Applications of Proper Orthogonal Decomposition for Inviscid Transonic Aerodynamics

Bui-Thanh Tan, Karen Willcox and Murali Damodaran

***Abstract*— Two extensions to the proper orthogonal decomposition (POD) technique are considered for steady transonic aerodynamic applications. The first is to couple the POD approach with a cubic spline interpolation procedure in order to develop fast, low-order models that accurately capture the variation in parameters, such as the angle of attack or inflow Mach number. The second extension is a POD technique for the reconstruction of incomplete or inaccurate aerodynamic data. First, missing flow field data is constructed with an existing POD basis constructed from complete aerodynamic data. Second, a technique is used to develop a complete snapshots from an incomplete set of aerodynamic snapshots.**

I. Introduction

Model reduction using the Proper Orthogonal Decomposition (POD) has been widely applied to many physical problems [1]. Sirovich introduced the method of snapshots as a way to efficiently determine the POD basis vectors for large problems [2]. In particular, the method of snapshots has been widely applied to computational fluid dynamic (CFD) formulations to obtain reduced-order models for unsteady aerodynamic applications [3], [4], [5], [6]. A set of instantaneous flow solutions, or "snapshots" are obtained using the CFD method. The POD process then computes a set of basis functions from these snapshots, which is optimal in the sense that, for any given basis size, it minimizes the error between the original and reconstructed data. For linearized problems, efficient POD methods have also been developed that compute the snapshots in the frequency domain [7], [4], [8].

While use of POD to capture the time variation of fluid dynamic problems has been widespread, the development of reduced-order models to capture parametric variation is less common. The POD has been used to develop reduced-order models for turbomachinery flows with sampling in both time and over a range of interblade phase angles [9]. The resulting models were applied to flows at varying Mach numbers, although the snapshot ensemble is computed at a single Mach number condition. Accurate results were obtained for Mach numbers close to that used in the snapshots. The POD has also been used to develop models for optimization purposes [10]. In this case, the POD modes span a range of airfoil geometries. In another example that includes parametric variation, a fast computation has been developed that uses a POD basis to predict the steady-state temperature distribution of flow in a square cavity as the Rayleigh number is varied [11]. This method is a simple combination of the POD basis and an interpolation procedure.

Another application of the POD is for the repair of damaged data and construction of missing or "gappy" data as introduced in Everson and Sirovich [12] for the characterization of human faces. This idea could be extended to the prediction of aerodynamic flow fields. For example, a set of complete flow solutions may be available from a CFD calculation. One might wish to use these solutions as an information database in the reconstruction of partial data, such as that obtained from experimental measurements.

In this research, two extensions to the POD for aerodynamic applications will be considered. The first is to combine the POD approach with a cubic spline interpolation to capture parametric variations. The second application will apply the POD for the data reconstruction of transonic flows. In this paper, the basic POD method will first be outlined, followed by a description of the extensions with interpolation and incomplete data sets. A set of results for transonic flows will then be presented. The first example considers steady transonic flow past an airfoil with varying angle of attack and Mach number. The second example addresses a situation in which the complete flow pressure field is reconstructed from pressure values only on the airfoil surface. Finally, a problem is considered in which the POD snapshots are constructed from an incomplete set of aerodynamic data.

II. Proper Orthogonal Decomposition Theory and Extensions

A. Proper orthogonal decomposition

The basic POD procedure is summarized briefly here. The optimal POD basis vectors Φ are chosen to maximize the cost:[1]

$$\max_{\Psi} \frac{\left\langle \left| (U,\Psi)^2 \right| \right\rangle}{(\Psi,\Psi)} = \frac{\left\langle \left| (U,\Phi)^2 \right| \right\rangle}{(\Psi,\Phi)} \tag{1}$$

where (U,Φ) is the inner product of the basis vector Φ with the field $U(x,t)$, x represents the spatial coordinates, t is the time, and $<>$ is the time-averaging operation. It can be shown that the POD basis vectors are eigenfunctions of the kernel K given by

$$K(x,x') = \langle U(x,t), U^*(x,t) \rangle \tag{2}$$

The method of snapshots, introduced by Sirovich [2] is the way of finding the eigenfunctions Φ without explicitly calculating

Bui Thanh Tan is a Master of Engineering student at the Singapore-MIT Alliance (SMA), Singapore (email: smav56@nus.edu.sg).

Karen Willcox is an Assistant Professor with Aerospace Computational Design Laboratory Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, USA, and a SMA Fellow in High Performance Computation for Engineered System (HPCES) program.

Murali Damodaran is an Associate Professor with School of Mechanical and Production Engineering, Nanyang Technological University (NTU), Singapore, and a SMA Fellow in High Performance Computation for Engineered System (HPCES) program.

the kernel K. Consider an ensemble of instantaneous solutions, or "snapshots". It can be shown that the eigenfunctions of K are the linear combination of the snapshots as follows

$$\Phi = \sum_{i=1}^{m} \beta_i U^i \tag{3}$$

where U^i is the solution at a time t_i and the number of snapshots is large. For fluid dynamic applications, the vector U^i contains the flow unknowns at each point in the computational grid. The coefficients β_i can be shown to satisfy the eigenproblem

$$R\beta = \Lambda\beta \tag{4}$$

where R is known as the correlation matrix

$$R_{ik} = \frac{1}{m}\left(U^i, U^k\right) \tag{5}$$

The eigenvectors of R determine how to construct the POD basis vectors [using (3)], while the eigenvalues of R determine the importance of the basis vectors. The relative "energy" (measured by the 2-norm) captured by the i^{th} basis vector is given by $\lambda_i / \sum_{j=1}^{m} \lambda_j$. The approximate prediction of the field U is then given by a linear combination of the eigenfunctions

$$U \approx \sum_{i=1}^{p} \alpha_i \Phi^i \tag{6}$$

where $p << m$ is chosen to capture the desired level of energy, Φ^i is the i^{th} POD basis vector, and the POD coefficients α_i must be determined as a function of time.

