AIAA 2001–0926

Investigation of Non-Linear Projection for POD Based Reduced Order Models for Aerodynamics

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

39th AIAA Aerospace Sciences Meeting & Exhibit
January 8–11, 2001/Reno, NV

AIAA 2001–0926

Investigation of Non-Linear Projection for POD Based Reduced Order Models for Aerodynamics

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

In this paper a method for inviscid airfoil analysis and design optimization using reduced order models has been reviewed, modifications to the methodology are proposed, and new results of flow computations and design optimization are presented. The goal of these reduced order models is to obtain solutions to the Euler equations with sufficient accuracy for design optimization, at a computational cost which is far lower than that from traditional Computational Fluid Dynamics (CFD) methods. Design work with these models begins with a series of flow solutions (snapshots) in which the design variables of interest have been perturbed: this is the up-front cost of the design procedure. Proper Orthogonal Decomposition (POD) is used to compute the optimal linear basis for the snapshots. This linear basis can then be used to compute new, approximate flow solutions at significantly reduced computational cost for arbitrary geometries. Our previous work with these models had shown that while a significant reduction in computational cost was achieved the tendency of all flow solutions computed with the linear basis to be very similar to the original data significantly impaired their use in the design framework. Modifications to the procedure for computing the linear basis and the projection procedure for computing flow solutions are presented that eliminate this problem. Results of flow computations, examples of inverse design optimization results, and assessments of gradient accuracy are presented.

Nomenclature

E	total energy (internal plus kinetic)
$\mathbf{f}, \mathbf{g}$	Euler flux vectors
H	total enthalpy
I	cost function
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
λ	Lagrange multiplier, an eigenvalue
η_i	coefficient of the i-th mode in a function expansion
Ω	domain of interest
ρ	density
$\varphi^j(x)$	j-th POD basis mode
$\langle \cdot \rangle$	averaging operator
$\| \cdot \|$	L^2-norm

*Graduate Student, Student Member AIAA
†Assistant Professor, AIAA Member

Introduction

AERODYNAMIC Shape Optimization (ASO) using gradient based optimization methods typically requires gradients with respect to a large number of design variables. Traditional methods of computing these gradients, such as finite differencing, are computationally unfeasible for realistic problems. However, the computational cost can be significantly decreased through the use of an adjoint equation. The adjoint method yields the gradient with respect to an arbitrary number of design variables for the cost of a single flow solution and a single adjoint solution.[1–3] The application of the adjoint has produced remarkable improvements in the ability to design for certain types of optimal behavior, but it is not possible, in general, to treat arbitrary cost functions. Application in multidisciplinary design work is thus limited and often requires artificial constraints to account for other disciplines. For example, in the case of wing design, planform and thickness constraints have often been imposed so that structural weight, fuel volume, and takeoff/landing requirements would not be adversely effected by the results of the aerodynamic design optimization.

The focus in multidisciplinary design thus moves from the high fidelity modeling of individual disciplines to integrating several of these disciplines at the same level of fidelity. Such a level of collaboration requires

work in reorganizing the design optimization process, creating software integration environments, and enhanced optimization techniques. Low order approximations provide a means of reducing the tremendous computational cost associated with design optimization with these coupled high fidelity models. In a manner analogous to that of computing a flow calculation on a series of coarse to fine grids to accelerate convergence, optimization can be performed on a series of coarse to fine levels to accelerate convergence of the entire design process.

One technique for generating such low order approximations is the method of response surfaces. Response surfaces can be constructed from a database of solutions to yield a parameter of interest as a function of a series of chosen design variables. A polynomial surface is usually fitted to the values of the function of interest obtained by variations of the design variables. The resulting equation can easily and inexpensively be differentiated to produce gradients for use in design optimization. However, there is no basis for selecting the form of the equation used to fit the data. It is generally chosen so that it is easily differentiable with respect to the design variables and so that the number of unknown coefficients is as low as possible. Also, there is no way to tell a priori how accurately a particular solution of the response surface will agree with the exact solution.

Reduced order models based on Proper Orthogonal Decomposition (POD) seem to address our needs for a low order model that is more refined than response surfaces. Using a series of snapshots, which could represent a structural or aerodynamic solution, a linear basis is computed for describing the snapshots. The important property of this basis is that it is guaranteed to be the optimum linear basis to describe the snapshots provided. The basis can be used, with the governing equations of the system, to produce a small set of non-linear equations that can be efficiently solved using various least squares approximation methods. Rather than selecting a particular polynomial approximation to interpolate among data points, you have used the governing equations to provide some knowledge of the physical phenomenon that is being fitted.

Because the POD basis will not generally be able to represent the entire solution space, approximation will carry a certain amount of error. However, because we are representing the entire spatial domain of the solution an approximate solution will have an associated cost function which represents how well the solution satisfies the governing equations of the system. This cost function can be used to gauge the accuracy of the approximate solution, perhaps resulting in refining the basis modes with additional exact solutions if the existing basis is unable to adequately describe the desired solutions.

This paper addresses some of the problems associated with our previous work with the POD based models and describes preliminary work in developing a framework capable of implementing a multidisciplinary design environment. Examples of flow analysis are presented, accuracy of gradients for inverse design optimization are investigated, and an example of this procedure as applied to an airfoil inverse design problem is shown.

Proper Orthogonal Decomposition (POD)

The POD procedure has been used in a wide range of disciplines, including random variables, image processing, signal analysis, data compression, process identification and control in chemical engineering, and oceanography.[4] In fluid mechanics such uses as the modeling of synthetic jet actuators for flow control,[5] the unsteady aerodynamic and aeroelastic behavior of a transonic airfoil,[6] and flutter prediction for supersonic flow[7] have been reported. Details of the POD have been described in Ref.[4] 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 produced by 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 (tens of millions for a full aircraft Navier-Stokes flow calculation) to very small numbers (in the tens or hundreds). If the accuracy penalty of such a projection is small enough a description of the system can be generated at a significantly reduced cost. The POD provides 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 $\mathbf{u}(\mathbf{x})$ can be constructed from a basis $\{\varphi_j(x)\}_{j=1}^{\infty}$ as

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

To define this basis, assume that we have a set of N empirical solutions (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 implies that the basis functions $\varphi_j(x)$ must be the maximizers of the expression

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

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

given by

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

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

The solution of Eq. 2 has multiple maxima 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 (see Ref.[4,8,9]) require that the basis functions satisfy

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

The POD basis is therefore composed of the eigenfunctions, $\{\varphi_j\}$, of the integral Eq. 3. In the process of constructing basis modes for the discrete case, we will be dealing with computational 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

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

The eigenvalues have the 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. We will therefore typically sort 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. However, a procedure called the method of snapshots due to Sirovich[8] reduces the problem to an $M \times M$ problem where M is the number of ensemble functions.

