EVOLUTIONARY AND DETERMINISTIC METHODS FOR DESIGN, OPTIMIZATION AND CONTROL
C. Poloni, D. Quagliarella, J. Périaux, N. Gauger and K. Giannakoglou (Eds.)

EFFICIENT POD-BASED MODEL ORDER REDUCTION FOR STEADY AERODYNAMIC APPLICATIONS

Alexander Vendl
Institut Computational Mathematics
AG Numerik
TU Braunschweig
Pockelstr. 14, 38106 Braunschweig, Germany
Email: a.vendl@tu-braunschweig.de

Heike Faßbender
Institut Computational Mathematics
AG Numerik
TU Braunschweig
Pockelstr. 14, 38106 Braunschweig, Germany
Email: a.vendl@tu-braunschweig.de

Abstract. In aerodynamic applications many model reduction methods use Proper Orthogonal Decomposition (POD). In this work a POD-based method, called Missing Point Estimation (MPE), will be applied to steady-state flows with variation of the angle of attack. The idea of MPE is to select a subset of the computational grid points (control volumes) and limit the governing equations to these. Subsequently, the remaining equations are projected onto the POD subspace. This approach has the advantage that the nonlinear right hand side of the governing equations has to be evaluated only for some selected points (control volumes).

Key words: Model Order Reduction, Proper Orthogonal Decomposition, Missing Point Estimation, CFD

1 Introduction

In fluid dynamics Proper Orthogonal Decomposition and Galerkin Projection are often used to obtain reduced order models. In many publications this is achieved by formulating the inner products that appear in the Galerkin projected system in terms of the POD coefficients; confer Rowley et al[1] for example. Since the dimension of the POD subspace (number of POD coefficients) is typically very small compared to the number of grid points, the inner products can then be efficiently evaluated.

More recently, an approach called the subspace projection method[2,3] has been proposed, which projects the full order model onto the POD subspace. This has the advantage that no elaborate modelling has to be done to obtain the reduced order model. However, since the projected full order residual is part of the reduced model, there is a dependence on the degrees of freedom of the original model.

In order to alleviate this shortcoming, we will present an alternative approach which aims at approximately evaluating the projected residual in a sparse way, that is, evaluating it at only a subset of all computational grid points. This idea is based on the Missing Point Estimation (MPE) method, which has been introduced by Astrid[4]. Since its first publication MPE has been used in many different applications such as heat transfer processes[5], electrical circuit modeling[6], and oil reservoir simulation[7]. The authors are not aware of any previous applications of MPE to

aerodynamics.

2 Reduced order modelling

In this work we will consider the two-dimensional Euler equations to be the governing equations of the original model, which after discretization in space with a finite volume scheme can be written as

$$\frac{d}{dt}\vec{w}(t;\alpha) = -\Omega^{-1}\vec{R}(\vec{w}(t;\alpha)), \tag{1}$$

where $\vec{w}$ is the vector of conservative variables, Ω is a diagonal scaling matrix with the volumes on the diagonal such that each volume appears once for each conservative variable and $\vec{R}$ is the residual vector constituting the spatial discretization of the fluxes through the faces of the control volumes. The solution of the equations depend on the angle of attack α, which is a parameter to the model and is defined as the angle between the oncoming flow and the body reference line. The four conservative variables are density, the momentum densities in both spatial directions, and total energy density. We will assume that the computational grid consists of n points. Hence, the vector of conservative variables and thus also the discretized system (1) is of size $4n$ for a two-dimensional geometry. In this work our goal is to obtain a system of reduced size, which accurately captures the dynamics of the original model. To achieve this Proper Orthogonal Decomposition is used to construct a suitable basis. It will be outlined next.

2.1 Proper Orthogonal Decomposition

Proper Orthogonal Decomposition is a technique, which yields a basis for the solution space of a model if a set of solutions - called snapshots - are given. In our work the snapshots are steady-state solutions $\vec{w}_i$ to the Euler equations (1) for different angles of attack, that is $\vec{w}_i = \vec{w}(\alpha_i)$. They are computed with the finite volume CFD solver TAU[8], which uses a cell-vertex scheme with dual control volumes[9].

After computing the average $\vec{\bar{w}} = \frac{1}{m}\sum_{i=1}^{m}\vec{w}_i$ of the m given snapshots, the so-called snapshot matrix is constructed as

$$Y = \left[\, (\vec{w}_1 - \vec{\bar{w}}) \;\cdots\; (\vec{w}_m - \vec{\bar{w}}) \,\right]. \tag{2}$$

We will consider the POD in the discretized L_2 space with its inner product given by $(\vec{w}_i, \vec{w}_j)_\Omega = \vec{w}_i^T \Omega \vec{w}_j$, where Ω is defined as before. The idea of POD is to find a set of basis vectors u_i, which optimally describe the snapshot matrix Y. This leads to the maximization problem[10]

$$\begin{aligned} &\max_{u_1,\ldots,u_d \in \mathbb{R}^{4n}} \sum_{i=0}^{d}\sum_{j=0}^{m} |(w_j, u_i)_\Omega|^2 \\ &\text{s.t.} \quad (u_k, u_l) = \delta_{kl} \quad \text{for} \quad 1 \le k, l \le d, \end{aligned} \tag{3}$$

where d is the number of basis vectors used in the reduced order model.

