



**AIAA 2003–0250**

**Dynamic Domain Decomposition  
and Error Correction for Reduced  
Order Models**

Patrick A. LeGresley and Juan J. Alonso  
*Stanford University, Stanford, CA 94305*

**41st AIAA Aerospace Sciences Meeting &  
Exhibit**

**January 6–9, 2003/Reno, NV**

# Dynamic Domain Decomposition and Error Correction for Reduced Order Models

Patrick A. LeGresley\* and Juan J. Alonso†  
*Stanford University, Stanford, CA 94305*

The use of reduced order models based on Proper Orthogonal Decomposition (POD) has been demonstrated to be of practical use for modeling low-speed aerodynamics. However, the use of such techniques for transonic flows with the presence of shocks has been more limited. This is because the reduced order model will only be able to represent a shock in a given location if one of the snapshots used to build the model has a discontinuity at that same location. Lucia *et. al.* proposed domain decomposition as a means of overcoming this limitation. This technique uses a reduced order model over the majority of the computational domain while solving the full governing equations in a small region which would otherwise require an unacceptably high number of snapshots to achieve sufficient accuracy of the approximate solution. In this work, we extend the concept of domain decomposition as a dynamic, *a posteriori* verification and if necessary, correction of the approximate solution. Given an approximate solution with unknown accuracy generated with a set of POD basis functions the error is estimated by augmenting the POD basis with top hat basis functions and computing the first order change in the solution due to the additional basis functions. By comparing the results from a solution of known accuracy, such as one of the snapshots used to generate the POD basis, with the results for a solution of unknown accuracy the need for domain decomposition and its spatial extent can be determined. Once the boundaries of the domain decomposition have been determined, a POD solver coupled with a full order solver can be used to generate solutions of a coupled system. Results for 2-D transonic flow show that this method can generate good approximate solutions for flows with significant shock movement. Application to a sample drag minimization problem indicate that this type of variable fidelity model would be a good candidate for use in a multilevel optimization framework.

## Nomenclature

$E$	total energy (internal plus kinetic)
$\mathbf{f}, \mathbf{g}$	Euler flux vectors
$H$	total enthalpy
$M$	number of modes used in approximation
$p$	static pressure
$R(x, x')$	autocorrelation function
$\mathbf{R}$	autocorrelation tensor, residual
$\mathcal{R}$	autocorrelation matrix for method of snapshots
$u$	$x$ -component of velocity
$v$	$y$ -component of velocity
$\mathbf{v}$	vector of state variables
$\mathbf{u}$	arbitrary function to be generated
$x$	vector of independent variables
$\lambda$	Lagrange multiplier, an eigenvalue
$\eta_i$	coefficient of the $i$ -th mode in a function expansion
$\Omega$	domain of interest
$\rho$	density
$\varphi^j(x)$	$j$ -th POD basis mode
$\langle \cdot \rangle$	averaging operator
$\  \cdot \ $	$L^2$ -norm

## Introduction

REDUCED order models based on Proper Orthogonal Decomposition (POD) have been used for aerodynamic shape optimization,<sup>1,2</sup> flutter<sup>3</sup> and panel response.<sup>4</sup> However, the flows which could be modelled with a sufficient level of accuracy using a reasonable database were restricted to subsonic flows or flows with weak shocks which had little movement. The inherent simplicity of the linear basis generated via POD also limits it to representing phenomena which have been observed in the database used to construct it.

A simple example using a database consisting of flow solutions for a single airfoil geometry at fixed angle of attack and varying Mach numbers illustrates the problem. In Figure 1 are numerical results computed using FLO82<sup>5</sup> for the flow over the RAE 2822 airfoil at 3° angle of attack at Mach numbers of 0.3, 0.4, 0.5, 0.6 and 0.7. The flow at the first three Mach numbers is entirely subsonic, while the flow at Mach 0.6 has a small shock at approximately 5% chord, and the flow at Mach 0.7 has a significant shock at approximately 50% chord. Figure 2a shows a POD solution for the flow at Mach 0.35 compared with the full order solution. For this subsonic case the results are virtually identical. However, Figure 2b shows a solution at Mach 0.67 in which the POD solution is essentially

\*Graduate Student, Student Member AIAA

†Assistant Professor, AIAA Member

Copyright © 2003 by the authors. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

useless as an approximation of the true solution. For reference the subspace projection of the true solution into the space defined by the basis functions is also shown. This simply verifies that the basis is inherently not capable of representing the true solution.

This inability to represent a moving shock essentially eliminates the use of these models for the transonic flow regime. However, Lucia *et. al.*<sup>6</sup> proposed and demonstrated the decomposition of a flow with moving shocks into a subdomain in which the flow is represented with a POD model and a small subdomain in which a full order flow solver is used. In this case the domain was decomposed *a priori* with a basis computed only for the subdomain which was anticipated to be representable via POD.

The basic idea is that if the basis used to generate the approximate solutions is of sufficient accuracy, it is likely that the solution will be acceptable in most of the domain with just a small portion of the domain which cannot be resolved using the POD basis. In this case a domain decomposition approach allows one to solve the full order governing equations in only a small portion of the domain while still using the original reduced order model over the majority of the domain. This is equivalent to augmenting the POD basis, which is expensive to obtain, with very inexpensive top hat basis functions.

The fidelity of this coupled model can be varied from extremely low - only a handful of degrees of freedom coming solely from POD - to very high as the model recovers the fidelity of a full order solver in the limit. This type of behavior appears to be a good fit for multilevel optimization.<sup>7</sup> The goal of multilevel optimization is to use tools of varying fidelity to perform accurate design optimization on high fidelity models while making minimal costly computations on the high order models. Typically, multilevel optimization has employed data fitting approximations, variable convergence models, or models of varying physical fidelity. However, depending on the ability of the lower fidelity models to accurately capture the behavior of the higher fidelity models, the multilevel method may be no better than conventional optimization. The coupled approximation model seems to satisfy the needs of multilevel optimization by providing a low cost, approximate model of the system with the ability to capture many of the higher order effects and accuracy in the limit as we vary the extent of the approximated portion.

It would be preferable to only add additional basis functions where they are necessary to resolve a particular solution. Therefore, in addition to the solver that actually does the correction, a way is needed to identify that portion (if any) of the domain that needs to be corrected. The next sections give an overview of the POD method, the approximate solution methodology, a method for determining the need, and, if necessary,

the spatial extent for domain decomposition, the coupling methodology for the approximate and full order solvers, and results of the methodology applied to transonic flow over an airfoil.

## Proper Orthogonal Decomposition

Complete details of the POD method have been described in Holmes *et. al.*<sup>8</sup> and only a brief review is presented here. The POD is a procedure for computing the optimal linear basis for representing a set of data. This data could be obtained from experimental results or, as presented here, by the numerical solution of a system of partial differential equations. The projection of a function of interest onto the basis functions (or modes) provides a finite set of scalar coefficients that represent that function. The order of the system is reduced from very large numbers (millions for the external aerodynamics of a full aircraft) to very small numbers (in the tens or hundreds). If the error in such an approximation is small enough a useful description of the system can be generated at a significantly reduced cost.