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

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

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

$$\mathcal{R}a = \lambda a, \tag{6}$$

from which the POD basis functions can be calculated as

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

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

So far, our discussion has focused on ensembles of scalar functions. For a system of equations we could apply the methodology described to each variable and form a distinct basis mode for each variable.[9] 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 the vector state variable

$$\mathbf{v} = (\rho, \mathbf{u}, p),$$

where ρ, $\mathbf{u}$, and p are the density, Cartesian velocity components, and pressure respectively and the variables are assumed to have zero mean. The inner product is then computed as

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

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

Evaluating the inner product in Eq. 8 presents a problem when the domain of the l-th and m-th snapshots are not the same, as when the airfoil geometry is different for each snapshot. One approach is to use a common domain for every snapshot and apply transpiration boundary conditions to account for the change in the boundary.[7] For the structured meshes used to discretize the domain in our problems we have taken a modified approach.

The physical mesh can be transformed to a Cartesian mesh with unit cell areas and indices denoted by i and j. Furthermore, the state variables are taken to be constant in each cell. The discrete equivalent of the inner product in Eq. 8 is then

$$(\mathbf{v}^{(l)}, \mathbf{v}^{(m)}) = \sum_{i,j} \sum_{k=1}^{N} \bar{\mathbf{v}}_k^{(l)}(i, j) \mathbf{v}_k^{(m)}(i, j) A(i, j), \tag{9}$$

where $A(i, j)$ is taken to be the *ensemble average* of the i, j-th cell area as measured in the physical domain. In the case where the domain is the same for all snapshots, Eq. 9 is then equivalent to the continuous definition given in Eq. 8.

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

$$\mathbf{v}(x,y) = \sum_{i=1}^{M} \eta_i \varphi_i(x,y). \tag{10}$$

Now the flow around an arbitrary airfoil can be described, to a certain degree of accuracy, using the linear expansion given in Eq. 10. Because we will be using a linear expansion to represent a system governed by a set of non-linear equations, it is important to use a basis computed from a set of snapshots that as closely as possible resembles the solutions we will try to obtain. For this work, the snapshots were computed as a series of flow solutions using FLO82, the two-dimensional, cell-centered Euler solver by Jameson[10] for the analysis of airfoil geometries. Different flow solutions were obtained by parameterizing the airfoil shape and perturbing a baseline geometry.

Flow Analysis Procedure

Past uses of POD based expansions in fluid mechanics have primarily focused on the solution of the unsteady, incompressible Navier-Stokes equations. By utilizing a linear expansion for the unknowns a partial differential equation is reduced to a set of ordinary differential equations that are advanced in time. The boundary conditions are typically periodic, and because the modes themselves will be periodic, the boundary conditions are inherently satisfied. However, for our airfoil and analysis design purposes we are interested in modifying the geometry and hence modifying the boundary conditions. We therefore need the solution that best satisfies not only the governing equations on the domain interior but also satisfies the appropriate boundary conditions.

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

$$\frac{d}{dt}\iint_{\Omega} \mathbf{w}\, dx\, dy + \oint_{\partial\Omega} (\mathbf{f}\, dy - \mathbf{g}\, dx) = \mathbf{0}, \tag{11}$$

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

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

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. 11 can be applied independently to obtain a set of ordinary differential equations of the form

$$\frac{d}{dt}(\mathbf{w}_{ij}\, V_{ij}) + \mathbf{R}(\mathbf{w}_{ij}) = \mathbf{0}, \tag{12}$$

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. 11. In the steady state, the time derivative term drops out and we are left with

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

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. 10, several possibilities exist for exploiting the reduction in complexity of the problem. One strategy is to substitute the linear basis into Eq. 12 to produce a set of ordinary differential equations for the expansion coefficients. The system is advanced in time to a steady state to generate the steady solution. The number of equations for this system is unchanged from the number that would be obtained in a traditional CFD solver. Additionally, the computational cost of evaluating the flux integral is unchanged. The primary advantage achieved over a traditional CFD solver is the ability to advance the equations in time at a significantly greater rate.[7]

We have taken an alternate approach for solving the steady problem. An exact solution of Eq. 12 using the reduced basis and exactly satisfying all of the boundary conditions in the problem will typically not be possible since we have drastically reduced the number of available degrees of freedom. The expansion coefficients should therefore be computed in such a way that the governing equations are satisfied as closely as possible. We split the residual as

$$\mathbf{R}(\mathbf{w}_{ij}) = \mathbf{R}^{+}(\mathbf{w}_{ij}) + \mathbf{R}^{-}(\mathbf{w}_{ij}) = \mathbf{0},$$

where the $+$ and $-$ denote the positive and negative contributions to the residual in each cell. These correspond physically to the flux leaving and entering a

cell. Defining a new POD residual for the governing equations to be

$$\mathbf{R_{POD}}(\mathbf{w}_{ij}) = \mathbf{2} + \frac{\mathbf{R}^+(\mathbf{w}_{ij})}{\mathbf{R}^-(\mathbf{w}_{ij})} + \frac{\mathbf{R}^-(\mathbf{w}_{ij})}{\mathbf{R}^+(\mathbf{w}_{ij})} = \mathbf{0}, \quad (13)$$

non-dimensionalizes the residuals of continuity, momenta, and energy so that the residuals are given even weighting.

Using a linear expansion of the form of Eq. 10 the flow variables in Eq. 13 can be considered functions of the expansion coefficients, η_i, and the POD residual can be expressed as a non-linear function of each of the expansion coefficients

$$\mathbf{R_{POD}}(\eta_l) = \mathbf{0}, \;\; l = 1, \ldots, M. \quad (14)$$

As mentioned above, we have drastically decreased the number of degrees of freedom in the problem and an exact solution to Eq. 14 will usually not be possible. We define a POD cost function as

$$I_{POD} = \sum_n \mathbf{R_{POD}}^2(\eta_l), \quad (15)$$

where the summation, n, is over a given set of cells in the domain. This renders the problem well posed and defines the solution that for a given set of modes most closely satisfies the governing equations.

The summation in Eq. 15 should be over every cell in the domain if the governing equations are to be satisfied as closely as possible over the entire domain. However, for the types of problems we are considering in which the airfoil geometry is to be perturbed the effects of these changes will decay as the distance from the airfoil increases. For cells located a sufficient distance away from the body, the contribution to the cost function in Eq. 15 will be negligible regardless of the value of the expansion coefficients so there is little effect on the approximate flow solution of neglecting the equations for these cells. Typically, the flow solution is unchanged by including cells beyond the 20-30% closest to the body.

The flow solution procedure consists of finding the least squares minimizer of Eq. 15. In this work we have used the Levenberg-Marquardt method which combines elements of quasi-Newton and steepest descent. Solutions using this method for typical problems (15 to 30 modes) can be found in a few iterations, typically 5 to 10, with gradient information that is obtained analytically.