Its solution is related to the Singular Value Decomposition (SVD) of the weighted snapshot matrix $\bar{Y} = \Omega^{1/2}Y \in \mathbb{R}^{4n\times m}$. The SVD is given by $\bar{Y} = \bar{U}\Sigma\bar{V}$, where $\bar{U} = [\bar{u}_1 \ldots \bar{u}_{4n}] \in \mathbb{R}^{4n\times 4n}$ and $\bar{V} = [\bar{v}_1 \ldots \bar{v}_m] \in \mathbb{R}^{m\times m}$ are orthogonal matrices,

which consist of the left and right singular vectors, respectively, and where $\Sigma = \operatorname{diag}\left(\sigma_1, \ldots, \sigma_{\min\{4n,m\}}\right)$ is a diagonal matrix with the singular values as entries, ordered such that $\sigma_1 \geq \ldots \geq \sigma_{\min\{4n,m\}}$. The maximization problem (3) is solved by the vectors $u_i = \Omega^{1/2}\bar{u}_i$ with $i = 1, \ldots, d$, which form the *POD basis*. This follows either after some technical computations from the Schmidt-Eckart-Young-Mirsky Theorem or by solving (3) with the help of Lagrangian multipliers as done in Theorem 1.8 in Volkwein[10].

It will be assumed that the solutions $\vec{w}^*$, reside in the affine subspace spanned by the POD basis, that is, they can be written as

$$\vec{w}^* \approx U\vec{a} + \vec{\bar{w}}, \tag{4}$$

where $\vec{a}$ is a vector of suitable coefficients for the POD basis $U = [u_1 \ldots u_d]$ of rank d.

2.2 Missing Point Estimation

After having established a basis for the solution space with the help of POD, we are able to set up a reduced order model. In this work the Missing Point Estimation[5] is pursued, whose idea is to evaluate the right hand side of the governing equations only at a subset of computational grid points. Recasted in the CFD context this means that the residual vector is computed only for the faces of those (dual) control volumes, which are associated with the selected points. This has the obvious advantage that the residual evaluations become less costly.

In order to formulate the idea of selecting only few points, we will define the selection matrix. To this end, assume that we want to choose the points $\mathbb{X} = \{j_1, \ldots, j_{\tilde{n}}\} \subset \{1, \ldots, n\}$, where $\tilde{n}$ is the number of the selected points. The selection matrix is then given by $\tilde{P} = [e_{j_1} \cdots e_{j_{\tilde{n}}}] \in \mathbb{R}^{n \times \tilde{n}}$, with the jth unit vector $e_j \in \mathbb{R}^n$. Since we will need the point selection for each variable, we define $P = \operatorname{diag}(\tilde{P}, \tilde{P}, \tilde{P}, \tilde{P}) \in \mathbb{R}^{4n \times 4\tilde{n}}$.

With the help of the selection matrix a projection onto the selected points can be defined as $\Pi_P = PP^T$. After inserting the POD representation (4) into the governing equations (1), which introduces an error $\vec{\epsilon}$, we apply the projection. This yields

$$PP^T U \frac{d}{dt}\vec{a}(t;\alpha) = -PP^T\Omega^{-1}\vec{R}(U\vec{a}(t;\alpha) + \vec{\bar{w}}) + PP^T\vec{\epsilon}.$$

Note that the term $\frac{d}{dt}\vec{\bar{w}}$ on the left hand side is dropped, since it is zero due to the fact that the average is independent of time. The constructed system is highly overdetermined. Therefore the number of equations is reduced by imposing orthogonality conditions upon the system, which force it to be orthogonal to the POD subspace in L_2 sense, that is

$$\left(u_i, PP^T\left[\frac{d}{dt}U\vec{a}(t;\alpha) + \Omega^{-1}\vec{R}(U\vec{a}(t;\alpha) + \vec{\bar{w}})\right]\right)_\Omega = 0$$

with $i = 1, \ldots d$. Recasting the above equations back into matrix representation yields

$$(U^T PP^T \Omega U)\frac{d}{dt}\vec{a}(t;\alpha) = -U^T PP^T \vec{R}(U\vec{a}(t;\alpha) + \vec{\bar{w}}).$$

As a last step we eliminate the matrix in front of the time derivative on the left hand side by multiplying with its inverse and obtain the MPE system

$$\frac{d}{dt}\vec{a}(t;\alpha) = -(U^T P P^T \Omega U)^{-1} U^T P P^T \vec{R}(U\vec{a}(t;\alpha) + \vec{w}). \tag{5}$$

2.2.1 The projection of the Missing Point Estimation

The derivation of the Missing Point Estimation can also be recasted into a projection framework. Consider the projection

$$\Pi = U(U^T P P^T \Omega U)^{-1} U^T P P^T \Omega. \tag{6}$$

Obviouly, if Π is applied to (1), then the reduced system of the MPE (5) is obtained. In general, Π will not be orthogonal, but oblique, and is a Petrov-Galerkin projection.