B. POD with interpolation

The basic POD procedure outlined in the previous section considered time-varying flows by taking a series of flow solutions at different instants in time. The procedure could also be applied in parameter space, that is, obtaining flow snapshots while allowing a parameter to vary. Assume that the parameter of interest is denoted by δ. This could, for example, be the flow freestream Mach number or airfoil angle of attack.

A procedure for rapid prediction of the flow solution U at any value of δ is as follows:

1) Let $\left\{U^{\delta_i}\right\}_{i=1}^{m}$ be the set of snapshots corresponding to the set of parameter values $\{\delta_i\}_{i=1}^{m}$.
2) Perform the basic POD procedure described above on $\left\{U^{\delta_i}\right\}_{i=1}^{m}$ to get the POD basis $\left\{\Phi^k\right\}_{k=1}^{m}$.
3) The reconstruction of each snapshot is given by

$$U^{\delta_i} = \sum_{j=1}^{p} \alpha_j^{\delta_i} \Phi^j \tag{7}$$

where $p < m$ is the number of eigenfunctions used in the reconstruction. The POD coefficients $\alpha_j^{\delta_i}$ are given by

$$\alpha_j^{\delta_i} = \left(\Phi^j, U^{\delta_i}\right) \tag{8}$$

4) If $\left\{\alpha_j^{\delta_i}\right\}_{i=1}^{m}$ is a smooth function of δ, interpolation can be used to determine the POD coefficients for intermediate values of δ that were not included in the original ensemble. The prediction of U^δ at any value of δ via the POD expansion is given by (6)

$$U^\delta = \sum_{j=1}^{p} \alpha_j^\delta \Phi^j \tag{9}$$

where the coefficients α_j^δ are found by cubic spline interpolation of the set $\left\{\alpha_j^{\delta_i}\right\}_{i=1}^{m}$. Note that no discussion of a smoothness requirement was given by Hung [11]; however, this is important for the interpolation result to be reliable.

C. POD for reconstruction of missing data

Typically, the POD has been used as a way of performing data reduction on large systems, such as those encountered in CFD applications. However, here a procedure will be described in which the POD can be used to reconstruct an incomplete data set. This procedure is based on that developed by Everson and Sirovich [12] for the reconstruction of human face images.

The task is began by defining the "mask" vector [12] which describes for a particular flow vector where the data is available and where the data is missing. For the flow solution U^k, the corresponding mask vector n^k is defined as follows:

$$\begin{aligned} n_i^k &= 0 \text{ if } U_i^k \text{ is present} \\ n_i^k &= 1 \text{ if } U_i^k \text{ is missing or incorrect} \end{aligned}$$

where U_i^k denotes the i^{th} element of the vector U^k. For convenience in formulation and programming, zero values are assigned to the elements of the vector U^k where the data is missing, and the pointwise multiplication is defined as $\left(n^k, U^k\right)_i = n_i^k U_i^k$. Then the gappy inner product is defined as $(u,v)_n = ((n,u),(n,v))$, and the induced norm is $(\|v\|_n)^2 = (v,v)_n$.

Let $\left\{\Phi^i\right\}_{i=1}^{m}$ be the POD basis for the snapshot set $\left\{U^i\right\}_{i=1}^{m}$, where all snapshots are complete. Let g be another vector that has some elements missing, with corresponding mask vector n. Assume that there is a need to reconstruct the full or "repaired" vector from the incomplete vector g. Assuming, that the vector g represents a solution whose behavior can be characterized with the existing snapshot set, the expansion (6) can be used to represent the intermediate repaired vector $\tilde{g}$ in terms of p POD basis functions as follows:

$$\tilde{g} \approx \sum_{i=1}^{p} b_i \Phi^i \tag{10}$$

To compute the POD coefficients b_i, the error, E, between the original and repaired vectors must be minimized

$$E = \|g - \tilde{g}\|_n^2 \tag{11}$$

where E is defined using the gappy norm so that only the original existing data elements in g are compared. The coefficients b_i that minimize the error E can be found by differentiating (11) with respect to each of the b_i in turn. This leads to the linear system of equations

$$Mb = f \tag{12}$$

where $M_{ij} = \left(\Phi^i, \Phi^j\right)_n$ and $f_i = \left(g, \Phi^i\right)_n$. Solving equation (12) for b and using (10), the intermediate repaired vector $\tilde{g}$ can be obtained. Finally, the complete g is reconstructed by replacing the missing elements in g by the corresponding repaired elements in $\tilde{g}$, i.e. $g_i = \tilde{g}_i$ if $n_i = 0$.