Given a set of basis modes the cost of an approximate flow solution is simply the cost of finding the least squares minmizer in Eq. 15. Each iteration requires evaluating the residual in no more than 30% of the cells, plus the cost of the analytic derivatives of the residual with respect to the expansion coefficients. Typically 10 iterations or less are required, so the total cost of evaluating the necessary residuals is equivalent to 3 evaluations of the residual over the entire domain. The cost of the derivatives is proportional to the number of modes, M, and is typically about 15 for airfoil design problems. Assuming each derivative has the same computational cost as a single residual the cost of the approximate flow solution from a set of basis modes is equivalent to about *50* evaluations of the residual over the entire domain. By comparison, the FLO82 flow solver, using a 5-stage Runge-Kutta scheme and multigrid, requires the equivalent of approximately *1,000* such residual evaluations when artificial dissipation evaluations are also considered. An order of magnitude decrease in computational cost is achieved for two-dimensional flows, and the reduction in three-dimensions is even greater.

Implementation Issues

The successful implementation of a multidisciplinary design environment depends not just on the algorithms for aerodynamics, structures, mission performance, etc. that comprise the aircraft or spacecraft model but also on the ability to provide an interface between existing codes for analysis of these various design disciplines. These codes exist in a number of different programming languages and often have unique interfaces and data structures. For example, an aeroelastic optimization code might utilize an existing CFD code written in Fortran for the aerodynamic analysis, Nastran for the finite element analysis of the structure, an optimization code written in C, and a data visualization program for analyzing the results.

We do not wish to duplicate the efforts that produced existing analysis codes, rather, to generate a common interface so that data can be exchanged between programs. If such a uniform interface existed, we could treat the various CFD solvers, structural analysis, and optimization codes as modules which can be easily combined, substituted, and exchanged. To implement our design environment, we use the Python language as a top level interface to the various subroutines, codes, and visualization programs. Python is an object-oriented language similar to Java and runs on Unix/Linux, Windows, and Mac platforms.

The Python language itself is written in C, and extending Python to use code written in C and C++ is quite straight forward. Using automatic interface generators such as Pyfort[11] and f2py,[12] interfaces to Fortran programs can be created with no knowledge of the coding details necessary to move data between Python and Fortran. These interfaces allow us to retain all of the speed advantages of compiled C/C++ or Fortran code over interpreted languages such as Java or Python, which are typically substantially slower. Also, interfaces to the engines of Matlab and the Visualization Toolkit (VTK) for plotting and data visualization are available.

Although these are primarily computer science and

information transfer/storage issues, they are equally important to the successful implementation of a multidisciplinary design environment. The work presented in this paper utilized the existing FLO82 code written in Fortran, for CFD results. The POD algorithms were also written in Fortran. Using Pyfort, interfaces were generated to facilitate communication at the Python level. For plotting, the PyMat[13] interface between Python and Matlab was used.

Results

In this paper, POD based reduced order models are used in a design environment for evaluating cost functions, performing line searches, and finite differenced to compute derivatives of a cost function with respect to the design variables. An airfoil inverse design problem in which a specified pressure is achieved by varying the geometry is carried out using POD models. Although this problem can be efficiently treated using an adjoint formulation, we use it as a representative model problem for more complicated aerodynamic and multidisciplinary problems. The full value of POD-based reduced order models will be obtained when the approach is used to model problems or disciplines which have traditionally used polynomial representations of their cost functions.

Flow Computations

For purposes of generating snapshots of varying airfoil geometry, we make modifications to the geometry using a series of Hicks-Henne sine bump functions,[14] which make smooth changes to the geometry. A series of snapshot geometries were defined by adding 14 bump functions, 7 each on the upper and lower surfaces, to the RAE 2822 airfoil as shown in Fig. 1a. Flow solutions for the original airfoil plus the 14 modified airfoils were computed using FLO82 and a basis with a total of 15 modes was computed.

Using the previously described flow computation procedure the solution for the RAE 2822 airfoil was computed. Because any of the snapshots used to construct the basis can be exactly represented by the basis we should recover the exact solution. In Fig. 1b-d the results for pressure are shown. As expected the exact solution was recovered.

Results for the Korn airfoil, shown along with the snapshots used to construct the basis modes in Fig. 1e, are shown in Fig. 1f-h. Because the solution for the Korn airfoil was not part of the snapshots, we don't expect to get an exact solution. However, the geometries are quite similar and we expect the solution to be similar to that for the RAE 2822. As seen in Fig. 1f the surface C_P is quite good, except for a small region on the forward upper surface.

More substantial changes in geometry are investigated in Fig. 2. For the NACA 4410 airfoil, shown in Fig. 2a with the RAE based snapshots, there is a substantial change in geometry from the data the basis modes are based on. However, the results shown in Fig. 2b-d show the results are quite reasonable for our purposes of generating approximate solutions for guiding the design process.

A solution for the NACA 1413 airfoil, shown with the snapshots in Fig. 2e, is another example of a substantial change in geometry from the original snapshots. The resulting pressure distribution, shown in Fig. 2f, has a general trend that is correct but is not really close enough for design work. This illustrates a potential problem with using such a limited number of empirical basis modes. Even though a set of modes may be able to represent a wide range of different flow computations, we can not expect all possible results to be approximated with sufficient accuracy. However, we would like to utilize results that are sufficiently accurate while disregarding those that are not. This can be accomplished using the POD cost function defined in Eq. 15, which as a measure of how well the approximate flow solution satisfies the governing equations, can be used as an indicator of the accuracy of the surface pressure distribution, lift coefficient, drag coefficient, etc that we will be interested in for design purposes. For example, the flow solution for the Korn airfoil shown in Fig. 1e-h has a POD cost function, $I_{POD} = 1.285$, and the flow solution for the NACA 4410 in Fig. 2a-d has a cost function of $I_{POD} = 0.358$. By comparison, the flow solution for the NACA 1413 in Fig. 2e-h has a cost function of $I_{POD} = 6.214$.

It was stated while describing the flow computation procedure that the least squares fit to the governing equations does not need to take into account every cell in the domain for the types of problems we have considered. Equations for the cells at the far-field and airfoil surface boundary are always used since this is how the boundary conditions are incorporated. Fig. 3 illustrates the effect on the aerodynamic coefficients of incorporating the equations for varying numbers of cells in the POD cost function in Eq. 15. For the RAE 2822, the results are essentially independent of how many equations are used. For the other airfoils the lift and drag vary somewhat over the first 20-30%, beyond which the use of additional equations has no effect on the computed solution. The error present when every cell in the domain is considered results not from neglecting any equations, but the inability of the basis modes to represent the true flow solution.