Note that the projection of Missing Point Estimation can be seen as a catenation of two different projections: The first one projects onto the selected points, while the second one projects onto the POD subspace.

2.2.2 Point Selection

Since up to now it has not been outlined how to select the points, this issue shall be addressed now.

It has been observed that if the points are chosen such that they are spatially clustered, the information obtained at these points is limited. For this reason, we choose those points, which are contained in three rings around the airfoil: the first ring is close to the surface, the second further away and finally the last one contains the points on the farfield.

Table 1 defines the three rings by specifying the minimal and maximal distance from the surface. The distances are measured with the help of the reference length of the model. For example for airfoils, which are considered in this work, the reference length is determined by the length of the airfoil, which is set to be one.

ring	1	2	3
minimal distance from surface	0.2	2.0	99.0
maximal distance from surface	1.0	9.0	101.0

Table 1: Definition of the rings for choosing $\mathbb{X}$.

Since quite a lot of points are selected in this way, we investigate another point selection strategy found in the literature. Following the approach of Willcox[11] and Astrid et al[5] another option to choose the points is based on minimizing the condition number of the matrix $M = (U^T P P^T \Omega U)$. This can be interpreted as demanding the projection of the basis to be as orthogonal as possible in the L_2 sense for the chosen points, considering that the condition number of a matrix is one if and only if it is the multiple of an orthogonal matrix.

Note that selecting the points is a combinatorial problem and hence very costly. For this reason we preselect an index set of points $\mathbb{X}$, which seem to be good candidates for selection. In fact, we will choose those points lying in the three rings defined by the distances given in Table 1.

The pseudo-code given in Algorithm 1, in which κ stands for the condition number of M and $\mathbb{I}$ for the index set of the selected points, sketches the algorithm for point selection. In each iteration the point of all points that have not been considered before is chosen, which if considered along with those points of the previous iterations yields a minimal condition number for M. This procedure is repeated until a user defined bound δ for the condition number is reached.

Algorithm 1 Point Selection Algorithm

Input: $\mathbb{X}$, U, Ω, δ # Indices of preselected points, POD basis, volumes and user defined bound for condition number

Output: $\mathbb{I}$ # Indices of selected points

1: $\tilde{P} = [\,],\ \ \mathbb{I} = \{\}$ # Initialize selection matrix $\tilde{P}$ and index set $\mathbb{I}$
2: $\kappa_{\min} = \infty$ # Initialize minimal condition number
3: **while** $\kappa_{\min} > \delta$ # Repeat until target condition number is met
4: **for** $i \in \mathbb{X} \backslash \mathbb{I}$ # Loop over points that have not been considered
5: $\bar{P} = \left[\ \tilde{P} \quad e_i\ \right]$ # Add current index' unit vector to selection matrix
6: $\kappa_i = \kappa(U^T \bar{P} \tilde{P}^T \Omega U)$ # Compute condition number after adding index
7: **end for**
8: $j = \arg\min_i \kappa_i$, # Determine optimal index
9: $\kappa_{\min} = \kappa_j$ # Update minimal condition number $\kappa_{\min}$
10: $\mathbb{I} = \mathbb{I} \cup \{j\}$, # Update index set $\mathbb{I}$ with chosen index
11: $\tilde{P} = \left[\ \tilde{P} \quad e_j\ \right]$ # Update selection matrix $\tilde{P}$
12: **end while**

3 Results

The Missing Point Estimation approach is tested for the NACA 0012 airfoil in this work. All snapshots are computed using the Euler equations with an unstructured grid with $n = 1140$ points (grid cells). Since we are computing four variables, the full order CFD model is of size $4n = 4560$. The grid is displayed in Figure 1 and 2. Note that the predicted solutions are never included in the snapshot set.

Note that we are looking for steady states, for which the residual $\vec{R}$ in (1) is zero. Therefore instead of time-integrating the MPE reduced order system (5), we search for the root of the right hand side. This is realized with Powell's Dog Leg method, which is implemented in the function *fsolve* of the subpackage *optimize* of *scipy*[12]. The function *fsolve* is a wrapper around MINPACKs *hybrd* and *hybrj* algorithms[13]. As a start vector for the root finding algorithm, the mean flow is chosen, i.e. $\vec{a} = \vec{0}$.

3.1 Test case 1: Subsonic flow with variation of the angle of attack

As a first simple test case a subsonic flow at a free stream Mach number of $M_\infty = 0.3$ is considered. Our goal is to predict the solution at $\alpha = 7°$. As snapshots we take steady-state solutions for the angles of attack $\alpha \in \{0°, 2°, 4°, 6°, 8°, 10°, 12°\}$ and at the same Mach number. After computing the POD, we take a look at the relative energy contained in each mode in L_2 sense, which is given by

$$E(d) = \frac{\sigma_d^2}{\sum_{i=1}^{m} \sigma_i^2},$$

where σ_i are the singular values of the POD. The relative energy is shown in Table 2.

d	$E(d)$	d	$E(d)$	d	$E(d)$
1	9.99091e-01	4	2.73605e-08	6	2.79418e-11
2	9.08121e-04	5	1.65420e-09	7	2.09805e-16
3	3.71393e-07				

Table 2: Relative information content of the first test case.