The POD describes a specific set of linear basis functions which make the description of the system optimal for a set of observations of finite size. An approximation to a function  $\mathbf{u}(x)$  can be constructed from a basis  $\{\varphi_j(x)\}_{j=1}^{\infty}$  as

$$\mathbf{u}_M = \sum_{j=1}^M \eta_j \varphi_j(x). \quad (1)$$

To define the basis, assume that we have a set of  $N$  datasets (such as the results from several computational runs) denoted by  $\{\mathbf{u}^k\}$ . We would like to minimize the error in truncating the summation in Eq. 1 at  $M$  basis functions, which mathematically means that the basis functions  $\varphi_j(x)$  must be the maximizers of the expression

$$\max_{\varphi} \frac{\langle |\mathbf{u}, \varphi|^2 \rangle}{\|\varphi\|^2}, \quad (2)$$

where  $|\cdot|$  denotes the modulus and  $\|\cdot\|$  is the  $L^2$ -norm given by

$$\|f\| = (f, f)^{\frac{1}{2}},$$

and the notation  $(\cdot, \cdot)$  expresses the inner product of two functions over a pre-defined interval or domain.

The expression in Eq. 2 exhibits multiple extrema which constitute the basis functions in the linear decomposition in Eq. 1. This is a calculus of variations problem in which we would like to maximize  $\langle |\mathbf{u}, \varphi|^2 \rangle$  subject to the constraint that  $\|\varphi\|^2 = 1$  and may be shown to require (see Ref.<sup>8</sup>) that the basis functions satisfy

$$\int_{\Omega} \langle u(x)u(x') \rangle \varphi(x') dx' = \lambda \varphi(x). \quad (3)$$

The POD basis is therefore composed of the eigenfunctions,  $\{\varphi_j\}$ , of the integral Eq. 3. In the process of constructing basis modes for discrete datasets from computational results, we will be dealing with results in the form of an ensemble of functions,  $\mathbf{u}^k$ , which are now a group of  $N$ -dimensional vectors. In this case, the kernel in Eq. 3 becomes the autocorrelation tensor

$$\mathbf{R} = \langle \mathbf{u} \otimes \mathbf{u} \rangle.$$

and the integral eigenvalue problem becomes

$$\mathbf{R}\varphi = \lambda\varphi.$$

After computing the basis functions or vectors, any member of the ensemble  $\mathbf{u}^k$  can be decomposed as

$$u(x) = \sum_{j=1}^{\infty} \eta_j \varphi^j(x). \quad (4)$$

The eigenvalues have the physical interpretation of measuring the magnitude of the quantity in Eq. 2. Modes with larger eigenvalues correspond to greater maxima than smaller eigenvalues. In a sense they measure the amount of the variation attributed to a particular mode, and this leads to a sorting of the basis functions in descending eigenvalue order.

For representing a large system (in terms of the dimension  $N$  of  $\mathbf{u}$ ) with a relatively small number of modes the computational cost of solving the  $N \times N$  eigenvalue problem may be substantial. A procedure called the method of snapshots due to Sirovich<sup>9</sup> reduces the problem to an  $M \times M$  problem where  $M$  is the number of ensemble functions.

Using the method of snapshots the modified autocorrelation matrix is given by

$$\mathcal{R}_{ij} = \frac{1}{M} \int_{\Omega} u_i u_j d\Omega \quad (5)$$

where  $u_i$  is the  $i$ -th snapshot and  $i, j = 1, 2, \dots, M$ , and  $M$  is the total number of snapshots. The eigenvectors of  $\mathcal{R}$  are computed as an intermediate step

$$\mathcal{R}a = \lambda a, \quad (6)$$

from which the POD basis functions can be calculated as

$$\varphi^K = \sum_{i=1}^M a_i^K u_i(x, y) \quad K = 1, 2, \dots, M \quad (7)$$

where  $a_i^K$  is the  $i$ -th element of eigenvector  $a$  corresponding to the eigenvalue  $\lambda_K$ .

For a system of equations we could apply the methodology described to each state variable and form a distinct set of basis modes for each.<sup>1</sup> However, this neglects the cross-coupling between the variables. By considering not only how the individual variables vary from one snapshot to another but also how variables

change relative to one another we include additional information in the basis and improve the ability of the basis to represent the system under consideration.

Therefore, we define a *vector* state variable consisting of the variables to be represented by the linear expansion. For the Euler equations we use

$$\mathbf{w} = (\rho, \rho\mathbf{u}, \rho E),$$

where  $\rho$ ,  $\rho\mathbf{u}$ , and  $\rho E$  are the density, momenta, and total energy respectively and the variables are assumed to have zero mean. The inner product is then computed as

$$(\mathbf{w}^{(l)}, \mathbf{w}^{(m)}) = \int_{\Omega} \sum_{k=1}^N \bar{\mathbf{w}}_k^{(l)}(\mathbf{x}) \mathbf{w}_k^{(m)}(\mathbf{x}) d\Omega, \quad (8)$$

where  $N = \dim(\mathbf{w})$  and for generality the bar denotes complex conjugation.

Once this procedure has been completed, we can expand a complete flow solution in the form

$$\mathbf{w}(x, y) = \sum_{i=1}^M \eta_i \varphi_i(x, y). \quad (9)$$

## Flow Analysis Procedure

The solution method is based on a finite-volume approach to the discretization of the governing equations. Let  $p$ ,  $\rho$ ,  $u$ ,  $v$ ,  $H$ , and  $E$  denote the pressure, density, Cartesian velocity components, total enthalpy, and total energy respectively. Then for an arbitrary control volume,  $\Omega$ , with boundary  $\partial\Omega$ , the governing equations may be expressed in integral form as

$$\frac{d}{dt} \iint_{\Omega} \mathbf{w} \, dx \, dy + \oint_{\partial\Omega} (\mathbf{f} \, dy - \mathbf{g} \, dx) = \mathbf{0}, \quad (10)$$

where  $\mathbf{w}$  is the vector of conserved flow variables

$$\mathbf{w} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{Bmatrix},$$

and  $\mathbf{f}$ ,  $\mathbf{g}$  are the Euler flux vectors

$$\mathbf{f} = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u H \end{Bmatrix}, \quad \mathbf{g} = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho v H \end{Bmatrix}.$$

Also, for an ideal gas, the equation of state may be written as

$$p = (\gamma - 1) \rho \left[ E - \frac{1}{2}(u^2 + v^2) \right].$$

For each cell in a mesh, Eq. 10 can be applied independently to obtain a set of ordinary differential equations of the form

$$\frac{d}{dt} (\mathbf{w}_{ij} V_{ij}) + \mathbf{R}(\mathbf{w}_{ij}) = \mathbf{0}, \quad (11)$$

where  $V_{ij}$  is the volume of the  $i, j$  cell and the residual  $\mathbf{R}(\mathbf{w}_{ij})$  is obtained by evaluating the flux integral in Eq. 10. In the steady state, the time derivative term drops out and we are left with

$$\mathbf{R}(\mathbf{w}_{ij}) = \mathbf{0}, \quad (12)$$

which incorporates the wall and far-field boundary conditions in the calculation of the boundary fluxes at the edges of the domain.