A portion of the basis modes that represents pressure are shown in Fig. 4. The ensemble pressure average in Fig. 4a is almost identical to the pressure for the original RAE 2822 airfoil in Fig. 1c. The modes then represent the change in pressure due to the perturbations present in the various snapshots. The first 4 modes, in Fig. 4b-e, primarily effect the pressure distribution on the upper surface only. This means that changes in geometry produce more significant changes

in the pressure on the upper surface as compared to the lower surface. Subsequent modes, with smaller eigenvalues, account for less of the variation in pressure as the geometry is modified and tend to affect primarily the pressure on the lower surface as seen in Fig. 4f-g. The complete eigenvalue distribution for the modes is shown in Fig. 4h.

Inverse Design and Gradient Accuracy

The suitability of gradients computed from POD based models was assessed by performing an inverse design procedure where a target pressure distribution is specified and the geometry is modified to most closely achieve such a distribution. Since a given pressure distribution may not be achievable an inverse design cost function is defined as

$$I_{ID} = \int_S (p - p_T)^2 ds, \tag{16}$$

where p_T is the target or specified pressured distribution and the integral extends over the complete surface of the airfoil, S.

The same set of 15 basis modes based on the RAE 2822 were used for the inverse design problem. The RAE 2822 was used as the initial geometry for the design with the pressure distribution for the NACA 4410 specified as the target pressure. A parameterization of the airfoil geometry was made using a total of 20 bump functions, 10 each on the upper and lower surface, and bunched towards the leading and trailing edges. Modifications to the geometry are made by varying the amplitude of the bumps which are the design variables of the problem. The initial and target geometry and pressure are shown in Fig. 5a-b. Gradients of the inverse design cost function with respect to the design variables were computed by finite differencing of the POD model to define a direction in which to make changes. A line search was used to determine the appropriate step size at each iteration.

The airfoil geometry and pressure distribution at design iterations 5, 10, and 15 are shown in the rest of Fig. 5. Referring to the cost function convergence in Fig. 6e, the cost function has been reduced approximately 2 orders of magnitude after 15 design iterations. If we follow the dashed line beyond 15 iterations we see that the cost function is decreased very slightly. This is because the basis modes based on the original RAE 2822 airfoil are no longer providing sufficiently accurate gradients. The only way to make further reductions in the cost function is to somehow improve the accuracy of the gradients.

A summary of the gradient accuracy for four of the design variables is shown in Fig. 7, where the exact gradients were computed by the complex step method for comparison purposes. The gradient errors originate from two sources - the approximation to the function from the POD model and the finite difference approximation to the derivative. Regardless, the errors are initially in the range of 20% or less for the step size used in the inverse design procedure (10^{-3}) which is quite sufficient for the steepest descent/line search method used to optimize the geometry. At subsequent iterations the gradient errors are increasing as the geometry is modified from the original RAE 2822 airfoil that the basis modes were based on. This was reflected in the decreasing slope of the cost function convergence and eventual inability to decrease the cost function when the gradient accuracy decreased too much. To make further modifications to the geometry, more accurate gradients must be obtained.

A new set of modes, using flow solutions based about the best currently available geometry, was used to improve the accuracy of the computed gradients. Using the airfoil geometry at design iteration 15 as a baseline, 14 bump functions, 7 each on the upper and lower surfaces, were used to define the geometries for 15 new snapshots. Flow solutions for the baseline airfoil and the 14 modified airfoils were computed using FLO82 and used to form a new set of basis modes. The inverse design procedure continued from design iteration 15 using the new set of modes. Airfoil geometries and pressure distributions for design iteration 20 and 25 are shown in Fig. 6a-d. The cost function convergence using the updated set of modes is shown by the dashed line in Fig. 6e.

Conclusions

A review of our past work of using POD reduced order models for aerodynamics was presented and modifications were presented to improve the usefulness of the models. These models reduce the cost of a flow computation to the solution of a non-linear set of equations which can be solved using non-linear least squares solvers.

To test the use of these types of models for design problems, a simple airfoil inverse design problem was solved. Although this type of problem can be solved by adjoint formulations, it is a representative model problem for the types of more complex systems we would like to apply this to in the future. An initial model based on the starting airfoil geometry was used for early iterations until the accuracy of the gradients had diminished. A second reduced order model was computed based on the best available solution to use for further iterations. Although we do not expect to obtain exact results using this method, the results show a significant reduction in computational cost can be achieved, with a quite small degradation in accuracy.

Future work will address improving the accuracy of the POD models for providing gradients. The use of a complex step for the derivative essentially eliminates the error due to the finite difference approximation. Also, modified POD procedures may be used to include derivatives with respect to a series of design

parameters. This would extend the usefulness of a given set of modes and decrease the need to compute additional flow solutions and modes.

Ultimately this work will be extended to 3-D aerodynamics and viscous flows. Applying POD to additional disciplines such as structures, performance, and the entire vehicle model will then allow new design optimization goals to be considered that are currently computationally unfeasible.

References

[1]A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.

[2]A. Jameson and J.J. Alonso. Automatic aerodynamic optimization on distributed memory architectures. *AIAA paper 96-0409*, AIAA 34th Aerospace Sciences Meeting, Reno, Nevada, January 1996.

[3]J. Reuther, A. Jameson, J.J. Alonso, M.J. Rimlinger, and D. Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers. *AIAA paper 97-0103*, AIAA 35th Aerospace Sciences Meeting, Reno, Nevada, January 1997.

[4]P. Holmes, J.L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, 1998.

[5]O.K. Rediniotis, J. Ko, X. Yue, and A.J. Kurdila. Synthetic jets, their reduced order modeling and applications to flow control. *AIAA paper 99-1000*, AIAA 37th Aerospace Sciences Meeting, Reno, Nevada, January 1999.

[6]K.C. Hall, J.P. Thomas, and E.H. Dowell. Reduced order-modelling of unsteady small-disturbance flows using a frequency-domain proper orthogonal decomposition technique. *AIAA paper 99-0655*, AIAA 37th Aerospace Sciences Meeting, Reno, Nevada, January 1999.

[7]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.

[8]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.

[9]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.

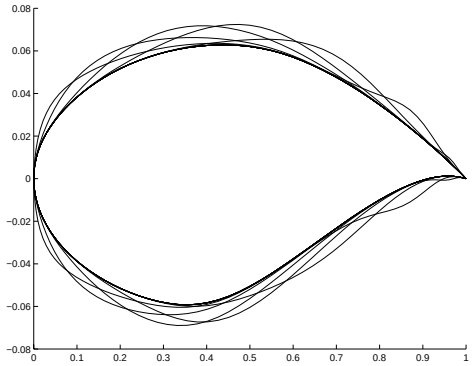
[10]A. Jameson. Multigrid algorithms for compressible flow calculations. In W. Hackbusch and U. Trottenberg, editors, *Lecture Notes in Mathematics, Vol. 1228*, pages 166–201. Proceedings of the 2nd European Conference on Multigrid Methods, Cologne, 1985, Springer-Verlag, 1986.