D. POD with an incomplete snapshot set

Assume an existence of a collection of incomplete data $\left\{g^k\right\}_{k=1}^m$, with an associated set of masks $\left\{n^k\right\}_{k=1}^m$. The POD basis can be constructed using an iterative procedure. The first step is to fill in the missing elements of the snapshots using average values as follows:

$$h_i^k(0) = \begin{cases} g_i^k & \text{if } n_i^k = 1 \\ \bar{g}_i & \text{if } n_i^k = 0 \end{cases} \tag{13}$$

where $\bar{g}_i = \frac{1}{P_i} \sum_{k=1}^{m} g_i^k$, $P_i = \sum_{k=1}^{m} n_i^k$ and $h^k(l)$ denotes the l^{th} iterative guess for the vector h^k. A set of POD basis vectors can now be computed for this snapshot set, and iteratively used to refine the guess for the incomplete data. The procedure can be summarized as follows, beginning with $l = 0$:

1) Use the basic POD procedure on the snapshot set $\left\{h^k(l)\right\}_{k=1}^m$ to obtain the POD basis vectors for the current iteration, $\left\{\Phi^k(l)\right\}_{k=1}^m$.
2) Use these POD basis vectors to repair each member of the snapshot ensemble, as described in the previous section. The intermediate repaired data for the current iteration is given by

$$\tilde{h}^k(l) = \sum_{i=1}^{p} b_i^k(l) \Phi^i(l) \tag{14}$$

3) The values from these intermediate repaired data are now used to reconstruct the missing data for the next iteration as follows

$$h_i^k(l+1) = \begin{cases} h_i^k(l) & \text{if } n_i^k = 1 \\ \tilde{h}_i^k(l) & \text{if } n_i^k = 0 \end{cases} \tag{15}$$

4) Set $l = l + 1$ and repeat steps 1 through 3.

The above iterative procedure should be repeated until the maximum number of iterations is reached or until the estimate of the repaired data has converged (see Everson [12] for details).

III. Results and Discussion

Results will be presented for transonic flow past a NACA 0012 airfoil with a baseline freestream Mach number of 0.8. Both the interpolation and data reconstruction techniques will be applied. Snapshots were obtained from an inviscid steady-state CFD code, which uses a finite volume formulation as presented in Jameson et al. [13] and Damodaran [14].

A. POD with interpolation for parametric variation

The first problem considered is steady flow with varying angle of attack and Mach number. The Mach number interval [0.75, 0.85] is divided into 20 uniform intervals, and the angle of attack interval $[0°, 1.25°]$ is divided into 10 uniform intervals. Hence, the total number of snapshots in the ensemble is 231. Based on this snapshot set, interpolation will be used to predict the flow at any arbitrary Mach number and angle of attack within the range considered. For demonstration, POD will be applied to the pressure field only; the procedure for the other flow fields is straightforward.

The first prediction considered is for the pair $(\alpha = 0.45,\ M = 0.8)$, in which $M = 0.8$ is one of the values used to generate the snapshots but $\alpha = 0.45$ is not. Figure 1 shows the comparison between the predicted pressure (dash contour) and the exact pressure (solid contour) corresponding to $(\alpha = 0.45, M = 0.8)$, where "exact" refers to the solution from the CFD model. It can be seen that with five eigenfunctions, Figure 1(a), the error is large in places and two contours are far apart. However, as the number of POD basis vectors is increased to 25, Figure 1(c), the contours match closely. It should be noted that the cost for pressure prediction with 20, Figure 1(b), or 25 eigenfunctions does not differ greatly, since the method requires only interpolation of the scalar POD coefficients. The number of eigenfunctions can therefore be increased to obtain the desired level of accuracy.

Figure 2 shows the predictions of pressure coefficients on the upper and lower surfaces of the airfoil corresponding to the pressure field in Figure 1. One interesting point that can be observed here is that with a small number of eigenfunctions, the pressure distribution away from the airfoil shows a large error while the pressure distribution on the airfoil surface is close to the exact one. This can be seen clearly by comparing Figures 1(b) and Figure 2(b). Therefore, beyond a certain point, increasing the number of the eigenfunctions improves the farfield pressure prediction without having much effect on the surface pressure distribution.

Now, let us define the error as

$$error = max|(U - U_p)./U| = \|(U - U_p)./U\|_\infty \tag{16}$$

which is the the maximum percentage error between two corresponding elements of the CFD solution U and p^{th}-order reconstructed solution U_p, "./" means elementwise division. Figure 3(a) shows the maximum percentage error in log scale versus the number of POD modes used to predict the pressure distribution. It can be seen that the error decreases very quickly as number of eigenfunctions is increased from one to 25.

The second prediction considered is for the pair $(\alpha = 0.5,\ M = 0.812)$, in which $\alpha = 0.5$ is one of the values used to generate the snapshots but $M = 0.812$ is not. Figure 4 shows the comparison between the predicted pressure (dash contour) and the exact pressure (solid contour) corresponding to $(\alpha = 0.5, M = 0.812)$. Experience has shown that prediction is more sensitive with Mach number than with angle of attack, hence it is expected that more eigenfunctions will be needed in this example to get a good result. As shown in Figures 4(c) and 3(b), 30 eigenfunctions are needed to make the predicted flow field almost identical to the exact.

The last example in this section is the prediction for the pair $(\alpha = 0.45, M = 0.812)$ in which both $\alpha = 0.45$ and $M = 0.812$ are not values used to generate the snapshots.

Therefore, it is expected that a higher number of eigenfunctions are needed to obtain a good result. Figure 5 shows the comparison between the predicted pressure (dash contour) and the exact pressure (solid contour) for this case. As expected, Figure 3(c) shows that with 30 eigenfunctions the error is larger than that shown in Figure 3(b) with the same number of eigenfunctions. It can be seen in Figure 3(c) that at least 40 eigenfunctions are needed to get the same level of accuracy obtained in the previous two examples.

The above results show that the POD method combined with interpolation allows models to be derived that accurately predict steady-state pressure fields over a range of parameter values. The approach can be extended to the case where more than two parameters vary. For example, one might wish to include geometric properties of the airfoil in order to apply these models in an optimization context. While the number of snapshots in this case might be large, the method presented in this paper is straightforward to apply.

