are also discussed, including updating the POD basis after every new time step.

15.1 Introduction

Reduced-order models (ROMs) are being used to replace computationally expensive full-order models in different fields of application, including computational fluid dynamics (CFD). Typically, ROMs are of interest in situations where the same

M. Ripepi · S. Görtz (✉)
DLR Institute of Aerodynamics and Flow Technology, Braunschweig, Germany
e-mail: stefan.goertz@dlr.de

M. Ripepi
e-mail: matteo.ripepi@dlr.de

R. Heinrich (ed.), *AeroStruct: Enable and Learn How to Integrate Flexibility in Design*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 138, https://doi.org/10.1007/978-3-319-72020-3_15

full-order model is to be evaluated many times for different parameter settings, such as different flow conditions. The goal is to make many predictions at lower computation cost, hence efficient but nonetheless sufficiently accurate ROMs are typically sought after.

A powerful tool currently considered state of the art in reduced-order modeling of linear and nonlinear systems is proper orthogonal decomposition (POD), a technique which has been demonstrated in many fields of application. When applied in the context of CFD, the basic idea of POD is to replace solving the full-order governing equations of fluid dynamics by determining a suitable linear combination of POD basis vectors, which are computed based on flow solution snapshots from selected full-order CFD simulations. There exist various methods for computing the coefficients of such a linear combination, including interpolation [1] and solving a low-order partial-differential equation (PDE) system. Following the approach in [2], effective POD-based ROMs for steady aerodynamic problems can also be obtained by minimizing the POD approximation's defect with respect to the governing equations of fluid dynamics. This minimum residual based nonlinear reduced order modeling approach has been further refined and demonstrated for subsonic and transonic steady aerodynamic applications in [3, 4].

Here, our idea is to extend this approach to unsteady aerodynamic applications by minimizing the unsteady residual. Although derived from high-fidelity unsteady CFD data, the resulting ROM approximation may lead to a substantial error in time in the predicted response and time accuracy is lost. Hence, our second idea is to use the ROM prediction for a given time step to initialize the full-order model at this time step, assuming this is a good initial guess, and then iterating the inner loop of dual time stepping of the flow solver until convergence is achieved for this time step. Ideally, the number of inner iterations to convergence is reduced in this fashion compared to how the inner loop is otherwise initialized. This procedure is conducted repetitively for a series of times steps, whereby for each and every new ROM prediction, the snapshot set and thus the POD basis may be augmented with the latest fully converged time step solution.

This idea of speeding up the computation of a series of computations using a residual-based reduced-order model has already been demonstrated for steady aerodynamic problems [5]. Here, the corresponding strategy is demonstrated for unsteady aerodynamic applications by using the extended ROM formulation.

15.2 Theoretical Background

The developed model order reduction approach for unsteady aerodynamic applications is based on a least-squares minimization of the unsteady residual, which is obtained by approximating the flow solution through modes arising from a Proper Orthogonal Decomposition (POD) of samples data. Given a set of flow solutions to the full order CFD model $\mathbf{w}(t) = [\rho, \rho\mathbf{v}, \rho E^t] \in \mathbb{R}^N$, N being the total number of flow states (number of conservative variables per grid point times number of grid points n), ρ the density, $\mathbf{v}$ the velocity vector, E the total energy, at different time steps t_k, with $k = 1, \ldots, m$, the POD yields an optimal basis for representing reduced order solutions of the governing equations. The idea of the reduced order modeling approach is to formulate the discretized unsteady equations as a steady-state problem for every time step in a similar fashion as dual-time stepping is used in solving unsteady CFD problems. This yields the so-called unsteady residual:

$$\widehat{\mathbf{R}} \overset{def}{=} \mathbf{R}(\mathbf{w}(t)) + \boldsymbol{\Omega}\frac{\partial \mathbf{w}(t)}{\partial t} = \mathbf{0} \qquad \boldsymbol{\Omega} : \text{cell volumes} \tag{15.1}$$

By discretizing the time derivative (assuming equidistant time steps Δt and an invariant computational grid) with a second-order accurate backward difference formula (BDF-2), it is:

$$\widehat{\mathbf{R}} \overset{def}{=} \mathbf{R}(\mathbf{w}(t_{k+1})) + \boldsymbol{\Omega}\frac{3\mathbf{w}(t_{k+1}) - 4\mathbf{w}(t_k) + \mathbf{w}(t_{k-1})}{2\Delta t} = \mathbf{0} \tag{15.2}$$

15.2.1 Unsteady Residual-Based Reduced-Order Modeling

The idea is to minimize the unsteady residual in the space spanned by the POD basis vectors $\mathbf{U}_r$, which is obtained by applying a singular value decomposition (SVD) to the snapshot matrix $\mathbf{Y} = \mathbf{USV^T}$, and by truncating the matrix $\mathbf{U}$ retaining only the most relevant r eigenvectors (i.e., those associated to the largest singular values). The snapshot matrix is obtained by collecting the flow solutions at different time steps and subtracting the average of the snapshots $\overline{\mathbf{w}} = \frac{1}{m}\sum_{k=1}^{m} \mathbf{w}_k$, i.e., $\mathbf{Y} = [\mathbf{w}(t_1), \ldots, \mathbf{w}(t_m)] - \overline{\mathbf{w}}$. As an alternative to the SVD, an Eigenvalue decomposition (EVD) of the correlation matrix $\mathbf{R} = \mathbf{YY}^T = \mathbf{US}^2\mathbf{U}^T$ can be applied to obtain the POD modes. As this space is of reduced size compared to the original problem, the minimization problem is as well.