[11]P.F. Dubois. *Pyfort Reference Manual*. Program for Climate Model Diagnosis and Intercomparison, Lawrence Livermore National Laboratory, August 2000. http://sourceforge.net/projects/pyfortran.

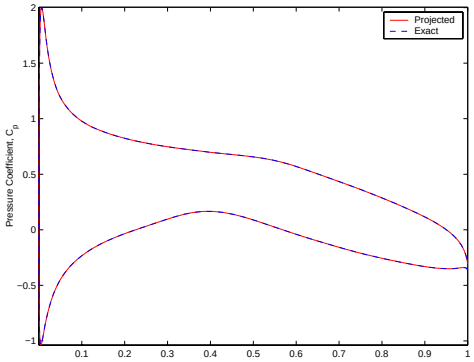
[12]P. Peterson. *f2py Fortran to Python Interface Generator*, December 2000. http://cens.ioc.ee/projects/f2py2e/.

[13]A. Sterian. PyMat. http://claymore.engineer.gvsu.edu/ steriana/Python/pymat.html.

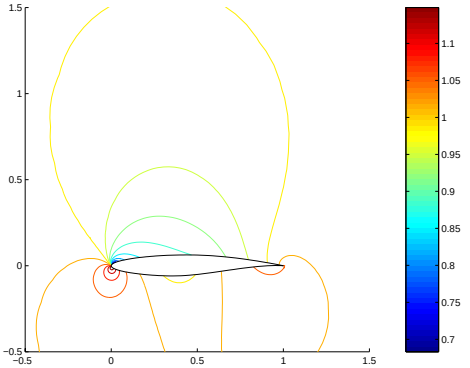
[14]R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15:407–412, 1978.



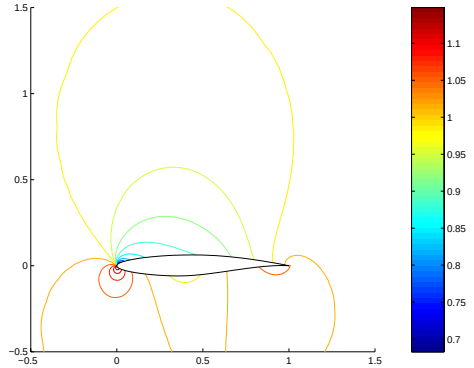
a) Snapshots Based on RAE 2822



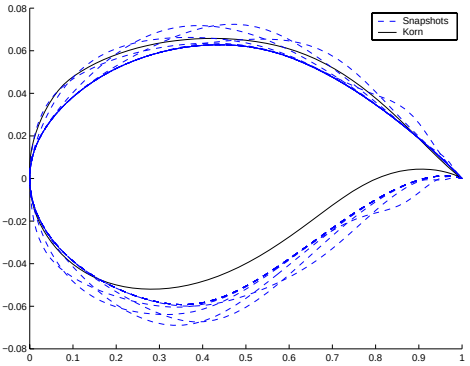
b) C_P Comparison for RAE 2822



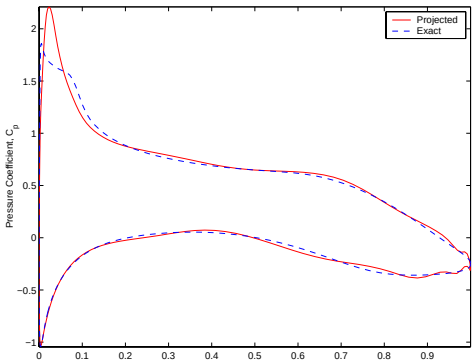
c) Exact Pressure for RAE 2822



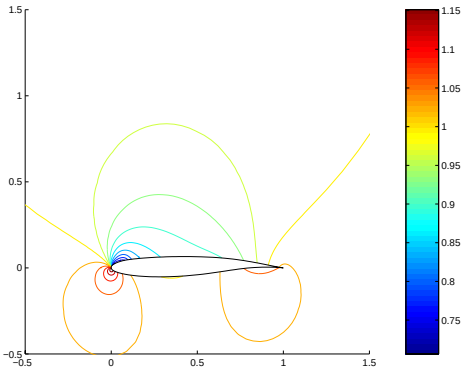
d) Projected Pressure for RAE 2822



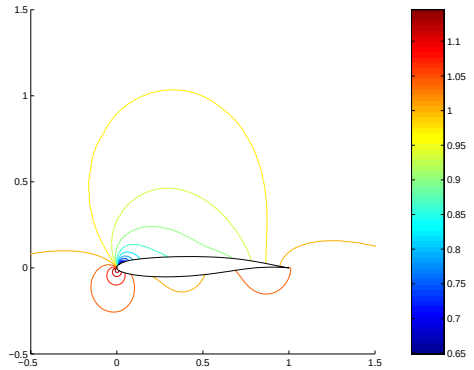
e) Snapshots and Korn Airfoil



f) C_P Comparison for Korn

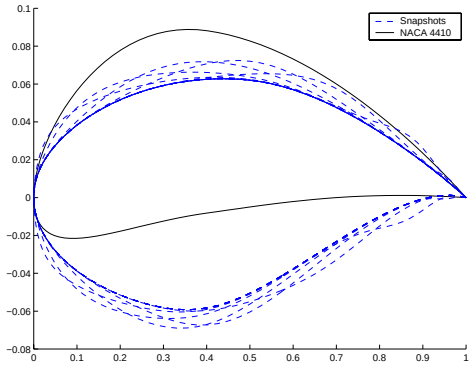


g) Exact Pressure for Korn

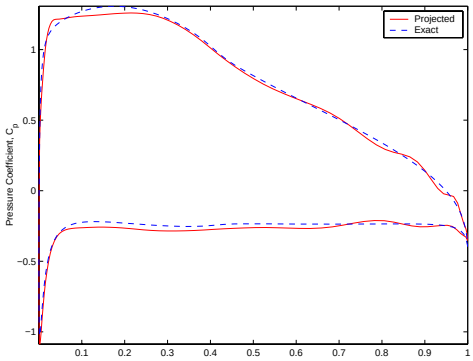


h) Projected Pressure for Korn

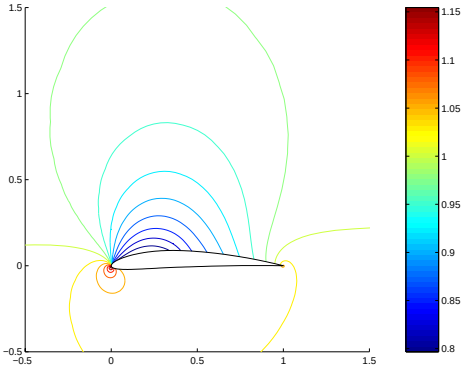
Fig. 1 Projections Using Modes Based on RAE 2822 Airfoil, $M = 0.50$**.**