B. POD with interpolation for maximum lift coefficient

In this example, the goal is to find the angle of attack such that the cost

$$J = C_l \tag{17}$$

is maximized, where C_l is the lift coefficient of the airfoil. This problem can be solved by using a golden section search method together with POD. The NACA 0012 airfoil is considered at its baseline freestream Mach number of 0.8, and 51 snapshots are computed corresponding to uniformly spaced values of angle of attack in the interval $[-1.25°, 1.25°]$ with a step of $0.05°$. Since the airfoil is symmetric, the values of the angle of attack in a positive interval such as [0.00, 1.25] only need to be found to maximize the cost in (17). Fifteen eigenfunctions are used for this case. As the the flow model used is inviscid, it is expected that the lift coefficient is maximized when the angle of attack is 1.25 degrees.

Tolerance	0.1	0.01	0.001	0.0001
# iterations	6	11	15	20
AOA	1.193643	1.244918	1.249258	1.24993

Table 1: The tolerance and corresponding result by Golden Search method for Maximum lift coeffi cient problem.

Table 1 shows that the solution converges to the expected value 1.25 as the tolerance decreases. Hence, the minimized wave drag coefficient or maximum of the ratio of lift coefficient and wave drag coefficient problems can be treated similarly.

C. POD with interpolation for a tracking problem

For demonstration purposes, the quantity of interest tracked in this example is the Mach field V corresponding to $AOA = 0.77$, assuming that the information that the AOA is 0.77 and other other corresponding fields are not known. The statement for this problem is to find a value of AOA such that the Mach field from POD expansion $V_p = \sum_{i=1}^{p} \alpha_i \Phi_V^i$ is close to V, where $\left\{\Phi_V^i\right\}_{i=1}^m$ is the POD basis for Mach field. The formulation is then to find an AOA to minimize the cost functional

$$J = \|V - V_p\|^2 \tag{18}$$

Once again, Golden Section Search method together with PODI is used here to find the minimum cost in (18) with the angle of attack in the interval [-1.00, 1.25], and 15 eigenfunctions in previous section are used for PODI method.

Tolerance	0.1	0.01	0.001	0.0001
# iterations	7	12	17	21
AOA	0.781541	0.772394	0.769920	0.770023

Table 2: The tolerance and corresponding result by Golden Search method for Tracking problem.

Once again, the result converges to the expected $AOA = 0.77$ as the tolerance decreases. Once the AOA is found, other flow variable fields corresponding to the Mach field can be obtained immediately by applying the method in the section III-A. The application of Tracking problem for the other fields is straightforward.

D. Reconstruction of missing data in Aerodynamics

For purpose of this section, the incomplete data set will be artificially created by throwing away some portion of the data. In this problem, it is assumed that the complete set of POD eigenfunctions has already been computed using the data in Section III-A. The pressure field at $AOA = 0.77$ (which is not one of the snapshots) is generated by the CFD code. Next, only the airfoil surface values of this solution are kept, and treated as if they were obtained from some experiment. Figure 6(a) shows the points on the NACA 0012 airfoil surface where pressure field values are available. Figures 6(b) and (c) show the construction (dash contour) of pressure contour and the exact one (solid contour) with four and five POD eigenfunctions, respectively. As expected, the more eigenfunctions used, the closer to the exact pressure the reconstruction is. The exact pressure field can be obtained very accurately with only five POD eigenfunctions, showing that the POD methodology for data reconstruction works very effectively for aerodynamic applications.

E. Reconstruction of missing snapshots

In the final example, a 26-member snapshot ensemble is used. These snapshots correspond to steady pressure solutions at angles of attack in the range $[0°, 1.25°]$ uniformly spaced with an interval of $0.05°$. To create the incomplete snapshot set, 30% of the data of each snapshot is then discarded randomly. The algorithm described in Section II-D is then used to repair the data as follows. By first repairing the missing data points in each snapshot with the average over available data at that point, a new ensemble of data is created that has no missing values. With this new ensemble, a first approximation to the POD basis is then constructed. Then, each snapshot in the ensemble is repaired using the first approximation of the POD basis. This repaired ensemble is then used to construct a second approximation to the POD basis. This process is repeated until some stopping criterion is met. For this example, the stopping criterion is the limited number of iterations.

In Figure 7, the 30% incomplete second snapshot is repaired by the above procedure with six POD eigenfunctions. The exact pressure contour is the solid contour and the dash contour is the reconstruction. Figure 7(a) shows the original damaged snapshot. After 1 iteration, the repaired snapshot in Figure 7(b) is much better, but it is still far away from the original. Figure 7(c) is the repaired snapshot after 25 iterations and can be seen to match closely with the exact solution.

Figure 8 shows the repairing process for the 23^{rd} snapshot. Compared to the contour with 30% data missing in Figure 8(a), Figure 8(c) is almost identical to the exact one with only seven iterations. It can be seen that the convergence of the reconstruction process depends on the details of the particular snapshot under consideration. In particular, it depends on the structure of the flow snapshot and how much data is missing. For the 23^{rd} snapshot shown in Figure 8, the convergence rate is much faster than for the second snapshot shown in Figure 7.

Now the rate of convergence for the construction of the POD eigenfunctions is investigated. In Figure 9(a), it can be seen that construction of the first eigenfunction after two iterations (dash contour) is already very close to the exact contour (solid). However, in Figure 9(b), the construction of the second eigenfunction after ten iterations (dash-dot contour) is still far way from the exact one (solid contour). At least 31 iterations are needed to obtain the construction of the second eigenfunction (dash contour) close to the exact one. It is observed that the rate of construction of an eigenfunction that captures more energy is faster than that of an eigenfunction that captures less energy. The first eigenfunction, which captures 90.65% energy, requires only two iterations to converge very close to the exact one; however, the second eigenfunction, which captures 7.7%, requires 31 iterations. This observation is also be related to the fact that successive POD eigenfunctions often correspond to higher spatial frequency flow structures. For example, it can be seen in Figure 9 that the second eigenfunction contains more high-frequency shock structure than the first mode. It is therefore not surprising that reconstruction of this mode requires more iterations.