The problem is formulated by searching for an approximate flow solution $\widetilde{\mathbf{w}}(t_k)$ in the subspace $\mathbf{U}_r \in \mathbb{R}^{Nxr}$, $r \ll N$, where only the r most relevant basis vectors have been retained:

$$\widetilde{\mathbf{w}} = \sum_{i=1}^{r} a_i \mathbf{U}_i + \overline{\mathbf{w}} = \mathbf{U}_r\mathbf{a} + \overline{\mathbf{w}} \tag{15.3}$$

with $\mathbf{a}$ being the vector of POD coefficients, minimizing the unsteady residual in the L_2 norm:

$$\underset{\mathbf{a}}{min}\|\widehat{\mathbf{R}}(\mathbf{U}_r\mathbf{a} + \overline{\mathbf{w}})\|_{L_2}^2 = \underset{\mathbf{a}}{min}\sum_{j=1}^{N} \Omega_j \widehat{R}_j^2(\mathbf{U}_r\mathbf{a} + \overline{\mathbf{w}}) \tag{15.4}$$

The arising nonlinear least-squares problem for the POD coefficients $\mathbf{a}$ is solved by using a Levenberg algorithm [7]. So, an iterative procedure is performed where the increment to coefficients $\Delta\mathbf{a}$ is obtained by solving the linear system:

$$(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\Delta\mathbf{a} = -\mathbf{J}^T\widehat{\mathbf{R}} \tag{15.5}$$

with $J_{ij} = \frac{\partial \widehat{R}_i}{\partial a_j} \in \mathbb{R}^{N\times r}$ the Jacobian matrix of $\widehat{\mathbf{R}}$ with respect to the POD coefficients $\mathbf{a}$, and the non-negative damping factor λ, which is automatically adjusted at each iteration depending on the convergence rate of the algorithm. Each component of the gradient vector $\mathbf{g} \equiv -2\mathbf{J}^T\widehat{\mathbf{R}}$ can be scaled according to the curvature, here approximated by the pseudo-Hessian matrix $\mathbf{B} \equiv \mathbf{J}^T\mathbf{J}$, in order to avoid slow convergence in the direction of small gradients. Therefore, replacing the identity matrix $\mathbf{I}$ with the diagonal matrix consisting of the diagonal elements of the pseudo-Hessian matrix, leads to the well-known Levenberg–Marquardt algorithm, which solves iteratively the linear problem:

$$(\mathbf{J}^T\mathbf{J} + \lambda\mathrm{diag}(\mathbf{J}^T\mathbf{J}))\Delta\mathbf{a} = -\mathbf{J}^T\widehat{\mathbf{R}} \tag{15.6}$$

The rank-one Broyden's method is used to approximate the Jacobian of the reduced-order system of equations, from the knowledge of the Jacobian matrix (exact or approximated) built at the previous iteration step:

$$\mathbf{J}_{k+1} = \mathbf{J}_k + \frac{\Delta\widehat{\mathbf{R}} - \mathbf{J}_k\Delta\mathbf{a}}{\|\Delta\mathbf{a}\|^2}\Delta\mathbf{a}^T \tag{15.7}$$

so avoiding the time consuming computation of the $\mathbf{J}$ by finite differences at each iteration of the minimization procedure. Moreover, thanks to Broyden's update procedure, the expensive matrix-matrix computation $\mathbf{B} \equiv \mathbf{J}^T\mathbf{J}$ can be avoided. Indeed, by directly substituting Broyden's formula in the matrix product, one can obtain a way to computing also the pseudo-Hessian matrix from the knowledge of its values during the previous iteration step:

$$\begin{aligned}
\mathbf{B}_{k+1} &\equiv \mathbf{J}\,{}^T_{k+1}\mathbf{J}_{k+1} && (15.8)\\
&= \mathbf{B}_k + \frac{\mathbf{J}_k^T\Delta\widehat{\mathbf{R}} - \mathbf{B}_k\Delta\mathbf{a}}{\|\Delta\mathbf{a}\|^2}\Delta\mathbf{a}^T + \Delta\mathbf{a}\frac{\Delta\widehat{\mathbf{R}}^T\mathbf{J}_k - \Delta\mathbf{a}^T\mathbf{B}_k}{\|\Delta\mathbf{a}\|^2} && (15.9)\\
&+ \Delta\mathbf{a}\frac{\Delta\widehat{\mathbf{R}}^T\Delta\widehat{\mathbf{R}} - \Delta\widehat{\mathbf{R}}^T\mathbf{J}_k\Delta\mathbf{a} - \Delta\mathbf{a}^T\mathbf{J}_k^T\Delta\widehat{\mathbf{R}} + \Delta\mathbf{a}^T\mathbf{B}_k\Delta\mathbf{a}}{\|\Delta\mathbf{a}\|^4}\Delta\mathbf{a}^T && (15.10)
\end{aligned}$$

which can be rewritten in the compact form:

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\boldsymbol{\omega}_k\Delta\mathbf{a}^T + \Delta\mathbf{a}\boldsymbol{\omega}_k^T}{\|\Delta\mathbf{a}\|^2} + (\boldsymbol{\phi}_k^T\Delta\widehat{\mathbf{R}} - \boldsymbol{\omega}_k^T\Delta\mathbf{a})\frac{\Delta\mathbf{a}\Delta\mathbf{a}^T}{\|\Delta\mathbf{a}\|^4} \tag{15.11}$$

with $\boldsymbol{\omega}_k \equiv \mathbf{J}_k^T\Delta\widehat{\mathbf{R}} - \mathbf{B}_k\Delta\mathbf{a}$, and $\boldsymbol{\phi}_k \equiv \Delta\widehat{\mathbf{R}} - \mathbf{J}_k^T\Delta\mathbf{a}$.

Using this formula, the computation of $\mathbf{B}$ (done at each iteration of the minimization process, for every physical time step) has complexity $\mathscr{O}(Nr + 16r^2 + N) \cong$

$\mathscr{O}(Nr)$, instead of $\mathscr{O}(Nr^2)$, where N is the order of the high-dimensional model (the number of conservative variables times the number of grid points) and r the order of the reduced order model (i.e. the number of POD modes used). It must be pointed out that this formula for updating the pseudo-Hessian it is not resulting from a direct approximation of $\mathbf{B}$, but relies on the approximation of $\mathbf{J}$ through Broyden's update of the Jacobian matrix.

15.2.2 Initializing the Flow Solver with ROM Predictions

The predicted ROM solution is successively used as a restarting solution, initializing the DLR's CFD code TAU [6] at the time step under consideration. The aim is to reduce the number of inner iterations to converge the residual with respect to the conventional choice of restarting the unsteady computation using the previous time step solution. The ROM is used therefore as a predictor providing an improved initial guess for the iterative process.

The ROM-based initial guess is based on the information of the solutions collected for a certain number of previous time steps, thus realizing a moving window strategy. In other words, the predicted ROM solution is a linear combination of the most energetic POD modes of the snapshots collected progressively during the running simulation. Local ROMs are thus computed by applying the POD on the snapshots taken at the various time intervals, and updated after a certain number of time steps have been computed. Such approach leads to predictions based on the most recent snapshots, and therefore is more adequate for analysis where certain events exist over a relatively short time intervals, so to better capturing transient phenomena. The advantage of such an approach is that the POD is cheap and leads to low-dimensional ROMs, since only few snapshots are considered. However the POD must be applied many times during the simulation, therefore increasing the total online computational cost of the ROM prediction. Moreover, a poor ROM prediction may happen when the POD is computed over a set of TAU snapshots, collected during a certain time interval, which do not contain information about a phenomena or behavior that will be present in the following time steps to be simulated (e.g. the appearance of a shock wave due to a change in the angle of attack). Restarting the CFD solver TAU with such a poor ROM prediction may lead to an increased number of iterations to converge, compared to restarting the flow solver using the TAU flow solution at the previous time step.