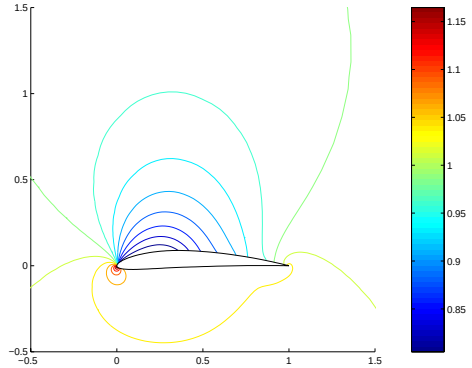
a) Snapshots and NACA 4410 Airfoil



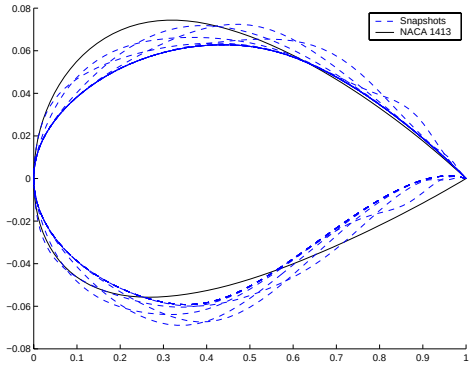
b) C_P Comparison for NACA 4410



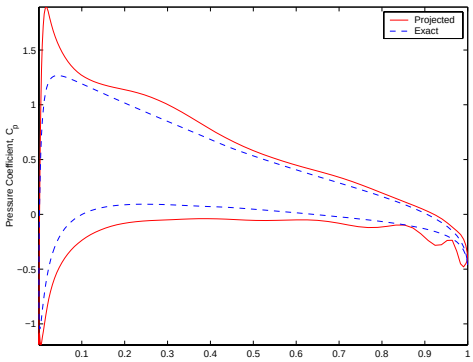
c) Exact Pressure for NACA 4410



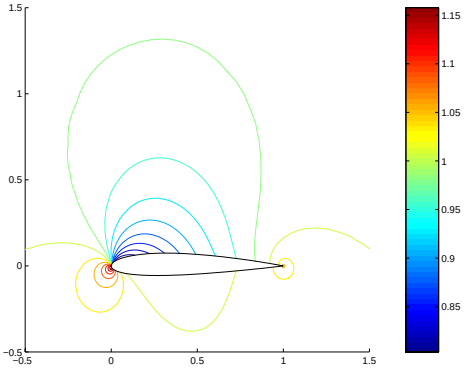
d) Projected Pressure for NACA 4410



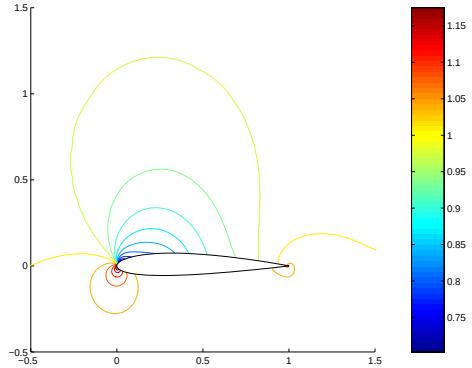
e) Snapshots and NACA 1413 Airfoil



f) C_P Comparison for NACA 1413

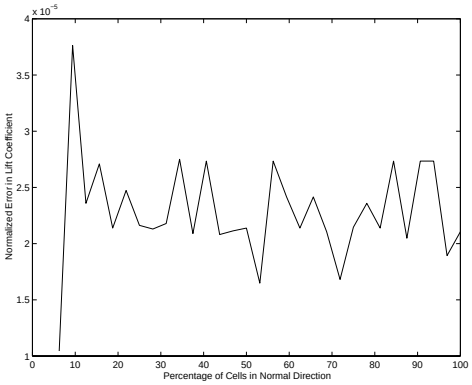


g) Exact Pressure for NACA 1413

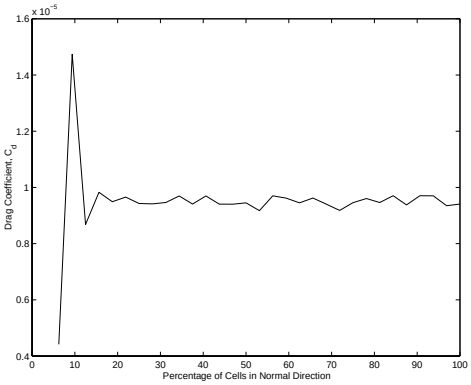


h) Projected Pressure for NACA 1413

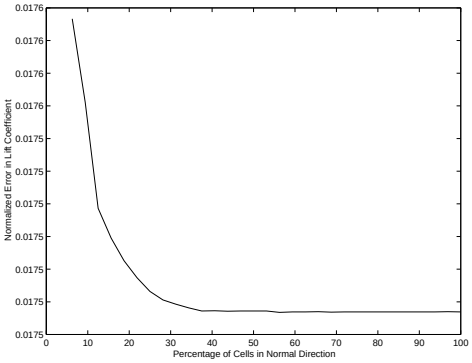
Fig. 2 Projections Using Modes Based on RAE 2822 Airfoil, $M = 0.50$**.**