The rate of convergence of the eigenfunctions through the error is defined by

$$E(n) = \sqrt{\sum_{k=1}^{m} (\lambda_k(n) - \lambda_k(n-1))^2}$$

where λ_k is the eigenvalue corresponding to k^{th} eigenfunction, m is the number of eigenfunctions, and n is the iteration number [12]. The rate of convergence of the error $E(n)$ in log scale is given in Figure 9(c). We can see that the error goes down very fast when the number of iterations is less than ten but it decreases slowly after that. It is also observed that the rate of convergence depends on the random generator which means the missing structure of the snapshots. Generally, the more data is missing, the more iterations we need to obtain the convergent reconstructions of snapshots.

Note that the procedure itself is quite expensive since we have to solve m (number of snapshots) systems (12) for constructing m snapshots (the more eigenfunctions we use the more systems we have to solve) for each iteration. Furthermore, an eigenvalue problem with size m has to solved to find the eigenfunctions for each iteration.

IV. Conclusion

The POD basis has been shown to be efficient for capturing relevant flow information for steady transonic aerodynamic applications. By coupling the POD basis with an interpolation method, models are obtained that give accurate flow field predictions. These predictions do not require a projection onto the CFD governing equations, but rather just a collection of flow snapshots that covers the parameter ranges of interest. The interpolation approach is applicable to any problem whose properties of interest are a smooth function of the parameters under consideration. The POD has also been shown to be very effective for reconstructing flow fields from incomplete data sets. While the rate of convergence of the reconstruction depends on the amount of missing data and the structure of the flow field, the method was found to work effectively for all problems considered.

References

[1] P. Holmes, J.L. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, Cambridge, UK, 1996.

[2] L. Sirovich, "Turbulence and the Dynamics of Coherent Structures. Part 1 : Coherent Structures," *Quarterly of Applied Mathematics*, vol. 45, no. 3, pp. 561–571, October 1987.

[3] E.H. Dowell, K.C. Hall, J.P. Thomas, R. Florea, B.I. Epureanu, and J. Heeg, "Reduced Order Models in Unsteady Aerodynamics," AIAA Paper 99-1261, 1999.

[4] K. C. Hall, J. P. Thomas, and E. H. Dowell, "Reduced-Order Modeling of Unsteady Small-Disturbance Flows Using a Frequency-Domain Proper Orthogonal Decomposition Technique," AIAA Paper 99-0655, 1999.

[5] M.C. Romanowski, "Reduced Order Unsteady Aerodynamic and Aeroelastic Models using Karhunen-Loéve Eigenmodes," AIAA Paper 96-194, 1996.

[6] P. Beran and W. Silva, "Reduced-Order Modeling: New Approaches for Computational Physics," AIAA Paper 2001-0853, 2001.

[7] T. Kim, "Frequency-Domain Karhunen-Loeve Method and Its Application to Linear Dynamic Systems," *AIAA Journal*, vol. 36, no. 11, pp. 2117–2123, 1998.

[8] K.E. Willcox, J.D. Paduano, J. Peraire, and K.C. Hall, "Low Order Aerodynamic Models for Aeroelastic Control of Turbomachines," AIAA Paper 99-1467, 1999.

[9] E. H. B. I. Epureanu, Dowell and K.C. Hall, ""A Parametric Analysis of Reduced Order Models of Potential Flows in Turbomachinery Using Proper Orthogonal Decomposition," 2001-GT-0434, Proceedings of ASME TURBO EXPO 2001, New Orleans, Louisiana, June 2001.

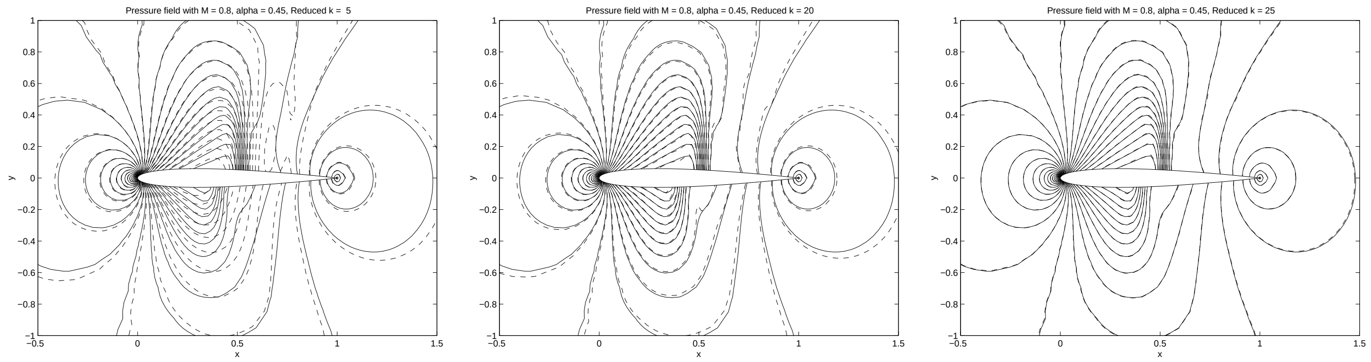
[10] P.A. LeGresley and J.J. Alonso, "Investigation of Non-Linear Projection for POD Based Reduced Order Models for Aerodynamics," AIAA Paper 2001-0926, presented at 39th Aerospace Sciences Meeting and Exhibit, Reno, NV, 2001.

[11] Hung V. Ly and Hien T. Tran, "Modeling and Control of Physical Processes using Proper Orthorgonal Decomposition," *Journal of Mathematical and Computer Modeling*.