Alternatively, a previously computed POD basis (e.g., generated based on data from a training maneuver exciting a broad band of frequencies and amplitudes) may be employed. This approach aims to collect snapshots coming from a time simulation covering a large domain of the solution space where the flow solutions to be predicted are assumed to lie. The drawback is that the POD is applied to a greater number of snapshots compared to the moving-window approach, and the resulting POD basis is larger in size and thus the ROM more expensive to evaluate. However, only one POD has been computed offline and updating during the online prediction is not required.

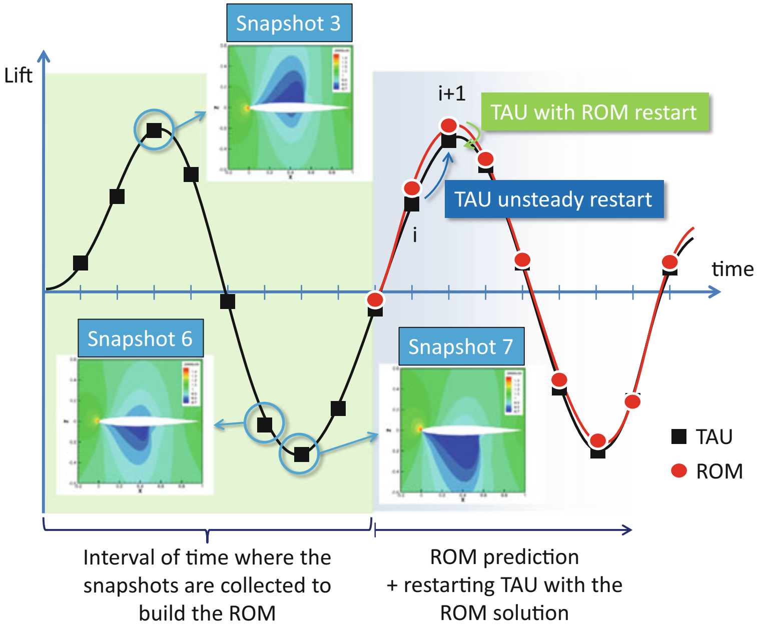

Snapshot 3
Lift
i+1
TAU with ROM restart
TAU unsteady restart
i
time
Snapshot 6
Snapshot 7
■ TAU
● ROM
Interval of time where the snapshots are collected to build the ROM
ROM prediction + restarting TAU with the ROM solution

Fig. 15.1 Procedure to initialize DLR'S CFD code TAU with ROM predicted solutions in an unsteady simulation (schematic). In this sketch the ROM is built by applying POD to the TAU snapshots collected during a previous time interval

In the present work, such strategy has been used; indeed, knowing the type of unsteady simulations to be performed (i.e. periodic pitching oscillations at different frequencies), it can be easier to devise a single maneuver covering a broad range of the parameter space of interest (e.g. with a pitching oscillations sweeping in frequency) instead of apply to each single unsteady simulation a moving-windows strategy.

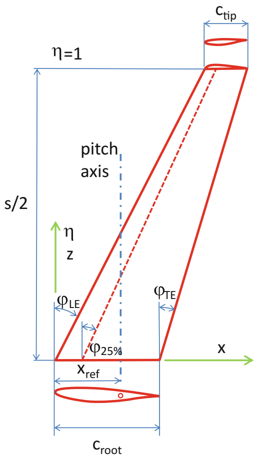
In any case, also for the moving window strategy it is generally recommended to employ an initial POD basis obtained from a generic training maneuver exciting different frequencies and amplitudes, and to then update the basis whenever a new high-fidelity CFD snapshot coming from the time-marching method is available.

Figure 15.1 sketches a restarting process where a ROM is built using snapshots of the flow field collected progressively during a time-marching simulation. In the standard TAU unsteady time marching procedure the solution at the $(i + 1)$th time step is initialized with the solution at the ith time step. Conversely, the ROM-based restarting procedure makes use of the ROM prediction at the $(i + 1)$th time step to initialize TAU at the same time step. Such an approximate ROM solution may be "closer" (depending on the level of unsteadiness of the simulation and on the size of the time steps used) to the final converged TAU solution at the $(i + 1)$th time step than that provided by the solution at the previous time step. Note that a ROM prediction is nothing else that a linear combination of POD modes derived from snapshots. Therefore, it is is only as good as the snapshots used to generate the ROM. It is not possible to correctly represent flow dynamics or physics not already contained in the snapshots dataset to which the POD is applied. Since the ROM predictions will depend on the training data used to generate the subspace spanned by the POD modes, the building of the ROM must be application driven, i.e., attention must be paid to the training maneuver, which must cover the desired parameter space (e.g. angle of attack range, amplitudes and frequencies) that the simulation will span. It is remarked here, however, that in this context the ROM is used as a pre-processing step in solving the full-order model, and it not intended as a substitute to the CFD solver TAU. Therefore, it must only provide for an approximate solution close enough to the solution to which the iterative procedure will converge, thus reducing the number of inner iterations and the computational cost.

15.2.3 Numerical Test Cases

The ROM-based restarting procedure is applied to the LANN wing, for which the geometrical data are shown in Table 15.1. The LANN wing is defined by two supercritical cross sections at the root and tip chord. Sections between root and tip are derived by linear interpolation. The model has a span of 1 meter and a root chord of 0.3608 m. The quarter-chord sweep angle is 25°. The taper ratio (c_{tip}/c_{root}) is 0.4, the aspect ratio is 7.92 and the airfoil thickness is about 12%. The twist between root and tip is 4.6°.