Note that the relative energy for $d = 7$ is numerically zero, that is, the rank of the snapshot matrix as given in (2) is $d-1$. It can easily be seen that this is due to the subtraction of the average from the columns of Y, whereby the columns become linearly dependent. In fact, the average can be considered as a mode and the dth mode should always be disregarded since it carries no information whatsoever.

Furthermore note that the high energy contained in the first mode is due to the L_2 scalar product.

We want to focus our attention on the impact of the truncation level of the POD modes. For this purpose we do the following test: We choose all 347 points, which lie in the three rings defined by Table 1, in order to keep the number of selected points fixed. In this way there is no influence of the number of points on the comparison of the accuracy for different choices of the number d of kept modes.

Table 3 shows the lift and drag coefficient c_l and c_d as well as the associated errors for the different truncation levels. It can be seen that by increasing the number of modes, the error can be reduced, but at the same time the number of residual evaluations increases. Furthermore, it is interesting to see that although only $\tilde{n} = 347$ out of 1140 are taken into account, the error is almost negligible when choosing all modes. This means that the information contained in some of the points suffices to get a very good agreement with the CFD full order reference solution.

modes	residual evaluations	c_l (error in %)	c_d (error in %)
$d = 1$	7	0.8442 (2.52%)	-0.007557 (214.07%)
$d = 2$	9	0.8643 (0.20%)	0.006912 (4.33%)
$d = 3$	10	0.8637 (0.27%)	0.006999 (5.65%)
$d = 4$	12	0.8645 (0.17%)	0.006848 (3.37%)
$d = 5$	13	0.8665 (0.06%)	0.006542 (1.25%)
$d = 6$	14	0.8660 (0.00%)	0.006618 (0.11%)
CFD	252	0.8660	0.006625

Table 3: Lift and drag coefficients compared for different truncation levels for the first test case.

Furthermore it can be seen that the more modes are taken into account, the more residual evaluations are needed and consequently, the more computationally expensive the reduced order model will be. For this reason we choose $d = 2$ to keep the computational costs low for the reduced order model. Note that the modes carry more than 99.9999% of the energy measured in L_2 norm.

Next, Algorithm 1 is applied. For this test case it yields only one point lying on the farfield of the computational domain, such that M has a condition number of

$\kappa(M) = 1.01$. With this single point, i.e. $\tilde{n} = 1$, we construct the MPE reduced order model (5). Figure 3 compares the plots of the pressure distribution c_p on the surface of the airfoil for the full and reduced order model. Both bear a striking resemblance to one another. The lift and drag coefficient c_l and c_d is exhibited in Table 4. It can be seen that the relative error for the lift coefficient compared to the full order solution is only 0.20%. The error for the drag coefficient is higher and is about 4.33%.

We want to investigate the influence that the number of selected points $\tilde{n}$ has. To this end, we modify line 3 of Algorithm 1 such that the loop is not ended when a target condition number is met, but when a certain number of selected points is reached. In Table 4 the lift and drag coefficient c_l and c_d for the MPE and the CFD reference solution are depicted.

selected points	residual evaluations	c_l (error in %)	c_d (error in %)
$\tilde{n} = 1$	9	0.8643 (0.20%)	0.006910 (4.30%)
$\tilde{n} = 10$	10	0.8643 (0.20%)	0.006910 (4.30%)
$\tilde{n} = 50$	10	0.8643 (0.20%)	0.006917 (4.41%)
$\tilde{n} = 200$	9	0.8643 (0.20%)	0.006916 (4.39%)
$\tilde{n} = 347$	9	0.8643 (0.20%)	0.006912 (4.33%)
$n = 1400$	9	0.8644 (0.18%)	0.006976 (5.30%)

Table 4: Lift and drag coefficients compared for different selected points for the first test case.

It can be observed that both the number of residual evaluations and the relative errors are about the same, no matter how many points are chosen. Interestingly, when taking into account all points the error for the drag coefficient c_d is a bit higher compared to disregarding some points.

Finally, we want to give a few remarks on the computational time. Since even the full order solution takes only 1.2 CPU seconds, all computational times, which are given, are error-prone and should therefore be regarded as trends rather than exact measurements. Furthermore note that the full order CFD solver TAU[8] is a highly optimized, industrially used C code, whereas the MPE is implemented in C and python and the code is rather prototypical.

The full order solution is computed such that the residual norm is dropped by seven orders of magnitude and the computation is started from free-stream values. For acceleration purposes a geometrical multigrid algorithm is used. With these settings a computational time of 1.2 CPU seconds is obtained in each of ten conducted runs.