Using a linear expansion of the form given in Eq. 9, many methods exist for exploiting the reduction in complexity of the problem. These methods all fall under the general class of weighted residual methods and include collocation, least squares, and the Galerkin method.<sup>10</sup> In the limit of a large number of basis functions the solutions from all methods tend to be similar, but for a small set of basis functions the solutions can vary significantly depending on the particular residual weighting method used. Unfortunately there is no way to know *a priori* which method will yield the best results. The strategy used here is to minimize the residual (12) in a least squares sense.

Defining the square of the residuals where  $\mathbf{R}$  can now be expressed as a function of  $\eta$

$$\tilde{R}(\eta) = \sum_{i,j} \mathbf{R}(\eta)^2 \quad (13)$$

the governing equations for the POD solution are then

$$\hat{R}_k(\eta) = \frac{\partial \tilde{R}}{\partial \eta_k} = 0. \quad (14)$$

The solution to Eq. 14 can be obtained using the Levenberg-Marquardt method which uses first derivatives of the residual with respect to the modal coefficients. Typically this solution method converges in three or fewer iterations.

The cost of the approximate solution is proportional to the number of modes used in the approximation and also has a dependence on the number of degrees of freedom in the original problem. This dependence of the approximate solution on the dimensionality of the original problem is in contrast to the linear case in which a POD solution can be obtained at a cost that is independent of the dimensionality of the original problem (excluding the upfront mode computation process).

To lessen the impact of the dimensionality for the non-linear case one can choose not to use every single residual in the least squares fit of (13). For example, in practice it was found that using only the equations for mass conservation led to results that were virtually indistinguishable from those using all four equations. In addition one could skip equations for some cells in each spatial direction to further reduce the cost. These approaches essentially achieve some of the low cost of

collocation with less dependence on the choice of collocation points and are sometimes referred to as least squares collocation. To achieve computational cost savings while still solving (13) in a mathematically rigorous sense, multigrid could of course be applied to the problem.

## Domain Decomposition

The coupling of the POD and full order solvers is essentially the same as that used in multiblock solvers where the spatial domain of the problem is broken into blocks and communication between subdomains is accomplished using halo cells surrounding each subdomain boundary. However, the coupling does require special attention to the interface between the approximate flow solution generated via POD and the full order solver.

The halo cells, or boundary conditions, for a subdomain of the full order solver will be an approximation generated by the POD solver and in general will not correspond to a physical solution since conservation cannot be mathematically guaranteed. To ensure conservation across the interface between the full order and approximate solver, sources/sinks of mass, momenta, and energy are placed in the first level of halo cells surrounding the full order domains. The strength of these sources/sinks is computed so as to leave a conservative interface between the POD and full order solver.

To generate the solution of the coupled POD and full order solver, we begin with the solution computed using the POD basis. Updating the POD solution is costly compared to iterations of the full order equations so we would like to make as few updates to them as possible. The portion of the solution domain which is going to still be represented using POD is fixed while explicit iterations are carried out in the full order domain, using the POD solution as an initial condition. After convergence has been reached, the solution in this domain is now fixed and the POD solution is adjusted based on the updated values. At this point the solution is typically well resolved, although further iterations could be carried out if necessary.

The computational cost of iterating a full order equation is identical to the cost in a full order solver. However, there are now far fewer of these equations. In our experience the number of iterations required to solve the full order equations in the coupled POD and full order system is approximately the same as the number of iterations required to solve the real full order system. Then there is the additional cost to recompute the solution in the POD subdomain as the solution in the full order subdomain evolves. This is typically one or two more iterations, beyond the three or four required to get the initial POD solution.

In the case of this 2-D work a multiblock solver was not available. To represent the multiblock solver in

this case modifications were made to FLO82<sup>5</sup> to solve the normal full order equations in only a specified portion of the domain. For a 3-D implementation it should be straightforward to use already developed multiblock solvers to accomplish this.

### Error Estimation

The coupling of the POD and full order solver allows one to generate solutions of widely varying computational cost by varying the spatial extent of the full order solver. But as is often the problem in reduced order modeling the degree of accuracy is unknown. Ideally we would like to have error bounds. Pierce and Giles<sup>11</sup> have developed a method of producing superconvergent functionals or error bounds by use of an adjoint problem. A discrete version of this technique was used by Venditti and Darmofal<sup>12</sup> to drive a grid adaptation procedure. This procedure could possibly be used to estimate the error for functionals of a POD solution and drive a refinement procedure by varying the spatial extent of the domain decomposition to generate solutions of specified accuracy.

There are several problems with applying the adjoint formulation as previously used to refine our approximate solutions. The approximate solution needs to be in the asymptotic range of the true solution, which as demonstrated in the earlier example that was used to motivate this work, may not be true for flows with shocks. And even if the solution is in the asymptotic range the adjoint solution, although a linear problem, would have nearly the same cost as computing the solution of the problem we are trying to approximate in the first place. This cost could be reduced if the adjoint solution were represented by a reduced order model also, but the adjoint solution itself often has shock type discontinuities and would suffer from the same types of problems as the primal problem.

However, a similar technique can be applied on subsets of the domain to prioritize which portions of the domain to augment with additional basis functions to generate the best solution for a given computational cost. Let subscript *POD* denote the solution as computed using the POD basis

$$\mathbf{w}_{POD} = \sum_{m=1}^M \eta_m \varphi_m(\mathbf{x}) \quad (15)$$

and *DD* the solution using the POD basis functions plus some additional top hat basis functions,  $\hat{\varphi}_n$ ,

$$\mathbf{w}_{DD} = \sum_{m=1}^M \eta_m \varphi_m(\mathbf{x}) + \sum_{n=1}^N \hat{\eta}_n \hat{\varphi}_n(\mathbf{x}) \quad (16)$$

where the addition of the top hat basis functions represents the decomposition of the problem into POD and full order subdomains. As few or as many top hat basis functions can be added and they can be placed

wherever one wants. However, the cost and accuracy of the error estimation will depend on the number of these basis functions that are used.

The residual operator representing the governing equations for this system are

$$\begin{Bmatrix} \hat{R}_k(\eta) \\ R_k(\hat{\eta}_n) \end{Bmatrix} = R_{DD} = \mathbf{0}. \quad (17)$$

Defining  $w_{DD}^{POD}$  as the solution using the expansion in (15) transferred to the basis in (16), the obvious transfer is that  $\eta$  is unchanged and  $\hat{\eta}$  are zero. The residual operator is then expanded as

$$\mathbf{R}_{DD}(\mathbf{w}_{DD}) = \mathbf{R}_{DD}(\mathbf{w}_{DD}^{POD}) + \frac{\partial \mathbf{R}_{DD}}{\partial \mathbf{w}_{DD}} \Big|_{\mathbf{w}_{DD}^{POD}} (\mathbf{w}_{DD} - \mathbf{w}_{DD}^{POD}) + \dots$$

This can be inverted to give an approximation of the error or change in the state vector

$$(\mathbf{w}_{DD} - \mathbf{w}_{DD}^{POD}) \approx - \left[ \frac{\partial \mathbf{R}_{DD}}{\partial \mathbf{w}_{DD}} \Big|_{\mathbf{w}_{DD}^{POD}} \right]^{-1} \mathbf{R}_{DD}(\mathbf{w}_{DD}^{POD}). \quad (18)$$

The change in the state vector (18) is the first order estimate of the change in the solution if one were to decompose the problem into a POD and full order subdomain. How useful and costly this estimate is depends on the number of additional basis functions that are introduced. If a single additional basis function is introduced the cost to get the error estimate is very modest, but it would only provide information at one point. Of course this process can be repeated for many points to build up a picture of what is going on. However, this completely ignores the higher order effects of introducing many additional basis functions at once.