[12] R. Everson and L. Sirovich, "The Karhunen-Loeve for Gappy Data," *The Rockefeller University*.

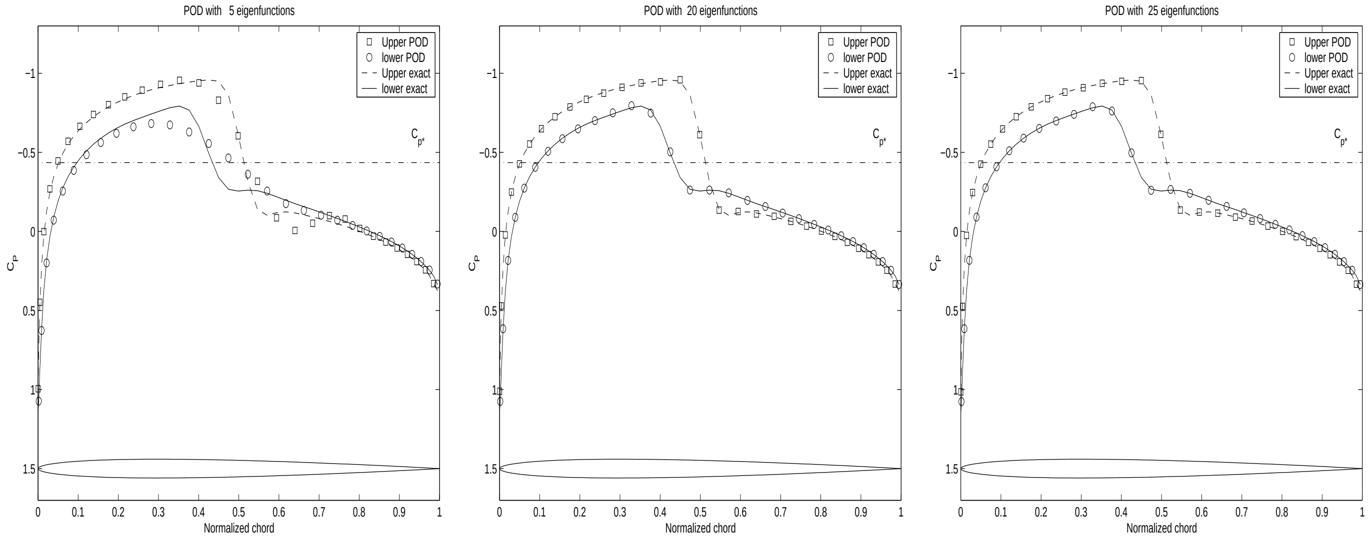
[13] Schmidt W. Jameson, A. and E. Turkel, "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, 1981.

[14] M. Damodaran, "Finite Volume Computation of Unsteady Inviscid Rotational Transonic Flows Past Airfoils in Rigid Body Motion," AIAA Paper 88-0006, 1988.



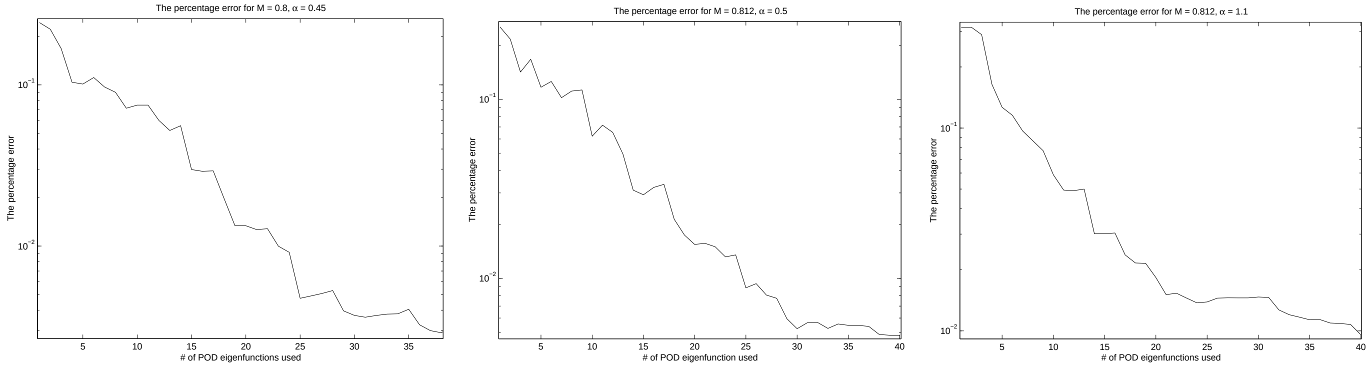
(a) (b) (c)

Fig. 1. Comparison of predicted pressure contours (dash) and exact pressure contours (solid) for a Mach number of 0.8 and angle of attack of $0.45°$; (a) five POD eigenfunctions, (b) twenty POD eigenfunctions, (c) 25 POD eigenfunctions.