Since the point-wise residual evaluation is not implemented efficiently yet, the computational times for the Missing Point Estimation cannot be given exactly. However, the computation when taking into account all points can be seen as an upper bound for the time needed for computing the POD and solving the system, if the number of residual evaluations is the same. The overall time for this computation is 0.228 CPU seconds (averaged out for ten runs), including 0.066 CPU seconds for computing the POD.

Note that for the before mentioned computation no effort has to be put into selecting the points with Algorithm 1. The computational times of the point selection algorithm for different numbers of selected points shall be investigated next. They

are given in Table 5, where each given time is the average of ten runs. Note that the computational time for one single point is negligible compared to the effort of computing the POD and solving the reduced system, while for ten points it is about the same. If $\tilde{n} = 50$ points are selected, using Algorithm 1 takes as long as a full order solution.

selected points	$\tilde{n} = 1$	$\tilde{n} = 10$	$\tilde{n} = 50$	$\tilde{n} = 200$
computational time (in CPU seconds)	0.026	0.248	1.173	3.647

Table 5: Computational time for Algorithm 1 for the first test case.

3.2 Test case 2: Transsonic flow with variation of the angle of attack

In the following we want to test the MPE on a flow in the transsonic flow regime at Mach number $M_\infty = 0.73$. Flows of this type are more relevant for designing wings and airplanes, since nonlinear effects have to be accounted for.

We want to predict the flow at an angle of attack of $\alpha = 5°$ and use snapshots at the angles of attack $\alpha \in \{3°, 4°, 6°, 7°\}$.

Having computed the POD, we take a look at the energy $E(d)$ contained in the modes given in Table 6. Once again the first mode carries most of the energy, which is due to the L_2 scalar product.

d	1	2	3	4
$E(d)$	9.99874e-01	1.15971e-04	9.97872e-06	2.21348e-18

Table 6: Relative information content of the second test case.

Next, we want to investigate the impact of the truncation level to the error in the lift and drag coefficient. To this end, we select all points contained in the three rings, as we have done in the previous test case. Table 7 presents the relative errors in the lift and drag coefficient for different truncation levels. Once again, the more modes are taken into account, the more residual evaluations are needed to find the solution. This time, using all three modes leads to a slightly greater error in the aerodynamic coefficients than taking only two modes. Note that a relative error of less than one precent can be achieved with $d = 2$.

modes	residual evaluations	c_l (error in %)	c_d (error in %)
$d = 1$	7	8.991e-01 (3.62%)	6.722e-02 (3.41%)
$d = 2$	12	9.416e-01 (0.93%)	7.007e-02 (0.69%)
$d = 3$	15	9.510e-01 (1.94%)	7.109e-02 (2.16%)
CFD	168	9.329e-01	6.959e-02

Table 7: Lift and drag coefficients compared for different truncation levels for the second test case.

In Figures 4, 5, and 6 the pressure distribution plots are shown. While in case that only one mode is used the shock is smeared out by the MPE, when using more modes the shock location is resolved well. However, the pressure level upstream of

the shock is overestimated. Since the pressure distribution is matched better when three modes are used, in the following tests we keep all $d = 3$ modes. In this way more than 99.9999% of the energy is contained in the modes.

When applying Algorithm 1 with the user defined bound δ set to 1.5, three points lying on the farfield are selected. This yields a condition number of $\kappa(M) = 1.446$. The first row of Table 8 shows that the error in the lift and drag coefficient is less than one percent. Furthermore Figure 7 depicts the pressure distribution plot for the surface. It can be observed that the shock location is nicely matched. Unlike before, the pressure level upstream of the shock is underestimated instead of overestimated.

In the following the influence of the point selection onto the solution is addressed. For this purpose we choose more points with Algorithm 1. Table 8 illustrates the number of residual evaluations and the error in the lift and drag coefficient for different point selections. Unlike in the subsonic test case, the number of residual evaluations increases with the number of points taken into account.

selected points	residual evaluations	c_l (error in %)	c_d (error in %)
$\tilde{n} = 3$	9	9.278e-01 (0.55%)	6.928e-02 (0.45%)
$\tilde{n} = 10$	12	9.304e-01 (0.27%)	6.920e-02 (0.56%)
$\tilde{n} = 50$	12	9.300e-01 (0.31%)	6.918e-02 (0.59%)
$\tilde{n} = 200$	13	9.359e-01 (0.32%)	6.952e-02 (0.10%)
$\tilde{n} = 347$	15	9.510e-01 (1.94%)	7.109e-02 (2.16%)
$n = 1400$	-	-	-

Table 8: Lift and drag coefficients compared for different selected points for the second test case.

While the relative error in the lift and drag coefficient is less than one percent for the first four rows in Table 8, it is about four times higher when all points of the three rings are regarded. This suggests that noise is added by including some points.