One could introduce new basis functions at every point in the domain to get a complete view of the state of the solution, but as was pointed out previously this leads to a problem which has the same dimensionality as the one we are trying to approximate. Thus we must compromise between accuracy and cost. For this work we found that an estimate of the error calculated by introducing new basis functions in blocks of 8 by 8 cells (in a computational domain which was 160 by 32 cells) gave a good compromise between capturing the higher order effects of introducing multiple basis functions at once and computational cost.

The updated state vector leads to a change in the solution over the entire domain since the POD basis have global support. For approximate solutions that have significant error in one region of the domain, such as the vicinity of a shock, it was found that introducing new basis functions in other regions of the domain which were already well resolved led to changes not in the region where they were introduced but in the region where the solution was not good. It clearly makes more sense to introduce new basis functions for the

purpose of solution refinement in the regions where the solution is bad and not where the solution is already sufficient. Therefore, to characterize the need to introduce the new basis functions we used the change in the state vector to compute the corresponding change in pressure. Each block was then assigned a value equal to the norm of the change in pressure for the cells inside that block. This gives priority to placing basis functions that lead to a local change in the solution and not in other portions of the domain.

Now that we have prioritized where to introduce additional basis functions to refine the solution we still need to decide how many to actually add. Adding too few basis functions may not lead to a significant improvement in the solution while adding too many, although yielding a good solution, is expensive and may defeat the purpose of approximation to reduce computational cost. Comparison of the error estimates for solutions of known and unknown accuracy computed using the above procedure as well as observing how the solution changes from snapshot to snapshot seems to be a robust and computationally affordable means of performing the decomposition.

## Results

### Flow Computations

A test case using a set of modes for the RAE 2822 airfoil based on snapshots at an angle of attack of  $3^\circ$  and Mach numbers of 0.3, 0.4, 0.5, 0.6 and 0.7 illustrates the use of domain decomposition. Figure 3 shows the results of a domain decomposition using a coupled POD and full order solver for the RAE 2822 airfoil at Mach 0.67. This is the same test case as in Figure 2, the only difference being the domain decomposition technique is now being used. In Figures 3a and b we show the error estimates for Mach 0.60, which is one of the snapshots, and Mach 0.67. The scales have been set to accommodate the extrema of both datasets. For the Mach 0.60 case the error estimation is insignificant over the majority of the domain except for blocks on the first two-thirds of the upper surface of the airfoil. This is due to the fact that the solution, although exactly reconstructed via POD, is the result of an iterative solver which was not converged to machine zero. At Mach 0.67 the error is much more widespread and greatest over the forward portion of the upper surface.

Using the error estimation data, a decomposition was made and the resulting surface  $C_P$  is shown in Figure 3c. Comparison of this approximate solution with the full order solution and the POD solution in Figure 2 shows that only by utilizing domain decomposition is a useful approximation obtained. A contour plot of the percentage error in  $C_P$  in Figure 3d shows the flow field is very well represented except for a few cells in the region of the shock which is mispredicted in position by around 3% chord. The dashed line denotes

the boundary between the portion of the domain represented by the POD model and that computed using the full order solver. The POD solution with domain decomposition has 20% of the degrees of freedom compared with that of the full order solver.

To verify these results over a wide range of conditions a sweep from Mach 0.50 to 0.70 was made in increments of 0.01. In Figure 4 is a comparison of the full order and domain decomposition POD coefficients for lift, drag and pitching moment. The lift coefficient is good over the entire range. The drag rise is well captured with some oscillations over the initial onset of drag rise. It is unlikely that this phenomena could be eliminated without significantly increasing the computational cost. The pitching moment coefficient is also well predicted with some differences observed as the coefficient becomes significantly more negative at higher Mach numbers.

### Drag Minimization

The use of domain decomposition with POD generates reasonable approximations of the flow field, but for design purposes we are also interested in the ability of the model to generate gradient information. To test the use of the approximate gradients generated via this method a drag minimization problem was performed in which the geometry of an airfoil is modified to reduce drag while maintaining fixed lift. The angle of attack is allowed to vary and the shape modification is restricted to make only small reductions in the thickness of the airfoil to eliminate producing a trivial solution in which the airfoil is reduced to a flat plate. This type of problem can be solved very efficiently using an adjoint approach for calculating gradients, and we use it here only as a representative model problem.

For purposes of generating a suitable POD basis we need to obtain snapshots that contain variations in parameters that we are interested in, such as shape and angle of attack for the drag minimization problem. A baseline NACA 4410 airfoil was perturbed with 4 Hicks-Henne<sup>13</sup> bump functions on both the upper and lower surfaces. Flow solutions for the 9 airfoil geometries (1 baseline plus 8 perturbed versions) were computed at Mach 0.67 and angles of attack of  $1.5^\circ$  and  $1.6^\circ$  using FLO82 to yield a basis with 18 modes. The drag minimization was begun using the NACA 4410 airfoil at Mach 0.67 and an angle of attack of  $1.5^\circ$ . The design parameters were the amplitude of 8 bump functions placed on both the upper and lower surfaces and angle of attack. Gradients were computed by finite differencing of the approximate POD model to determine the direction of change and a line search was made at each iteration to determine the best step size. No information from a completely full order model was used during the design process.

The flow solution for the initial geometry can be represented exactly since it is one of the snapshots,

but the flow solutions for the slightly perturbed versions computed to obtain the gradients via finite difference cannot be represented exactly. The previously described error estimation procedure is sensitive to regions of large error but not to the small errors that may occur during finite differencing and could have a tremendous impact on the accuracy of the gradients. It was found that without the use of domain decomposition for cells nearest the airfoil the accuracy of the approximate gradients was insufficient.

Design results for a case in which the first 8 cells (out of a total of 32) in the direction normal to the airfoil were solved using the full order solver while the remaining 75% of the domain was represented by POD are shown in Figure 5. Airfoil geometries, surface  $C_P$  distributions and numerical data for the lift, drag, and angle of attack are shown in Figures 5a-d for the initial design and design iterations five, ten, and fifteen. Although the design was done using only reduced order model data, the full order solutions were computed for comparison purposes. Through the fifteen design cycles the drag coefficient as computed using the reduced order model went down from 0.0115 to  $-0.0112$  while the real drag as computed using the full order model decreased from 0.0115 to 0.0003. As seen in Figure 5e both the approximate and full order drag appear to be asymptoting after 7 design iterations. Despite the rather significant error in the actual value of the drag, the gradients are still accurate enough to generate a reasonable design solution.

To assess the accuracy of the final solution a design was run using the same bump functions as design variables but this time using the exact gradients. A comparison of the final geometry from this method and the geometry generated using approximate gradients, with exaggerated scale in the vertical direction, is shown in Figure 5f. The final results do vary, particularly in the shape of the forward portion of the lower surface.