Table 15.1 Geometrical data of the LANN (Lockheed-Air Force-NASA-NLR) wing model [8]

Model Data		
 c_{tip} η=1 pitch axis s/2 η z $φ_{LE}$ $φ_{TE}$ $φ_{25\%}$ x x_{ref} c_{root}	half span $s/2$ c_{tip} c_{root} x_{ref} (pitch-axis) ϕ_{LE} ϕ_{TE} $\phi_{25\%}$ reference area	1.0 m 0.1444 m 0.3608 m 0.2240 m 27.493° 16.908° 25° 0.2526 m^2

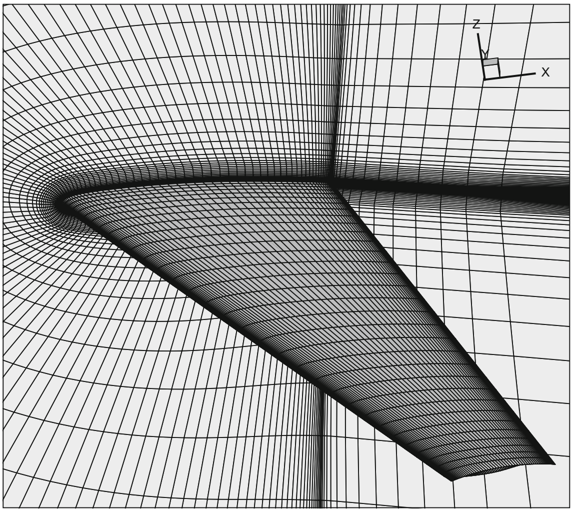


Fig. 15.2 Mesh of the LANN wing

A transonic viscous flight condition at a freestream airspeed of 271.66 m/s, a Mach number of 0.82, a Reynolds number based on c_{root} of $7.31 \cdot 10^6$ and 0.6° angle of attack is considered. At this condition, a periodic pithing oscillation about the unswept reference axis (located at 0.2240 m from the leading edge of the root airfoil) with 0.25° of amplitude and a reduced frequency based on c_{root} of 0.204 has been performed. It must be noted that the ROM-based restarting procedure does not require a periodic state. The procedure is valid for any kind of unsteady maneuver, periodic or not. The ROM can indeed also predict transitory responses.

The RANS equations with the Spalart–Allmaras (negative version) turbulence model have been used to model the flow with the TAU code. A structured mesh having 469, 213 grid points and 450, 560 elements is used for the computation, as showed in Fig. 15.2.

Since a periodic state can be achieved already at the second period, the simulation has been performed for 100 time steps, with 50 time steps per period (which have been shown to be sufficient for a good resolution with respect to time) using a dual time step method with a 2nd order backward difference discretization (BDF2). A study has been performed, where the convergence criterion has been set up to a minimum density residual of $1 \cdot 10^{-4}$, $1 \cdot 10^{-5}$ and $1 \cdot 10^{-6}$, with a maximum number of 100, 200 and 500 inner iterations to converge.

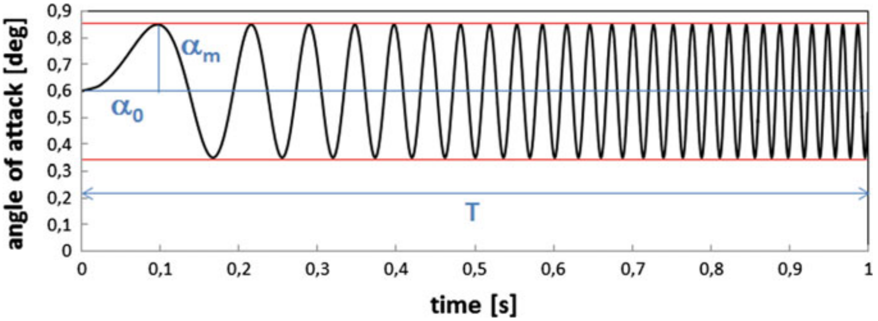


Fig. 15.3 Linear chirp signal used for the time-simulated pitching oscillation manoeuvre

A linear chirp maneuver defined as:

$$\alpha(t) = \alpha_0 + \alpha_m sin(\omega(t)t) \tag{15.12}$$

$$= \alpha_0 + \alpha_m sin\left(\frac{\omega_{max}}{T}t^2\right) \tag{15.13}$$

$$= \alpha_0 + \alpha_m sin\left(\frac{2\pi f_{max}}{T}t^2\right) \tag{15.14}$$

with $\alpha_m = 0.25°$ pitch amplitude and a maximum reduced frequency of $k_{max} = \omega_{max}c_{root}/V_\infty = 0.22$ ($f_{max} = 26.95\,\text{Hz}$), reached at time $t = T = 1\,\text{s}$, has been used as a training signal (see Fig. 15.3). The simulation has been performed using about 20 time steps per pseudo-period, linearly distributed with time over the total length of the simulation, for a total of 500 time steps. From this maneuver flow field snapshots have been collected and POD has been applied to them in order to get the POD modes. The modes have been used to build the ROM, which in turn has been used to predict, at each physical time step, an approximate flow solution for the periodic pitch oscillation. The approximate ROM prediction has successively been used to initialize the inner iterations of the CFD solver TAU.

15.2.4 Numerical Results

The time histories of the lift and moment coefficients of the LANN wing due to the periodic pitch oscillation are shown in Figs. 15.4 and 15.5, respectively. The results shown hereafter refer to the case study with convergence to a minimum density residual of $1 \cdot 10^{-4}$ and a maximum number 200 inner iterations. Both the approximate solution predicted by the ROM, which is used in the inner-loop restarting procedure as an initial guess, as well as the converged TAU solution are shown. The approximate ROM prediction is seen to be quite accurate (see e.g. the lift coefficient at time step 52 in the detailed view in Fig. 15.4), thus providing a better initial guess

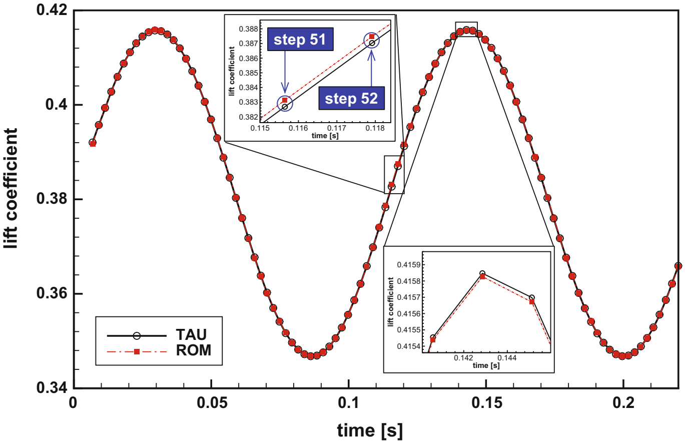


Fig. 15.4 Lift coefficient time history of the full-order model (TAU) and the ROM predictions used in the restarting procedure