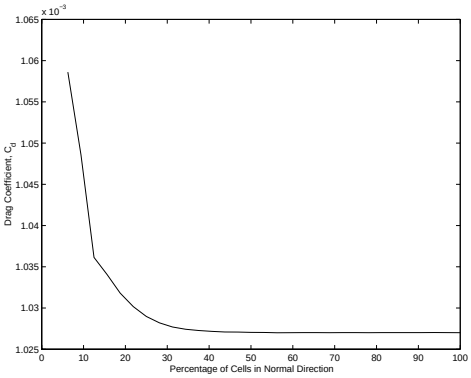
a) Normalized C_l Error for RAE 2822



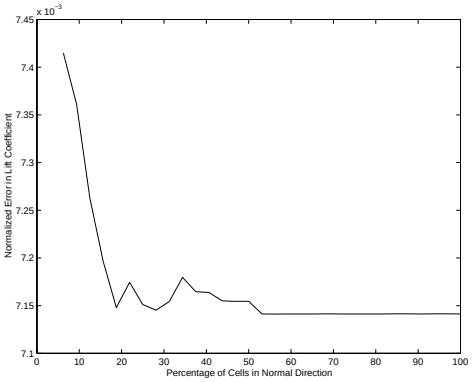
b) C_d for RAE 2822



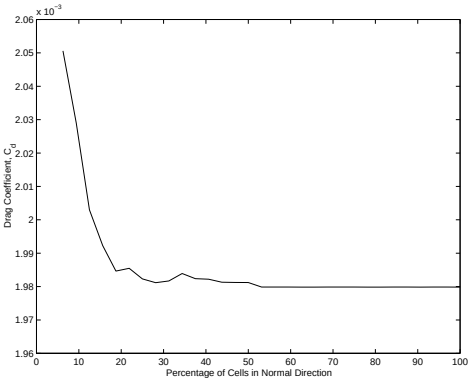
c) Normalized C_l Error for Korn



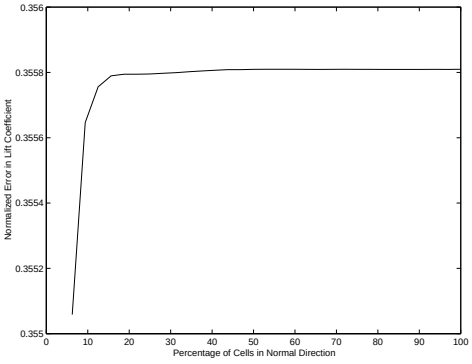
d) C_d for Korn



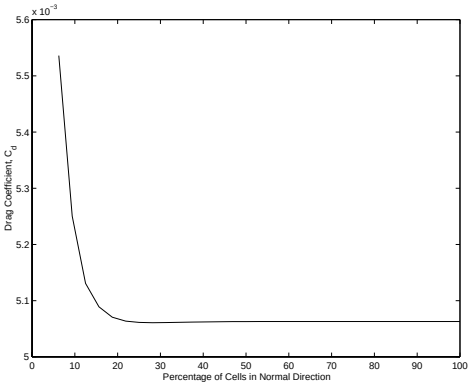
e) Normalized C_l Error for NACA 4410



f) C_d for NACA 4410

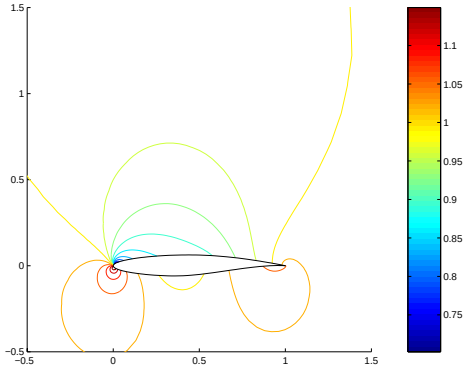


g) Normalized C_l Error for NACA 1413

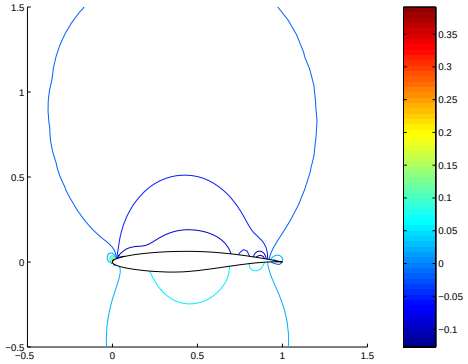


h) C_d for NACA 1413

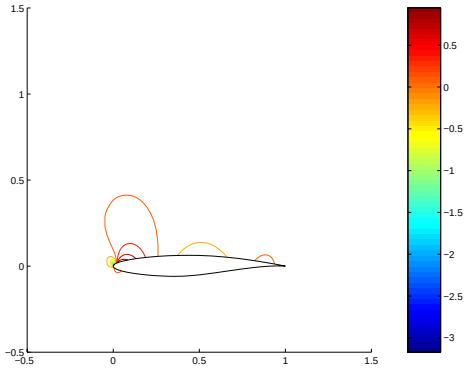
Fig. 3 Sensitivity of C_l and C_d to Percentage of Cells Used in Projection, $M = 0.50$.