In the following we will investigate the influence a particular point has, which lies directly on the shock. Its location is illustrated in Figure 8. Table 9 shows the errors in the aerodynamic coefficients when the point is not included ($\tilde{n} = 316$) and when it is included ($\tilde{n} = 317$). Note that the error both for c_l and c_d is more than ten times higher, when adding the point to the selection. In addition, the overestimation of the pressure level upwind of the shock is stronger, which can be seen in the c_p plots in Figures 9 and 10. This indicates that the MPE is very sensitive to shocks.

selected points	residual evaluations	c_l (error in %)	c_d (error in %)
$\tilde{n} = 316$	40	9.348e-01 (0.20%)	6.941e-02 (0.26%)
$\tilde{n} = 317$	45	9.604e-01 (2.95%)	7.151e-02 (2.76%)

Table 9: Lift and drag coefficients compared for different selected points for the second test case.

As a matter of fact, if all points of the computational domain are chosen, then there is no convergence at all. This is due to an unphysical value, reported by the flow solver TAU[8] that occurs while computing the Jacobian for Powell's Dog Leg method.

Finally, the computational time shall be addressed. Again, the CFD reference solution is computed with the convergence critrion that the residual norm decreases by seven orders of magnitude. Using multigrid, the main loop of the flow solver takes 0.8 CPU seconds in all ten conducted runs.

Since the reduced order model with all points taken into account does not converge, we will consider the reduced order model with $\tilde{n} = 3$ points. The overall computation time is 0.306 CPU seconds averaged out for three independent runs. This includes 0.055 and 0.093 CPU seconds for POD and Algorithm 1, respectively. Note that the pointwise residual evaluation is not implemented efficiently yet, so that the time needed is in fact less than the numbers given.

Finally, Table 10 shows the computational times for the point selection algorithm. Note that they are a bit higher than in the previous test case, since three instead of two modes are taken into account. It can be seen that selecting about ten points is still reasonable, but when choosing more, the point selection outweighs the savings of the reduced order model.

selected points	$\tilde{n} = 3$	$\tilde{n} = 10$	$\tilde{n} = 50$	$\tilde{n} = 200$
computational time (in CPU seconds)	0.093	0.302	1.416	4.418

Table 10: Computational time for Algorithm 1 for the second test case.

4 Conclusions and outlook

In this work a method for efficiently computing steady flows has been proposed. It is based on Proper Orthogonal Decomposition and the idea that the reduced order model does not require the information at all computational grid points, if an appropriate basis is available.

The Missing Point Estimation has been tested for two different test cases; one in the subsonic and the other one in the transsonic flow regime. In both cases only few points are needed to predict the lift and drag coefficient as well as the pressure distribution reasonably accurate.

Two different strategies for selecting the points, which are considered in the projection of the MPE, have been presented. The first chooses the points based on their location with respect to the distance from the surface and selects a lot of points, while the second one is based on minimizing the condition number of a matrix that is part of the projection. With the later strategy only few points are determined, but this comes at a high computational cost, as it has been shown in the test cases. Nonetheless, it seems to be an excellent means for point selection.

For applications, in which a fast real-time model is needed and the off-line cost is irrelevant, the point selection strategy based on the minimization of the condition number will be very useful. An example could be the optimal control of a flow, where the control is computed and fed to the experiment, which is ongoing in the wind tunnel simultaneously.

In the second test case in the transsonic flow regime we have seen that the MPE is very sensitive to shocks. Having included a point, which lies in the shock region, the error increased dramatically in both the aerdynamic coefficients as well as the pressure distribution. When selecting all points, there was no convergence due to an unphysical value in the flow.

Besides this convergence problem, it has been observed in other test cases that on the one hand the root finding algorithm stagnates and thus does not find a solution. And on the other hand, it sometimes does find a solution, which - however - is far from the reference solution. These issues have to be addressed in future work.

Acknowledgement

This work is supported by the German Federal Ministry of Economics and Technology (BMWi), grant no. 20 A 06 04 G. The authors would like to thank their collaborators in the *ComFliTe* project Stefan Görtz, Ralf Zimmermann, and Michael Mifsud from the Institute of Aerodynamics and Flow Technology of the German Aerospace Center (DLR) in Braunschweig for their help and support.

REFERENCES

[1] Rowley, C., Colonius, T., and Murray, R. Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena* **189**(1-2), 115–129 (2004).

[2] Lucia, D. J., King, P. I., and Beran, P. S. Reduced order modeling of a two-dimensional flow with moving shocks. *Computers & Fluids* **32**(7), 917 – 938 (2003).

[3] Lucia, D., Beran, P., and Silva, W. Reduced-order modeling: new approaches for computational physics. *Progress in Aerospace Sciences* **40**(1-2), 51–117 (2004).

[4] Astrid, P. *Reduction of process simulation models: A proper orthogonal decomposition approach*. PhD thesis, Technische Universiteit Eindhoven, (2004).

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

[6] Astrid, P. and Verhoeven, A. Application of Least Squares MPE technique in the reduced order modeling of electrical circuits. In *Proceedings of the 17th Int. Symp. MTNS*, 1980–1986, (2006).

[7] Cardoso, M. A., Durlofsky, L. J., and Sarma, P. Development and application of reduced-order modeling procedures for subsurface flow simulation. *International Journal for Numerical Methods in Engineering* **77**(9), 1322–1350 (2009).

[8] Gerhold, T., Friedrich, O., Evans, J., and Galle, M. Calculation of complex three-dimensional configurations employing the DLR-TAU-code. *AIAA paper* **167** (1997).