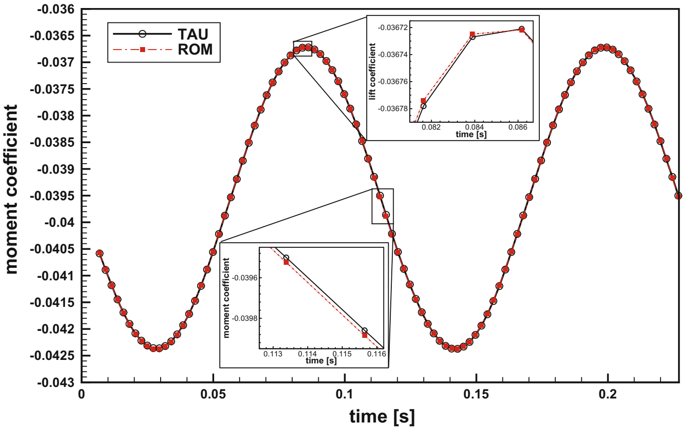


Fig. 15.5 Pitching moment coefficient (about the reference axis) time history of the full-order model (TAU) and the ROM predictions used in the restarting procedure

for TAU than by initializing the time step using the converged solution at the previous time step (see time step 51 in the detailed view in Fig. 15.4).

This is usually more emphasized when the solution presents large temporal gradients, as shown by the comparison of the nondimensional pressure between the approximate ROM solution and the CFD solution for time steps 51 and 52 (corresponding to the maximum time derivative of the solution). The pressure distribution of the wing's upper surface computed with TAU for time step 52 is shown in Fig. 15.6, the ROM prediction is shown in Fig. 15.7, while the TAU solution at the previous time step (51) is shown in Fig. 15.8. Figures 15.9 and 15.10 show the corresponding

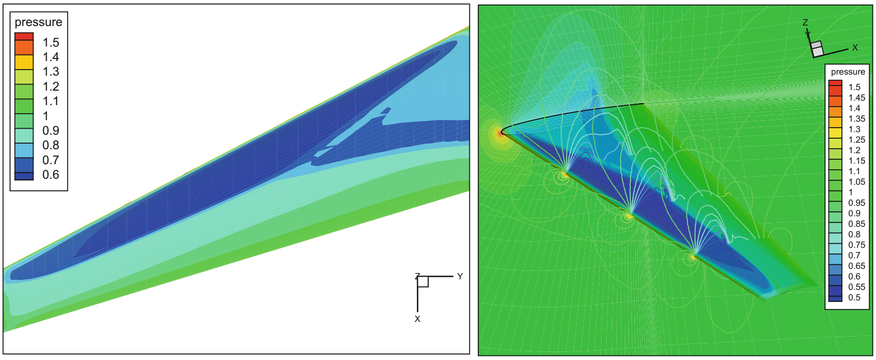


Fig. 15.6 CFD solution at time step 52

Fig. 15.7 ROM solution at time step 52

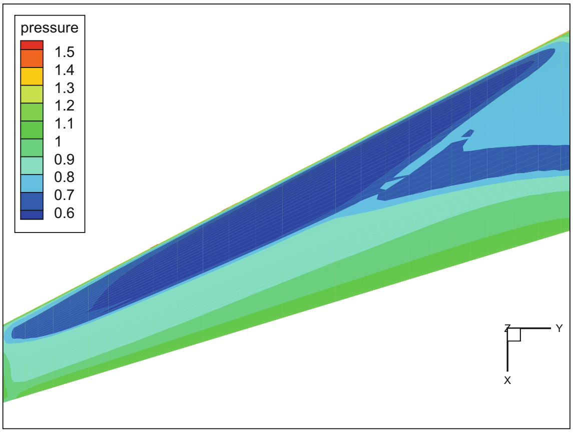
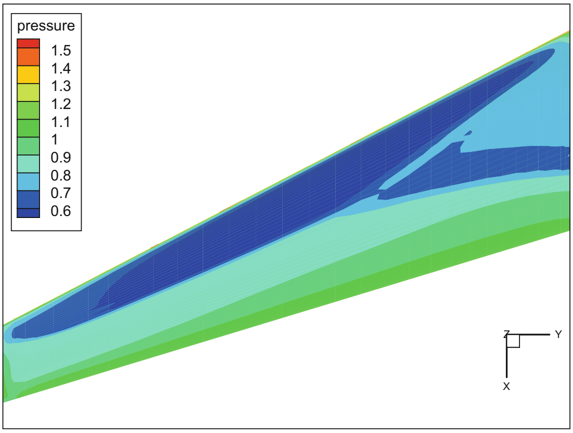


Fig. 15.8 CFD solution at time step 51



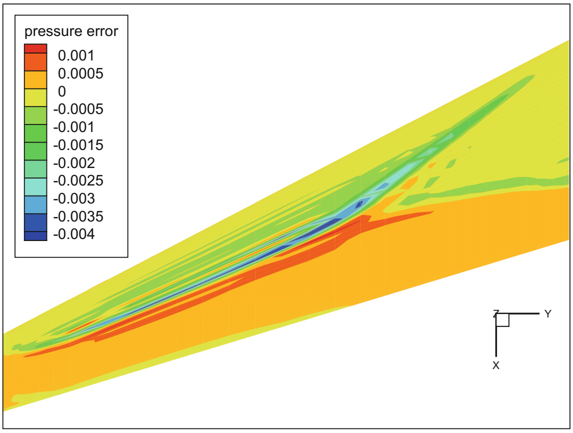


Fig. 15.9 Relative error between the nondimensionalized pressure of the CFD solution at time step 52 and the ROM prediction at time step 52

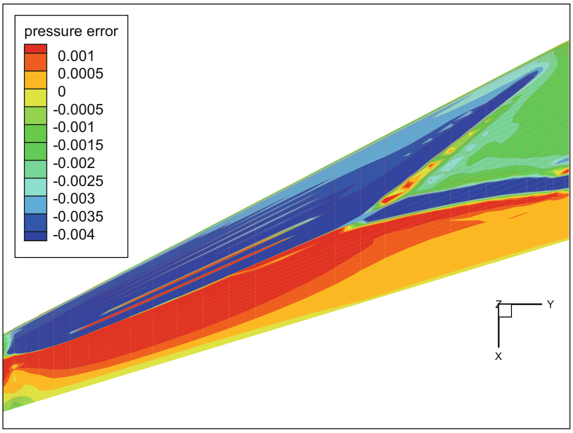


Fig. 15.10 Relative error between the nondimensionalized pressure of the CFD solution at time step 52 and the CFD solution at time step 51

signed relative difference $((p^{52}_{TAU} - p_{restart})/p^{52}_{TAU}$ between the TAU solution at time step 52 and the solution used for initializing TAU with $p_{restart} = p^{52}_{ROM}$ in Fig. 15.9 and $p_{restart} = p^{51}_{TAU}$ in Fig. 15.10. As expected, the difference between the TAU solutions at two consecutive time steps (51 and 52) is much larger than of the difference between the TAU solution and the ROM prediction at the same time step. Such differences are more relevant in the vicinity of the shock wave, which is moving back and forth in the chord-wise direction over the wing during the simulation.