To investigate the effect of the choice of decomposition in the problem another design was performed, this time representing 16 cells in the direction normal to the airfoil at full order and the remaining 50% with POD. The results of this design are shown in Figure 6. The actual reduction in drag is quite similar to the previous design although the drag as predicted by the reduced order model tracked with the real drag better than in the previous case. Comparison of the final geometry with that of the true solution shows very good agreement.

Although the two design cases were run separately, they illustrate the usefulness of the variable fidelity provided by coupling the POD and full order solver. One can imagine that if this problem were solved using a multilevel framework that much of the design work could be accomplished using the coarser models to achieve the exact solution at significantly reduced

computational cost.

## Conclusions

A procedure has been developed to use POD-based reduced order models for aerodynamic problems with moving shocks by applying an *a posteriori* check and, if necessary, a correction to a POD solution by decomposing the spatial domain of the problem into a portion represented by POD and a small portion in which the full order equations are used. These models retain much of the advantage of traditional POD based models but with the added robustness of a variable fidelity solver which in the limit recovers the exact solution.

To investigate the suitability of these types of models for use in design, a drag minimization problem was performed in which an airfoil geometry is modified to reduce drag by a gradient based method with the gradients approximated using the reduced order model. Although the results of the design performed using the reduced order model are not exactly the same as those from the full order model, the results are very similar.

Further work will extend the methodology to 3-D aerodynamics where the computational cost savings are more compelling than for the 2-D problem used here to demonstrate the technique. When fully integrated into a multilevel optimization procedure this method will produce accurate optimization results at significantly reduced computational cost and allow new design problems which are currently infeasible to be considered.

## References

- <sup>1</sup>P.A. LeGresley and J.J. Alonso. Airfoil design optimization using reduced order models based on proper orthogonal decomposition. *AIAA paper 00-2545*, Fluids 2000, Denver, Colorado, June 2000.
- <sup>2</sup>P.A. LeGresley and J.J. Alonso. Investigation of non-linear projection for pod based reduced order models for aerodynamics. *AIAA paper 01-0926*, 39th AIAA Aerospace Sciences Meeting & Exhibit, Reno, Nevada, January 2001.
- <sup>3</sup>C.L. Pettit and P.S. Beran. Reduced-order modeling for flutter prediction. *AIAA paper 00-1446*, AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Atlanta, Georgia, April 2000.
- <sup>4</sup>P.S. Beran and C.L. Pettit. Prediction of nonlinear panel response using proper orthogonal decomposition. *AIAA paper 01-1292*, 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Seattle, Washington, April 2001.
- <sup>5</sup>A. Jameson. Multigrid algorithms for compressible flow calculations. In W. Hackbusch and U. Trottenberg, editors, *Lecture Notes in Mathematics, Vol. 1228*, pages 166–201. Proceedings of the 2nd European Conference on Multigrid Methods, Cologne, 1985, Springer-Verlag, 1986.
- <sup>6</sup>D.J. Lucia, P.I. King, M.E. Oxley, and P.S. Beran. Reduced order modeling for a one-dimensional nozzle flow with moving shocks. *AIAA paper 01-2602*, AIAA 15th Computational Fluid Dynamics Conference, Anaheim, California, June 2001.
- <sup>7</sup>N. Alexandrov, R. Lewis, C. Gumbert, L. Green, and P. Newman. Optimization with variable-resolution models applied to aerodynamic wing design. *AIAA paper 00-0841*, AIAA



38th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2000.

<sup>8</sup>P. Holmes, J.L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, 1998.

<sup>9</sup>L. Sirovich. Turbulence and the dynamics of coherent structures. I - coherent structures. II - symmetries and transformations. III - dynamics and scaling. *Quarterly of Applied Mathematics*, 45:561–571,573–590, 1987.

<sup>10</sup>B.A. Finlayson. *The Method of Weighted Residuals and Variational Principles*. Academic Press, 1972.

<sup>11</sup>N.A. Pierce and M.B. Giles. Adjoint recovery of superconvergent functionals from PDE approximations. *SIAM Review*, 42:247–264, 2000.

<sup>12</sup>D.A. Venditti and D.L. Darmofal. A grid adaptive methodology for functional outputs of compressible flow simulations. *AIAA paper 01-2659*, AIAA 15th Computational Fluid Dynamics Conference, Anaheim, California, June 2001.

<sup>13</sup>R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15:407–412, 1978.

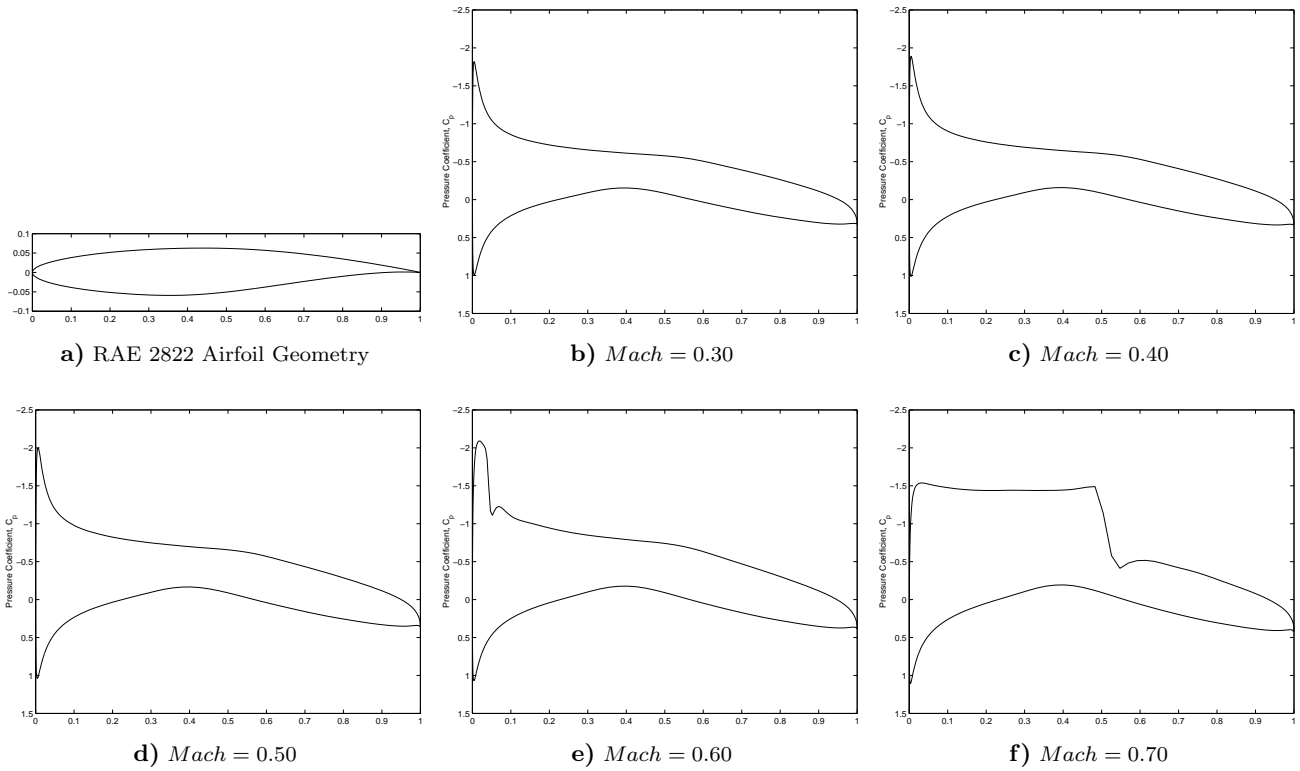


Fig. 1 Snapshots of the RAE 2822 Airfoil at  $\alpha = 3^\circ$  and Varying Mach Numbers.