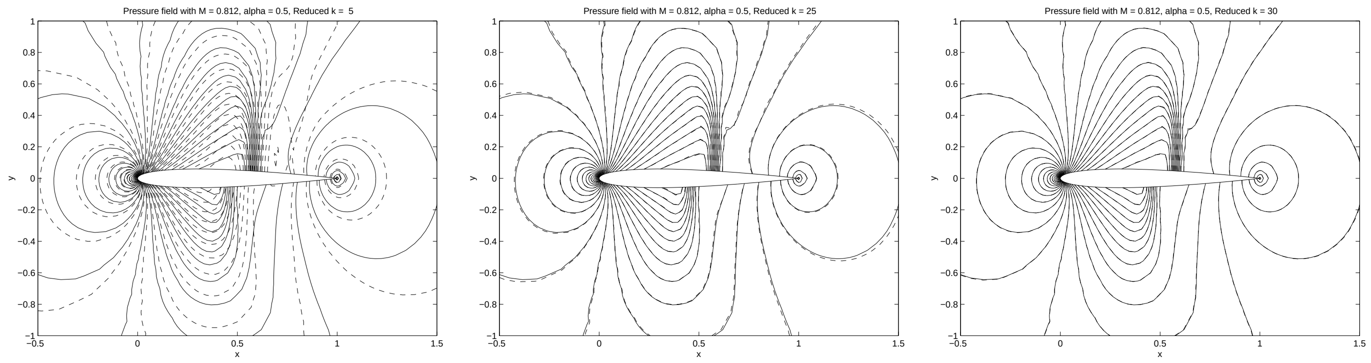
(a) (b) (c)

Fig. 2. Comparison of predicted upper pressure coefficients (square), predicted lower pressure coefficients (circle), exact upper pressure coefficients (dash) and exact lower pressure coefficients (solid) for a Mach number of 0.8 and angle of attack of $0.45°$. C_{p*} is the sonic pressure coefficient; (a) five POD eigenfunctions, (b) twenty POD eigenfunctions, (c) 25 POD eigenfunctions.



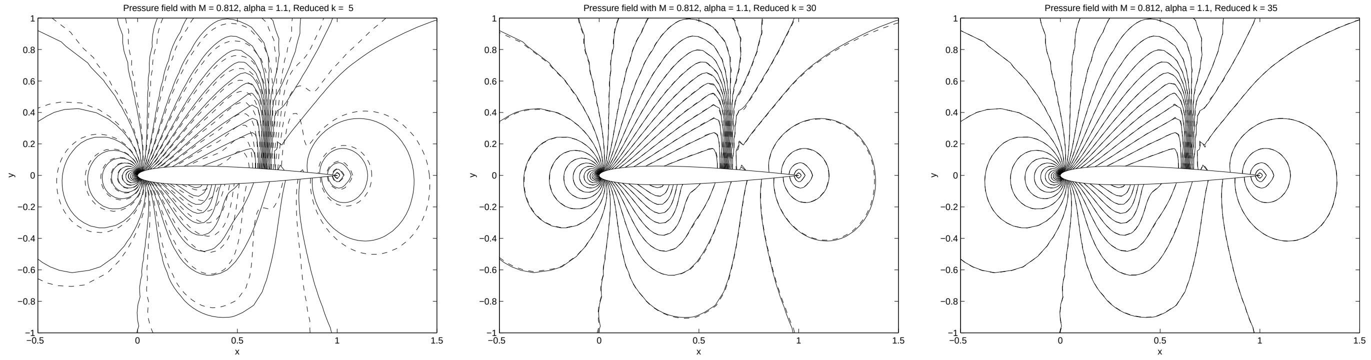
(a) (b) (c)

Fig. 3. Variation of percentage error versus the number of POD eigenfunctions in log scale; (a) M = 0.8 and $AOA = 0.45°$, (b) M = 0.812 and $AOA = 0.5°$, (c) M = 0.812 and $AOA = 1.1°$.



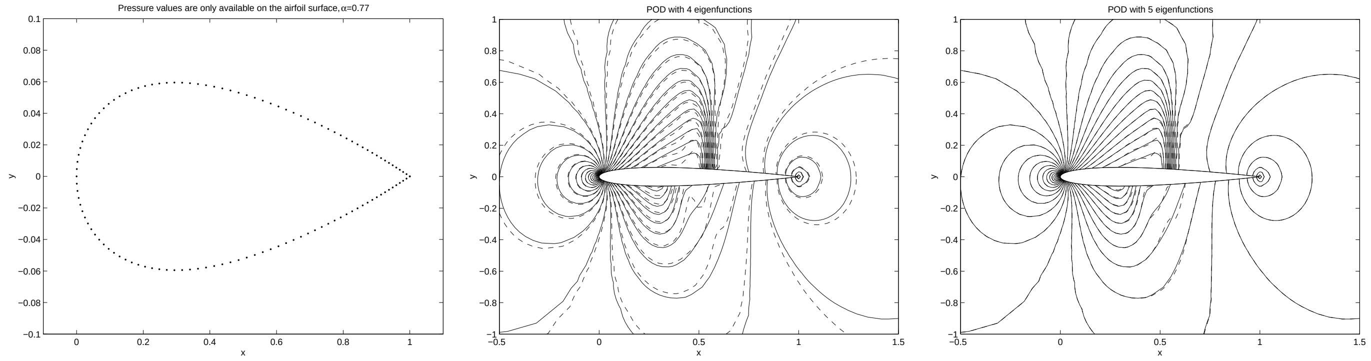
(a) (b) (c)

Fig. 4. Comparison of predicted pressure contours (dash) and exact pressure contours (solid) for a Mach number of 0.812 and angle of attack of $0.5°$; (a) five POD eigenfunctions, (b) 25 POD eigenfunctions, (c) thirty POD eigenfunctions.