A better initial guess provided by the approximate ROM solution is reflected in the computational cost of the simulation. In particular, using the ROM solution to restart the TAU computation at each time step decreases the number of residual calls (Fig. 15.11), the wall-clock time (Fig. 15.12) and the CPU time (Fig. 15.13) with respect to the standard unsteady restarting procedure.

It must be pointed out that number of residual calls for the TAU+ROM restarting procedure includes: the residual calls needed to build the ROM Jacobian by finite

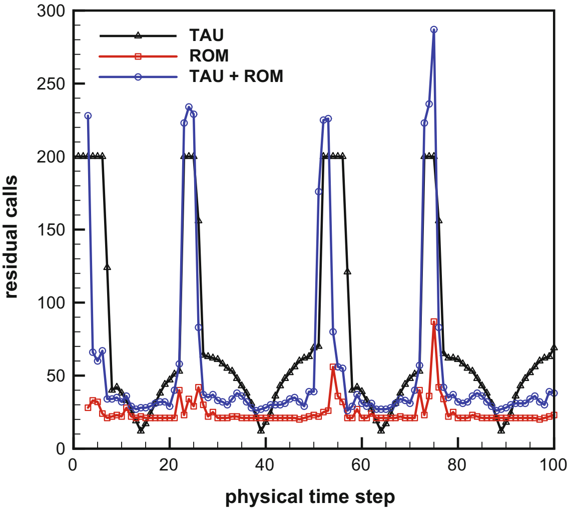


Fig. 15.11 Comparison of the number of residual calls needed to converge the solution at each physical time step by the standard TAU (black line) unsteady restarting procedure and when TAU is initialized with the ROM inner restart (blue line). The number of residual call needed to solve the nonlinear least squares problem for a ROM prediction are also shown (red line)

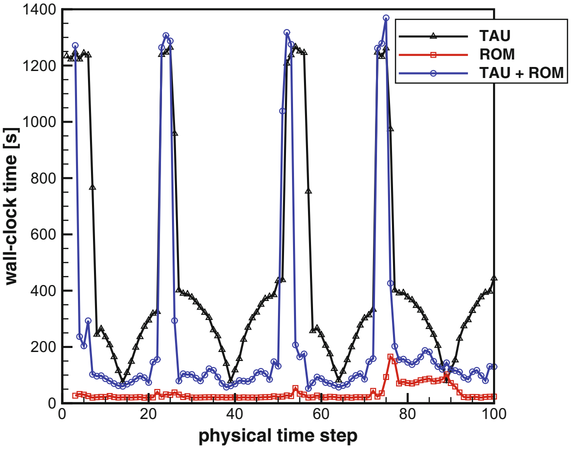


Fig. 15.12 Comparison of the wall-clock time needed at each physical time step by the standard TAU unsteady restarting procedure and by TAU with the ROM inner restart

differences (which is equal to the number of POD modes), the residual calls during the nonlinear least squares iterations to minimize the unsteady residual and the residual calls needed for the CFD solver (TAU) to converge to the solution. This is why the total number of residual calls for the TAU+ROM restarting procedure may exceed the maximum number of inner iterations, which has been fixed to 200 in the results shown here, see Fig. 15.11.

When comparing the number of residual calls (Fig. 15.11) and the computational time (Figs. 15.12 and 15.13) it must be noted that the nonlinear least-squares procedure used to compute the ROM predictions makes use of the so-called *residual-only solver* of TAU, whereas TAU makes use of the *flow solver*, which employs a multigrid procedure to converge the flow solution.

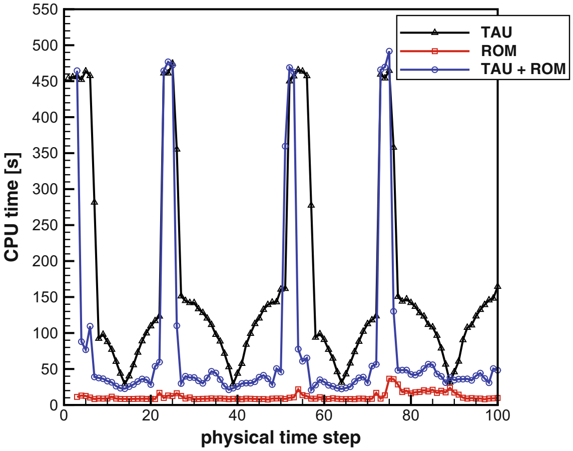


Fig. 15.13 Comparison of the CPU time needed at each physical time step by the standard TAU unsteady restarting procedure and by TAU with the ROM inner restart

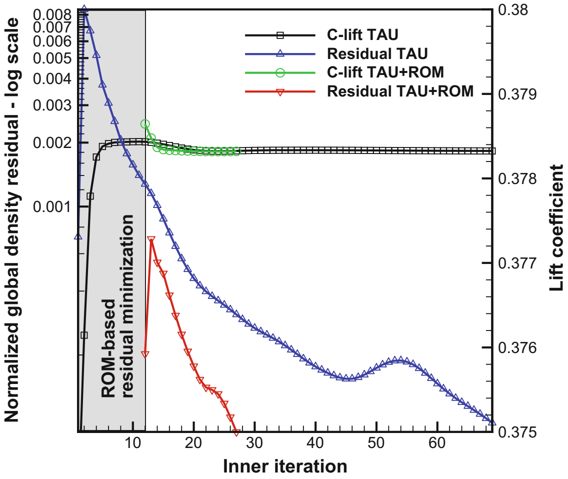


Fig. 15.14 Typical residual convergence history for a selected time step

Figure 15.14 compares a typical convergence history of the inner-loop residual (for a given physical time steps of the unsteady response) when the CFD solver (TAU) is initialized with the solution at the previous time step (the standard approach) and when it is initialized with the ROM predicted solution. In the latter case, the convergence history of the residual starts at the number of nonlinear least-squares iterations (in general about 10–15 iterations) needed to obtain the ROM solution. After every iteration of the TAU solver the norm of the residual is computed in order to monitor the convergence of the solution process. For each control volume a local residual vector is computed, composed of the residuals of the density, momentum and energy, and one or more components associated to the turbulence equations. The global density residual for the monitoring output, normalized with respect to the global residual of the steady state, is computed as the root mean square value:

$$\|res_\rho\| = \frac{\sqrt{\sum_{j=1}^{N} \frac{\mathrm{res}_{\rho,j}^2}{N}}}{\|\mathrm{res}_\rho\|_{\mathrm{steady}}} \tag{15.15}$$

with N being the number of grid points. Here, it is converged to a minimum value of $1 \cdot 10^{-4}$, with a maximum number of inner iterations of 200 per time step. The convergence history of the ROM is not shown because the Levenberg–Marquardt method employed to solve the nonlinear least-squares problem makes use of a different objective function and different stopping criteria to determine the convergence of the ROM solution than the TAU solver. From Fig. 15.14 it can be seen how the first TAU residual computed using the ROM predicted solution as an initial guess is lower than the first residual computed using the TAU solution at the previous time step.

The TAU solver shows some difficulty to converge for time steps with high angular velocity, i.e., the maximum temporal change of angle of attack and thus the maximum change in time of the flow field, as can be seen in Fig. 15.15 where the number of residual calls increases up to the point where the imposed maximum number of inner iterations (i.e. 200) is reached. In these cases, the global density residual shows an oscillating behaviour (Figs. 15.16 and 15.17) and starting with the better initial guess provided by the ROM may (Fig. 15.16) or may not (Fig. 15.17) have a beneficial effect, leading to a faster and oscillation-free convergence.

The convergence behaviour may be improved by choosing a different set-up of the TAU solver, e.g., reducing the CFL number or reducing the time-step size. The current setup has been chosen as a compromise between obtaining accurate results

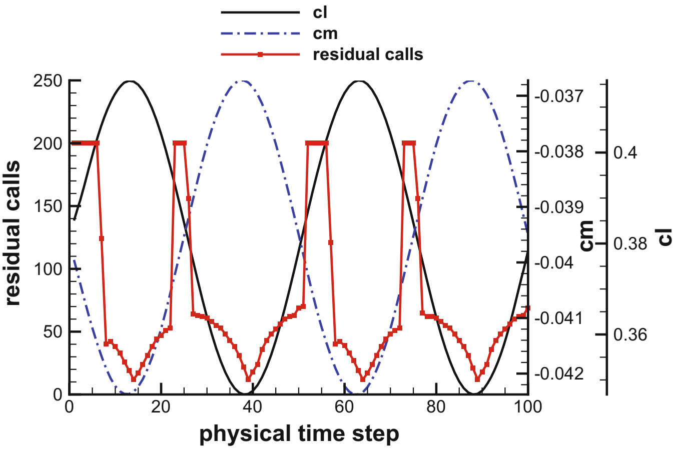


Fig. 15.15 Time history of the lift and moment coefficients and the number of residual calls (i.e. inner iterations) necessary to converge the global density residual to a value of $1e^{-4}$ for the standard unsteady restarting procedure of the TAU solver

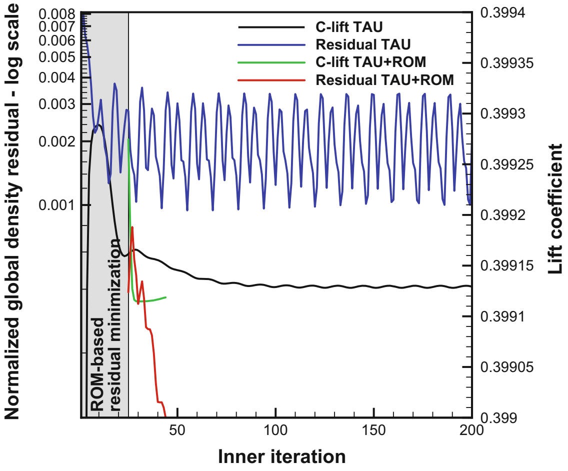


Fig. 15.16 Global density residual at a selected time step showing an oscillatory behavior for the standard TAU restarting procedure

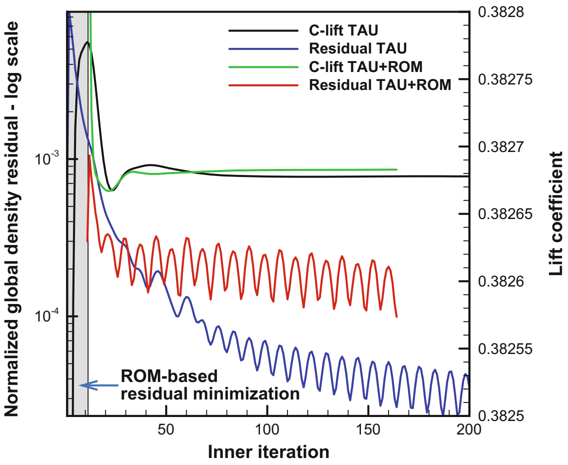


Fig. 15.17 Global density residual at a selected time step showing an oscillatory behavior for both the standard TAU and the ROM-based restarting procedure

in terms of aerodynamic loads (pressure distribution, lift and moment coefficient) and a reasonable computational time.

Table 15.2 summarizes the performance in terms of computational cost of the unsteady pitching oscillating simulations for different convergence criteria (i.e., minimum residual and maximum number of inner iterations). The *Performance* column shows the speed-up factor and the reduction in the number of TAU iterations (TAU calls). *Total calls* refers to how many residual calls were due to the ROM optimization process plus the following TAU iterations to convergence of the solution. *TAU calls* refers to the reduction in the number of TAU iterations to convergence (initialized with the standard approach or with a ROM predicted solution). The simulations have been performed in parallel using 10 processors.