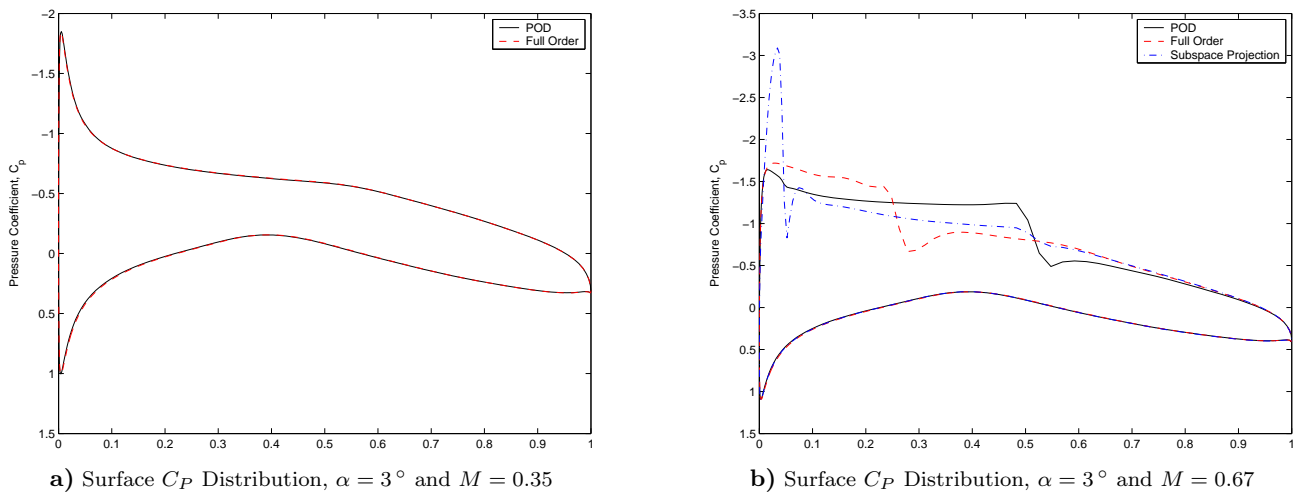
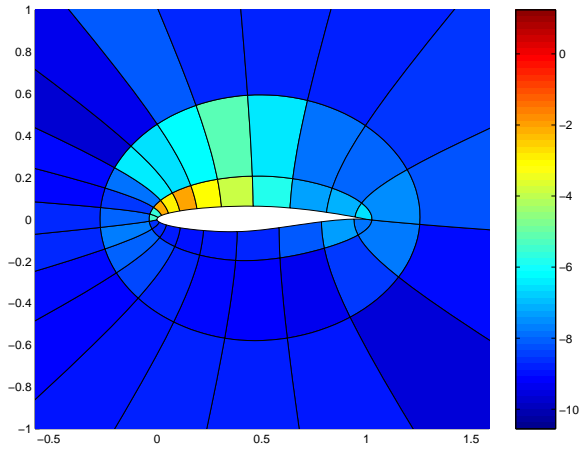
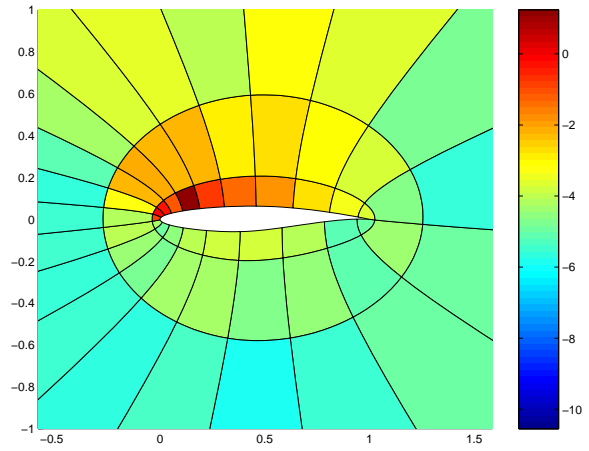


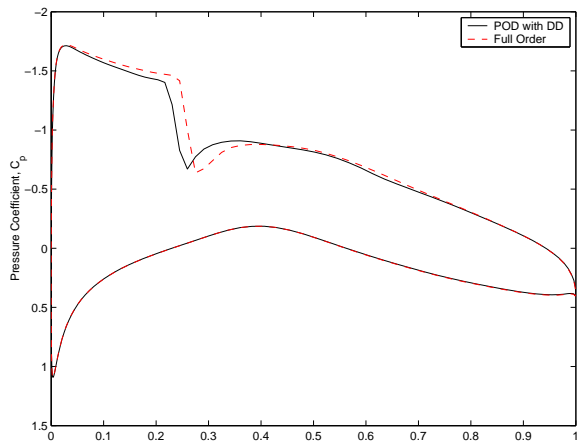
Fig. 2 POD solution for the RAE 2822 Airfoil at  $M = 0.35$  and  $M = 0.67$ ,  $\alpha = 3^\circ$ .



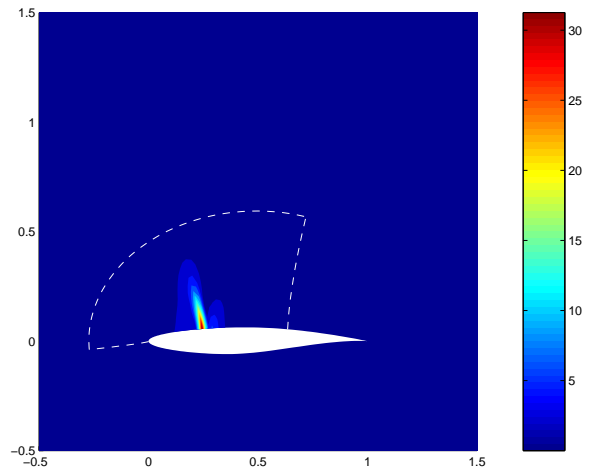
a) Error Estimation,  $M = 0.60$



b) Error Estimation,  $M = 0.67$

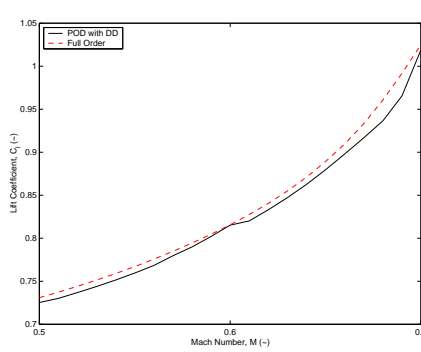


c) Surface  $C_P$  Distribution

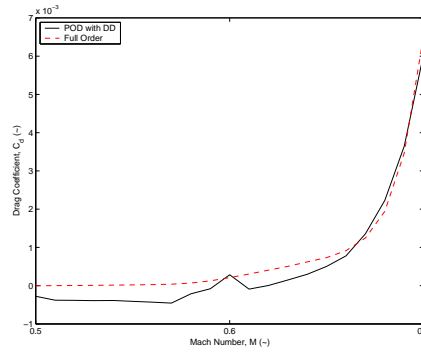


d) Percentage Error in  $C_P$  (Boundary of Domain Decomposition Denoted with Dashed Line)