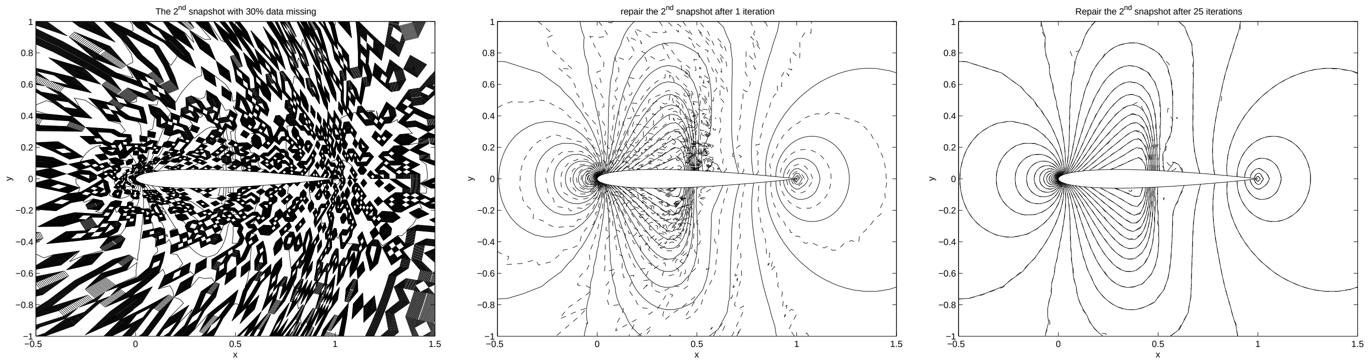
(a) (b) (c)

Fig. 5. Comparison of predicted pressure contours (dash) and exact pressure contours (solid) for a Mach number of 0.812 and angle of attack of $1.1°$; (a) five POD eigenfunctions, (b) thirty POD eigenfunctions, (c) 35 POD eigenfunctions.



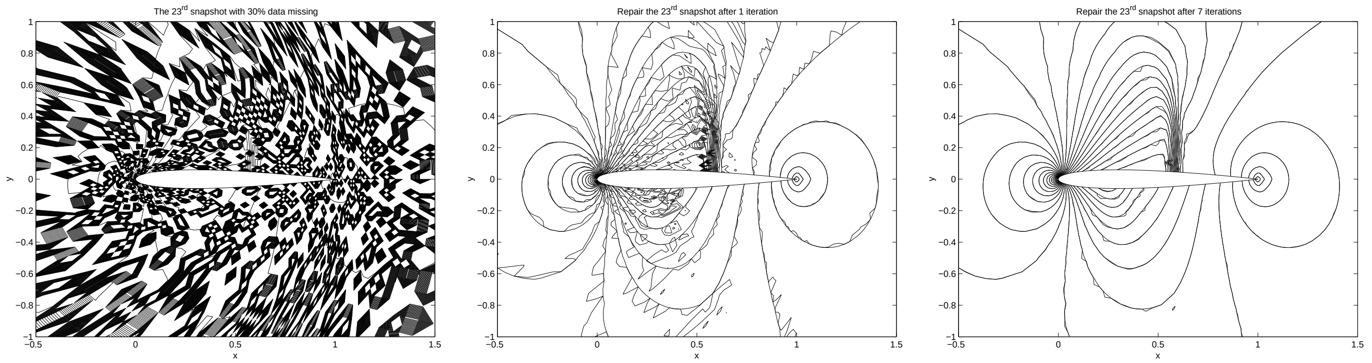
(a) (b) (c)

Fig. 6. The reconstruction of the pressure field from airfoil surface pressure distribution (dash) and the exact one (solid). (a) The points on the airfoil where pressure values are available, (b) the construction with four POD eigenfunctions, (c) the reconstruction with five POD eigenfunctions.



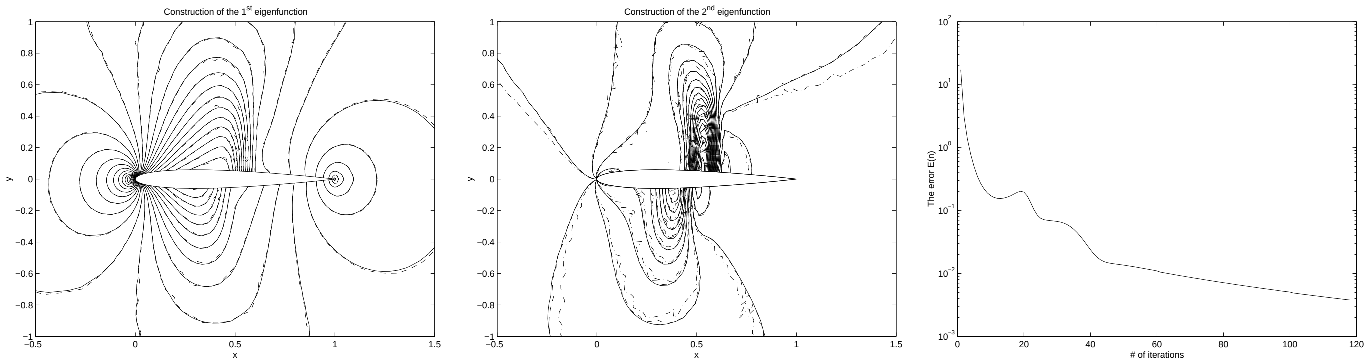
(a) (b) (c)

Fig. 7. The reconstruction of the 2^{nd} snapshot (dash), and the exact one (solid). (a) the 2^{nd} snapshot with 30% data missing (b) The reconstruction of 2^{nd} snapshot after one iteration (dash), (c) The reconstruction of 2^{nd} snapshot after 25 iterations (dash).



(a) (b) (c)

Fig. 8. The reconstruction of the 23^{rd} snapshot (dash), and the exact one (solid). (a) the 23^{rd} snapshot with 30% data missing (b) The reconstruction of 23^{rd} snapshot after one iteration (dash), (c) The reconstruction of 2^{nd} snapshot after seven iterations (dash).



(a) (b) (c)

Fig. 9. (a) The construction of the 1^{st} eigenfunction after two iterations (dash) and the exact one (solid), (b) The constructions of 2^{nd} eigenfunction after ten iterations (dash) and the exact one (solid), (c) The error $E(n)$.