Table 15.2 Performance of standard TAU and the ROM-based TAU restarting procedure

Convergence criteria		TAU with standard unsteady restart		TAU with ROM-based restart		Performance		
Min. residual	Max. inner iterations	WC time [s]	Residual calls	WC time [s]	Residual calls (TAU+ROM)	Speed-up factor	Total calls reduction	TAU calls reduction
$1e^{-4}$	100	10 h 53 min	5482	6 h 44 min	4671 (2272 + 2399)	1.62	1.17	2.41
$1e^{-5}$	100	17 h 27 min	9466	12 h 27 min	6757 (4677 + 2080)	1.40	1.40	2.02
$1e^{-6}$	100	18 h 26 min	10000	17 h 52 min	10000 (7936 + 2064)	1.03	1.00	1.26
$1e^{-4}$	200	12 h 52 min	7341	7 h 15 min	5464 (3052 + 2412)	1.77	1.34	2.41
$1e^{-5}$	200	25 h 50 min	14238	23 h 47 min	15309 (12905 + 2404)	1.09	0.93	1.10
$1e^{-4}$	500	22 h 49 min	12379	12 h 25 min	8556 (6222 + 2334)	1.84	1.45	1.99

It must be noted that these results do not include information of the offline cost in building the ROM, which must be also considered in the overall evaluation of the efficiency of the ROM-based restarting process. For the case analyzed here the offline cost is the combination of running a TAU simulation, collecting the snapshots and applying a proper orthogonal decomposition. The chirp maneuver used as training simulation was computed using 500 physical time steps, with a maximum of 100 inner iterations and a minimum residual of $1 \cdot 10^{-4}$. The wall-clock time of the simulation was 50h16 min. The proper orthogonal decomposition used to obtain the POD modes lasted for 7 min. The ROM was build using 17 POD modes.

As can be seen from Table 15.2, the number of residual calls of the ROM does not show significant variations with respect to changes in the converge criteria parameters (minimum residual tolerance and maximum inner iterations). Indeed, the ROM predicted solution ($\mathbf{w}_{rom}$) at a generic time step t_{n+1} depends on the TAU solutions ($\mathbf{w}_{TAU}$) at the actual and old time steps, t_n and t_{n-1}, respectively, only through the definition of the unsteady residual vector to minimize, as shown in the equation below:

$$\widehat{\mathbf{R}} \overset{def}{=} \mathbf{Res}(\mathbf{w}_{rom}(t_{n+1})) + \mathbf{\Omega}\frac{3\mathbf{w}_{rom}(t_{n+1}) - 4\mathbf{w}_{TAU}(t_n) + \mathbf{w}_{TAU}(t_{n-1})}{2\Delta t} \tag{15.16}$$

If the TAU solutions are properly converged for every minimum global density residual tolerance considered (i.e., $1 \cdot 10^{-4}$, $1 \cdot 10^{-5}$, and $1 \cdot 10^{-6}$) then the ROM predicted solution at the generic time step t_{n+1}, which is used to initialize TAU at that time step, is the same for all the convergence criteria considered. Only the number of inner iterations of the TAU solver to convergence changes for the selected tolerance. The more the tolerance is decreased the more iterations and computational time are required. Therefore the performance and effectiveness of the overall ROM-based restarting process are given mainly by two points: first by how accurate the ROM prediction is and second by how much the global density residual of the CFD solver is to be reduced, influencing thus the number of iterations. Requiring a high maximum residual (e.g. $1 \cdot 10^{-4}$) would give an advantage to the ROM-based restarting procedure over the standard one (assuming that the ROM prediction is quite accurate). This is because fewer iterations of the TAU solver will be necessary to converge. Decreasing the desired maximum residual will reduce the speed-up improvement because more TAU iterations (involving multigrid operations) will be required, thus nullifying the effect of the ROM in providing a good initial guess.

15.3 Conclusions and Remarks

An unsteady simulation of a LANN wing in transonic viscous flow has been performed. The motion analyzed is a periodic pitching oscillation which, for the selected amplitude and frequency parameters, does not show a nonlinear behavior in the lift and moment quantities, but it has an important nonlinear behavior of the flow field, characterized by a moving lambda-shaped shock wave, typical for swept wings under transonic on-flow conditions.

The CFD TAU unsteady simulation where the restarting process make use of the ROM predicted solution as initial guess for computing the flow field at a certain physical time step showed a speed-up, compared to running the CFD TAU solver directly and using the previous time step solution as restarting point.

Such speed-up may vary depending on the convergence criteria set-up for the minimum residual, but it is however not yet fully satisfactory.

It is nevertheless evident that improving the capability of the ROM prediction (e.g. using nonlinear manifold learning techniques, like Isomap) would accelerate the TAU inner-loop convergence in case of unsteady simulations, in particular when the required minimum global density residual is not very demanding.

Further speed-up of the ROM prediction step may be achieved by using hyper reduction techniques (e.g. gappy POD, missing point estimation, empirical interpolation), which make use of a subset of the computational mesh over which evaluate the data, or through sparsity-promoting techniques, which make use of a subset of the POD modes/snapshots having the greatest contribution on the quality of approximation.

An analysis of the ROM-based restarting approach focusing on the convergence of an output of interest (e.g. lift and moment) through the Cauchy convergence criteria would be helpful in understanding the real soundness of the method in an industrial context, where the main purpose of unsteady simulations is to get accurate aircraft loads.

References

1. T. Franz, R. Zimmermann, S. Görtz, N. Karcher, Interpolation-based reduced-order modelling for steady transonic flows via manifold learning, in special issue: reduced order modelling: the road towards real-time simulation of complex physics. Int. J. Comput. Fluid Dyn. **28**(3–4), 106–121 (2014)
2. P. A. LeGresley, J. J. Alonso, Investigation of non-linear projection for POD based reduced order models for aerodynamics, 39th AIAA aerospace sciences meeting and exhibit, AIAA Paper 2001-0926 (2001)
3. R. Zimmermann, S. Görtz, Non linear reduced order models for steady aerodynamics. Procedia Comput. Sci. **1**(1), 165–174 (2010). ICCS 2010
4. R. Zimmermann, S. Görtz, Improved extrapolation of steady turbulent aerodynamics using a non-linear POD-based reduced order model. Aeronaut. J. **116**(1184), 1079–1100 (2012)
5. M. Mifsud, R. Zimmermann, S. Görtz, Speeding-up the computation of high-lift aerodynamics using a physics-based reduced-order model. CEAS Aeronaut. J. **6**(1), 3–16 (2015)
6. D. Schwamborn, T. Gerhold, R. Heinrich, The DLR TAU-code: recent applications in research and industry, ECCOMAS CFD 2006 (2006)
7. J. Nocedal, S.J. Wright, *Numerical Optimization* (Springer, New York, 2006)
8. J.J. Horsten, R.G. den Boer, R.J. Zwaan, Unsteady transonic pressure measurements on a semispan wind tunnel model of a transport-type supercritical wing, Part I general description, Aerodynamic coefficients and vibration modes, AFWAL-TR-83-3039, Air Force Wright Aeronautical Laboratories (1982)