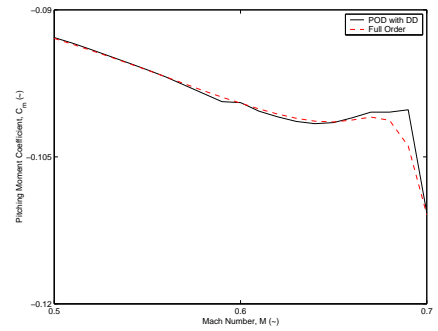
**Fig. 3** POD with Domain Decomposition Solution for the RAE 2822 Airfoil,  $M = 0.67$ ,  $\alpha = 3^\circ$ .



a) Lift Coefficient

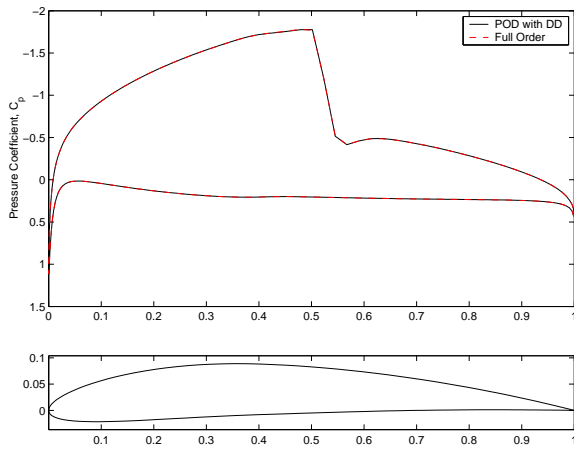


b) Drag Coefficient

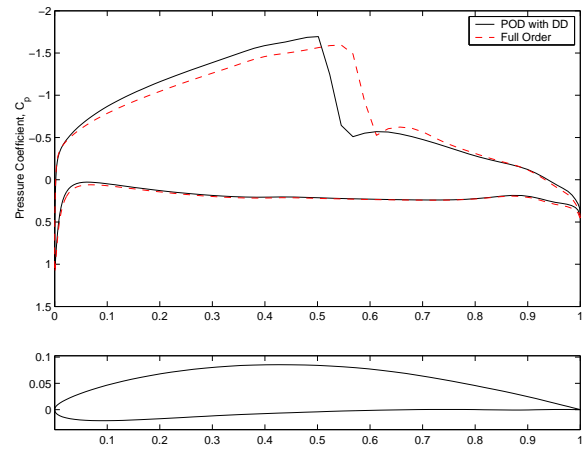


c) Pitching Moment Coefficient

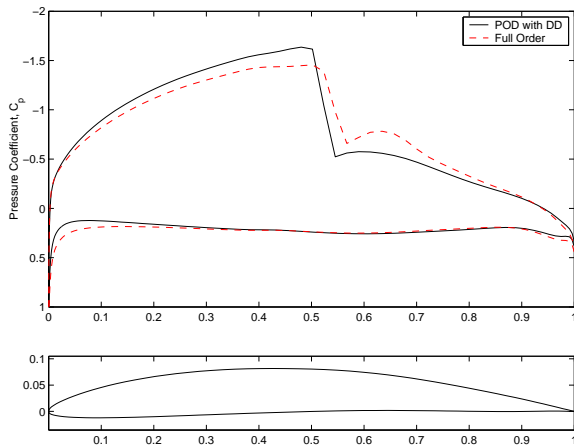
**Fig. 4** Mach Number Sweep for the RAE 2822 Airfoil,  $\alpha = 3^\circ$ .



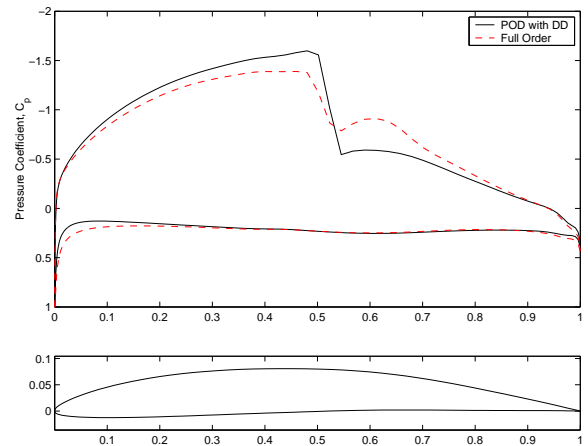
a) Initial Solution, POD with DD:  $C_l = 1.017$ ,  $C_d = 0.0115$ ; Full Order:  $C_l = 1.018$ ,  $C_d = 0.0115$ ;  $\alpha = 1.50^\circ$



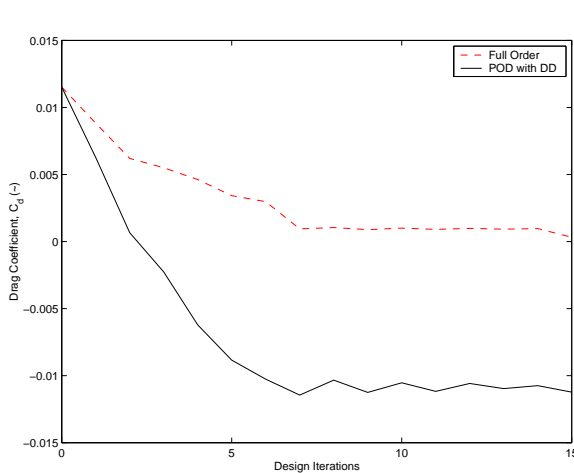
b) After 5 Design Iterations, POD with DD:  $C_l = 0.993$ ,  $C_d = -0.0088$ ; Full Order:  $C_l = 1.018$ ,  $C_d = 0.0034$ ;  $\alpha = 1.40^\circ$



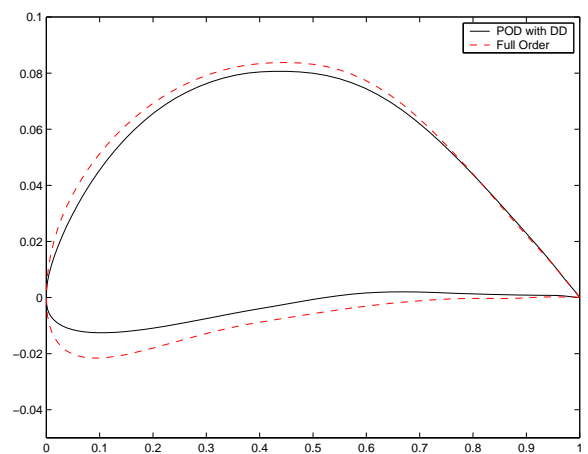
c) After 10 Design Iterations, POD with DD:  $C_l = 1.004$ ,  $C_d = -0.0105$ ; Full Order:  $C_l = 1.021$ ,  $C_d = 0.0010$ ;  $\alpha = 1.44^\circ$



d) After 15 Design Iterations, POD with DD:  $C_l = 1.003$ ,  $C_d = -0.0112$ ; Full Order:  $C_l = 1.012$ ,  $C_d = 0.0003$ ;  $\alpha = 1.42^\circ$