[9] Blazek, J. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, first edition, (2001).

[10] Volkwein, S. Model reduction using proper orthogonal decomposition. Lecture Notes, Institute of Mathematics and Scientific Computing, University of Graz. (URL: http://www.uni-graz.at/imawww/volkwein/POD.pdf).

[11] Willcox, K. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & Fluids* **35**(2), 208 – 226 (2006).

[12] Jones, E., Oliphant, T., Peterson, P., et al. SciPy: Open source scientific tools for Python, (2001–).

[13] Moré, J. J., Garbow, B. S., and Hillstrom, K. E. User guide for MINPACK-1. Technical Report ANL-80-74, Argonne National Laboratory, Argonne, IL, USA, , August (1980).

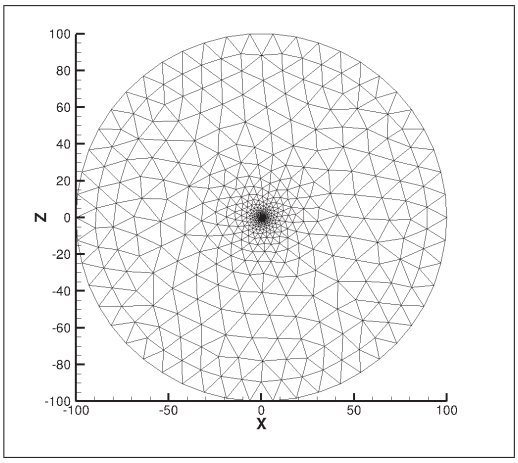

Figure 1: The grid for the NACA 0012 airfoil with $n = 1140$ points used for all computations.

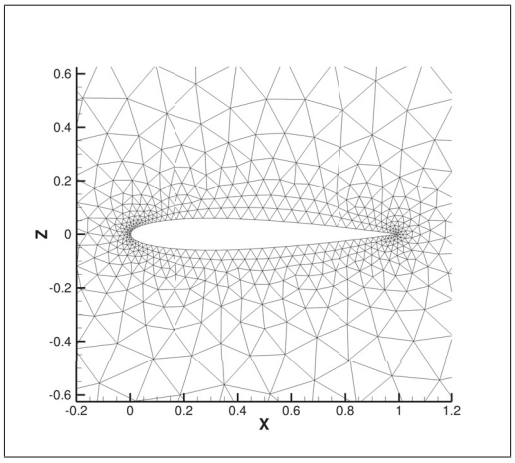

Figure 2: The grid for the NACA 0012 airfoil used for all computations zoomed in on the surface.

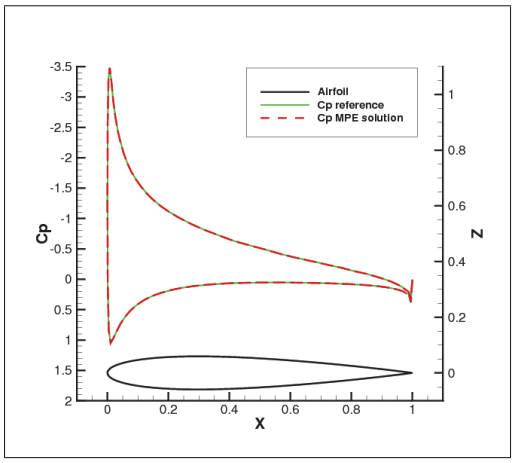

Figure 3: The pressure distribution c_p for the subsonic test case ($M_\infty = 0.3$) with only one point used for MPE.

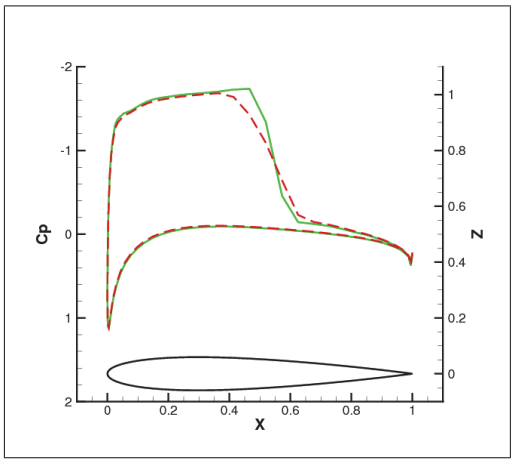

Figure 4: The pressure distribution c_p for the transsonic test case ($M_\infty = 0.73$) with one mode and $\tilde{n} = 347$ points used for MPE.

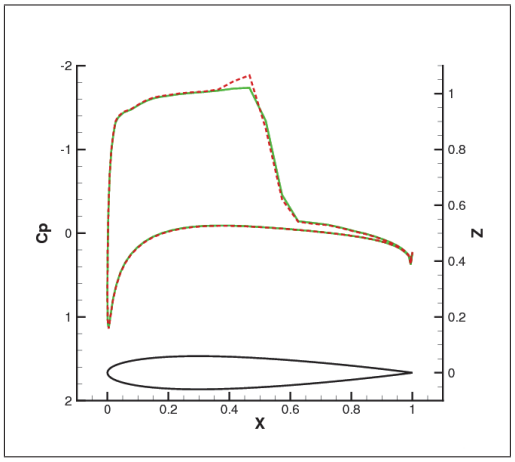

Figure 5: The pressure distribution c_p for the transsonic test case with two modes and $\tilde{n} = 347$ points used for MPE.