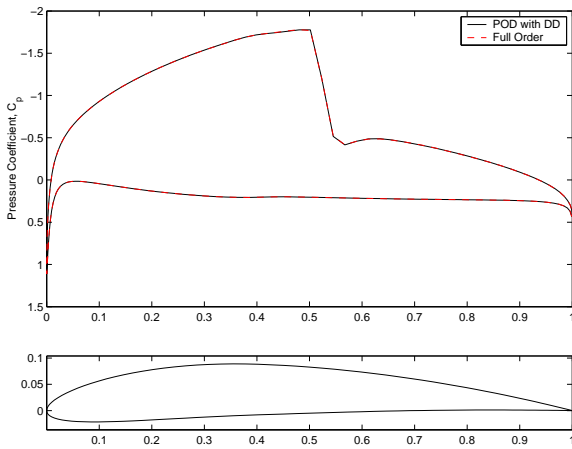


e) Drag Coefficient Design History

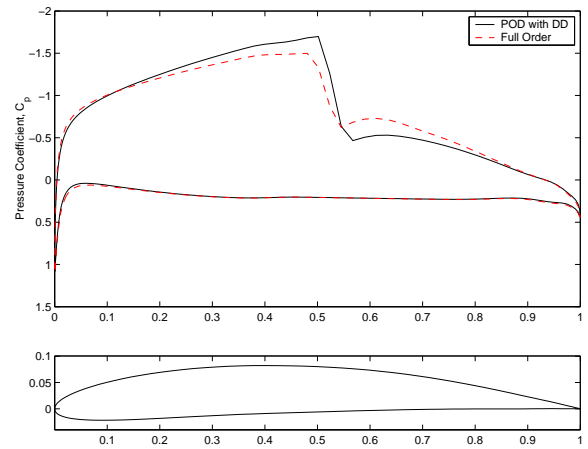


f) Comparison of Geometry for Approximate and True Solution

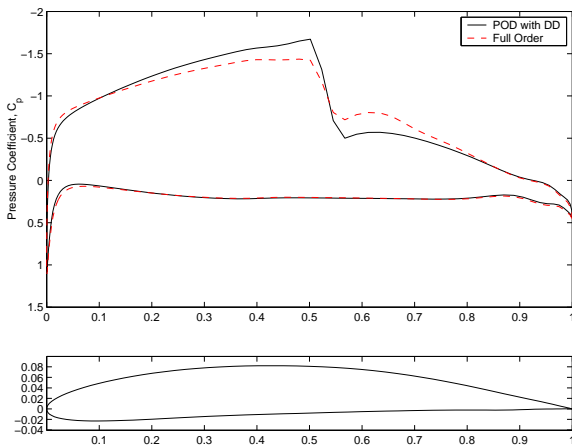
**Fig. 5 Drag Minimization for the NACA 4410 Airfoil using POD with Domain Decomposition and 25% of the Original Degrees of Freedom,  $M = 0.67$ .**



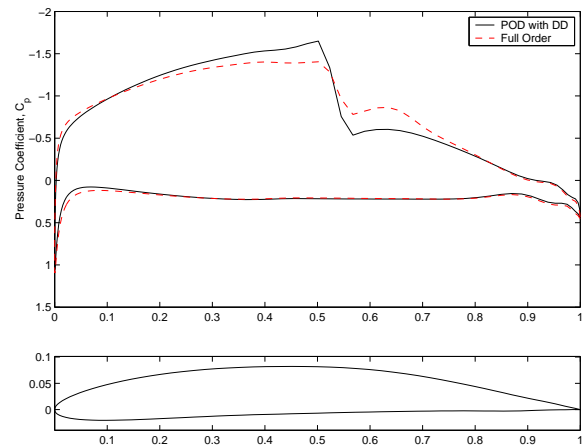
a) Initial Solution, POD with DD:  $C_l = 1.017$ ,  $C_d = 0.0115$ ; Full Order:  $C_l = 1.018$ ,  $C_d = 0.0115$ ;  $\alpha = 1.50^\circ$



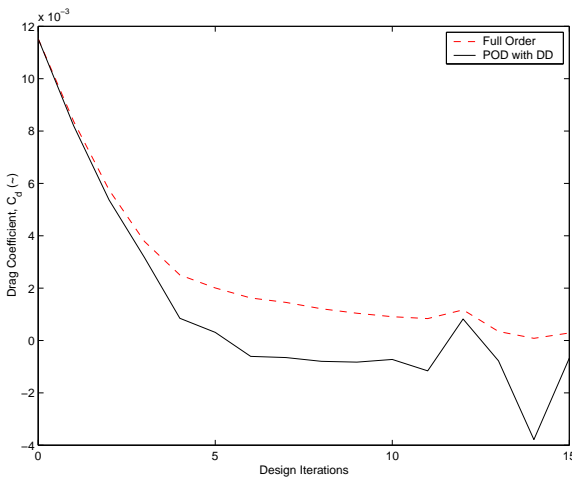
b) After 5 Design Iterations, POD with DD:  $C_l = 1.027$ ,  $C_d = 0.0003$ ; Full Order:  $C_l = 1.018$ ,  $C_d = 0.0020$ ;  $\alpha = 1.58^\circ$



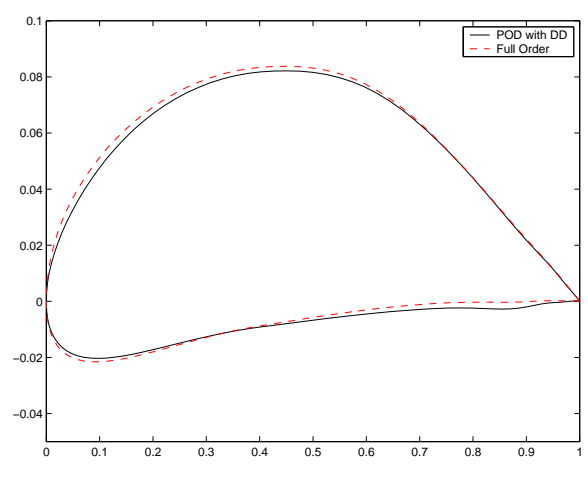
c) After 10 Design Iterations, POD with DD:  $C_l = 1.020$ ,  $C_d = -0.0007$ ; Full Order:  $C_l = 1.018$ ,  $C_d = 0.0009$ ;  $\alpha = 1.69^\circ$



d) After 15 Design Iterations, POD with DD:  $C_l = 1.023$ ,  $C_d = -0.0007$ ; Full Order:  $C_l = 1.035$ ,  $C_d = 0.0003$ ;  $\alpha = 1.78^\circ$



e) Drag Coefficient Design History



f) Comparison of Geometry for Approximate and True Solution

**Fig. 6 Drag Minimization for the NACA 4410 Airfoil using POD with Domain Decomposition and 50% of the Original Degrees of Freedom,  $M = 0.67$ .**