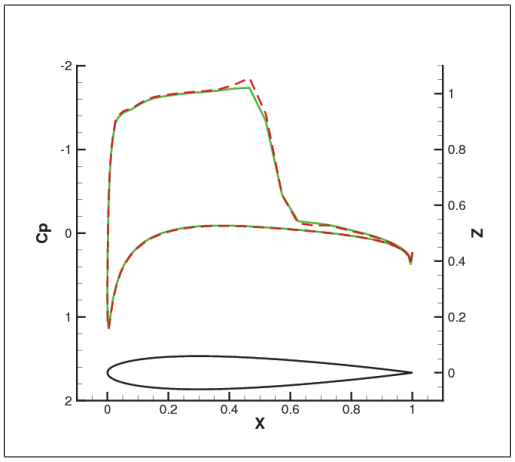

Figure 6: The pressure distribution c_p for the transsonic test case with three modes and $\tilde{n} = 347$ points used for MPE.

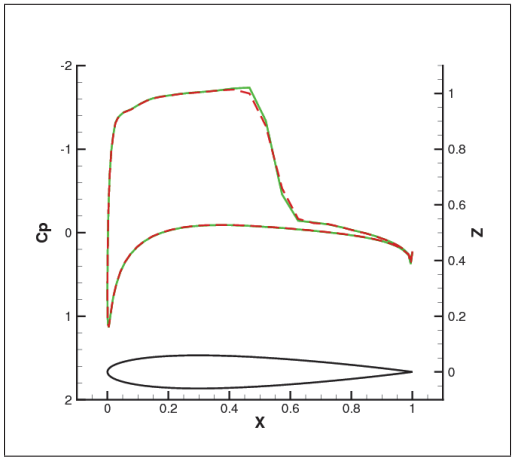

Figure 7: The pressure distribution c_p for the transsonic test case with three modes and $\tilde{n} = 3$ points determined by Algorithm 1.

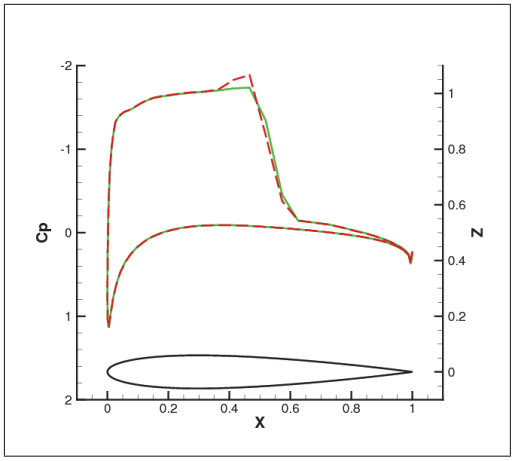

Figure 9: The pressure distribution c_p for the transsonic test case without the point lying on the shock.

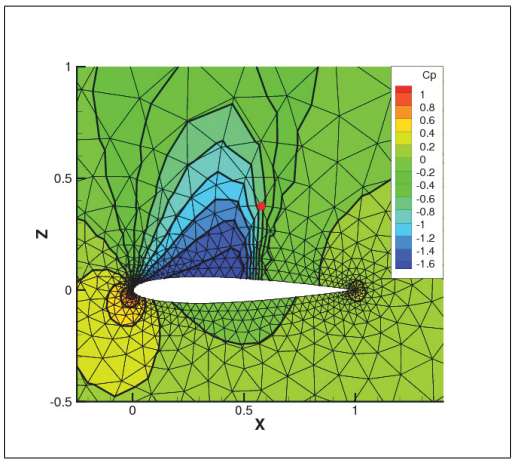

Figure 8: The point, which leads to the noise in the transsonic test case and lies directly on the shock.

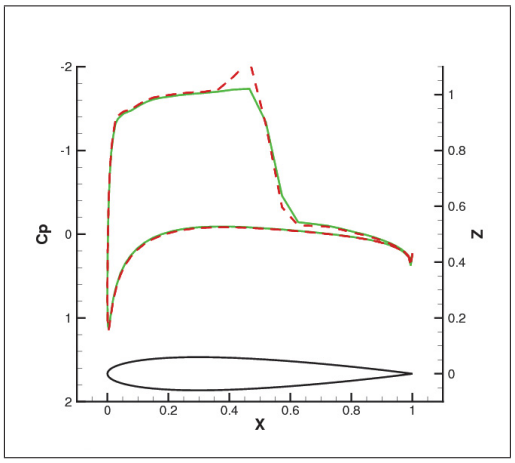

Figure 10: The pressure distribution c_p for the transsonic test case with the point lying on the shock.