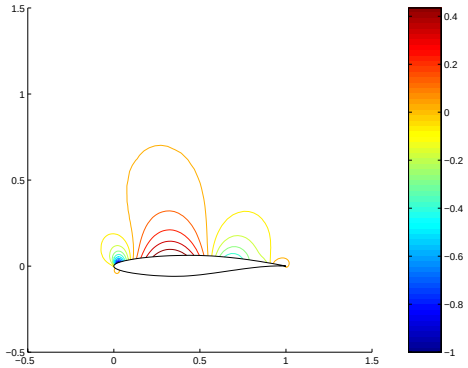
a) Ensemble Pressure Average



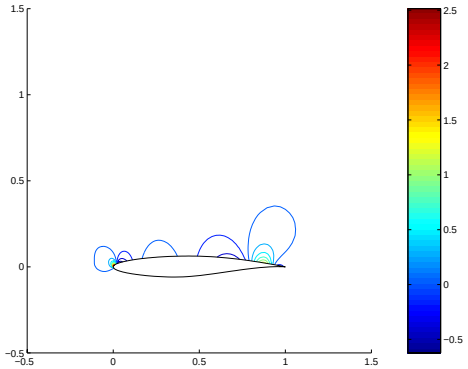
b) Pressure Mode 1



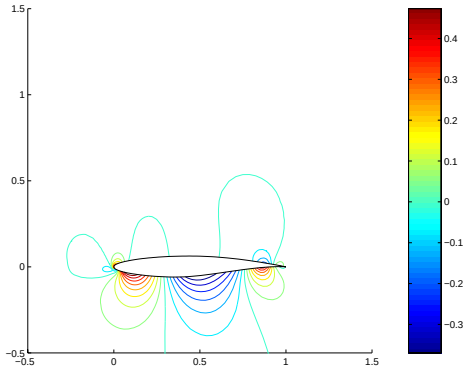
c) Pressure Mode 2



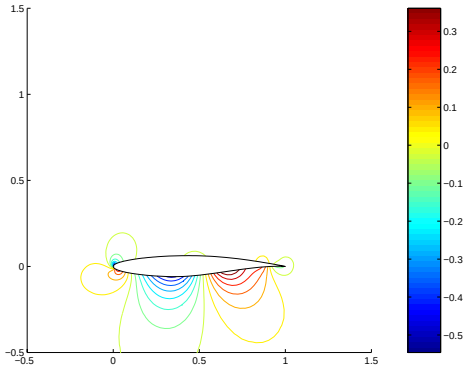
d) Pressure Mode 3



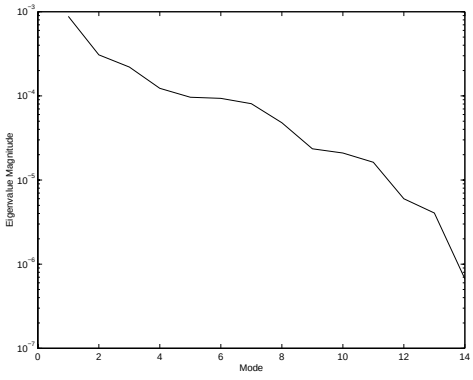
e) Pressure Mode 4



f) Pressure Mode 5

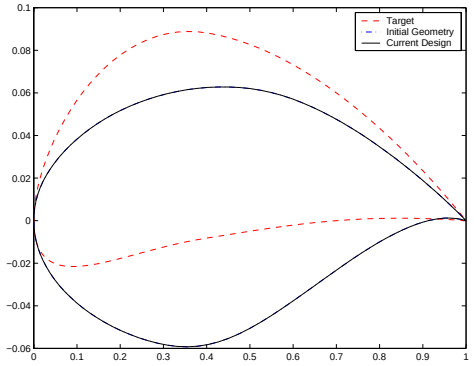


g) Pressure Mode 6

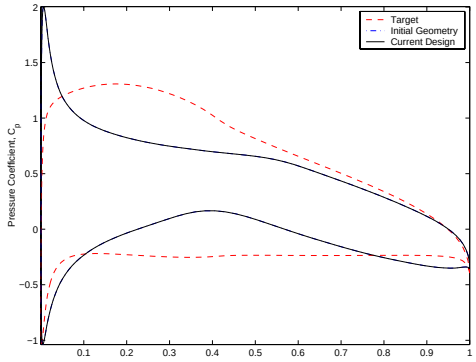


h) Eigenvalue Distribution

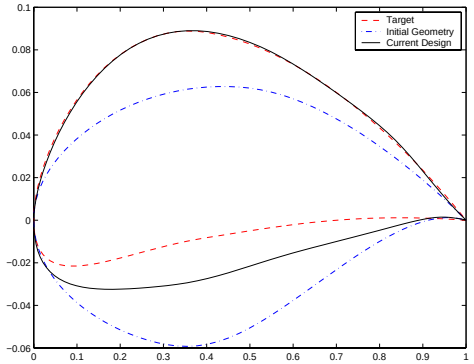
Fig. 4 Pressure Modes for Snapshots Based on RAE 2822 Airfoil, $M = 0.50$**.**



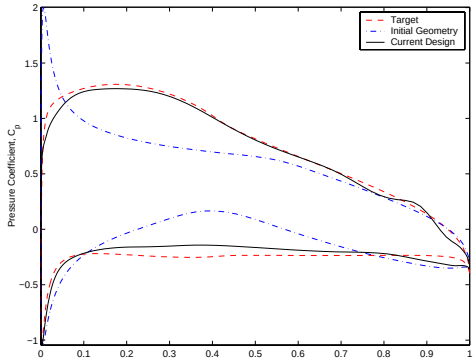
a) Geometry at Design Iteration 0



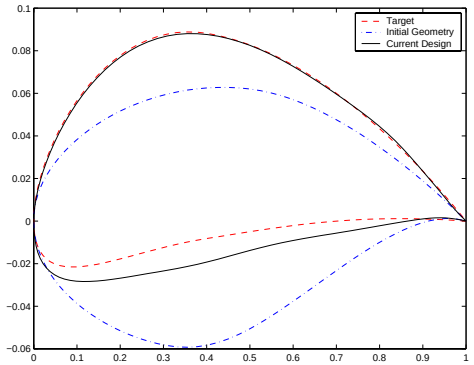
b) Pressure at Design Iteration 0



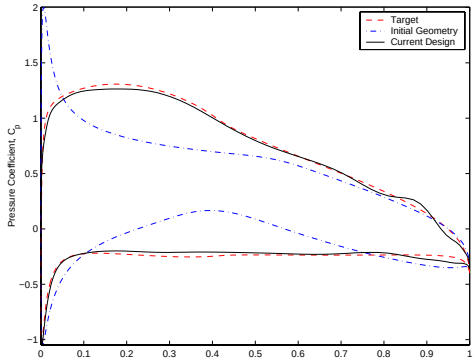
c) Geometry at Design Iteration 5



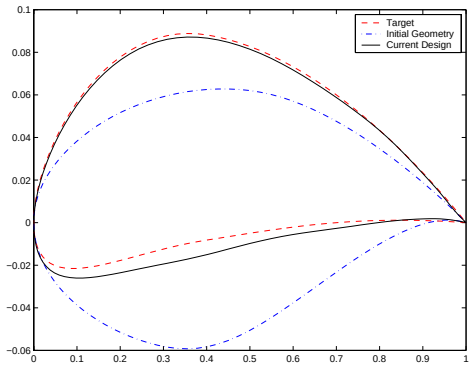
d) Pressure at Design Iteration 5



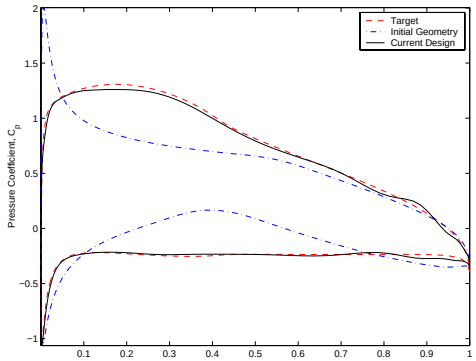
e) Geometry at Design Iteration 10



f) Pressure at Design Iteration 10

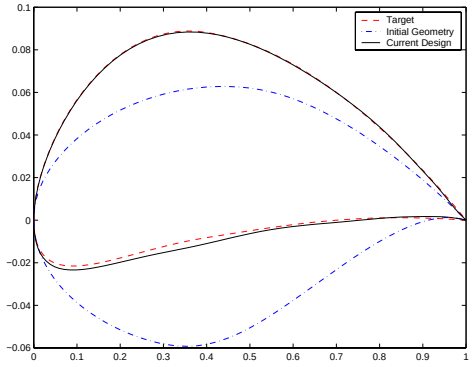


g) Geometry at Design Iteration 15

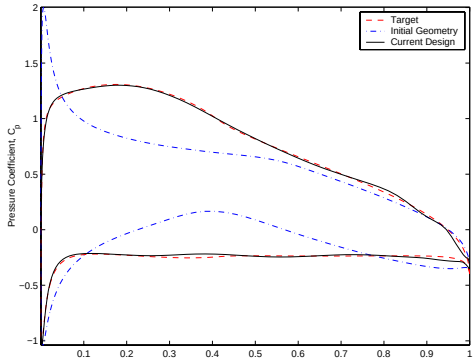


h) Pressure at Design Iteration 15

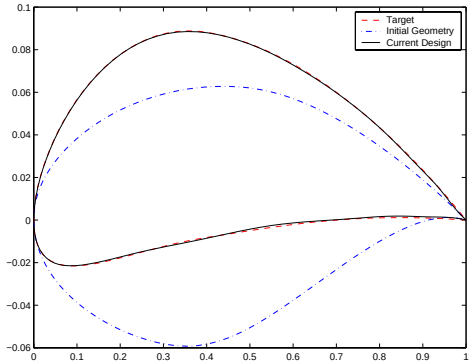
Fig. 5 Inverse Design Iterations for NACA 4410 Target Pressure, $M = 0.50$**.**



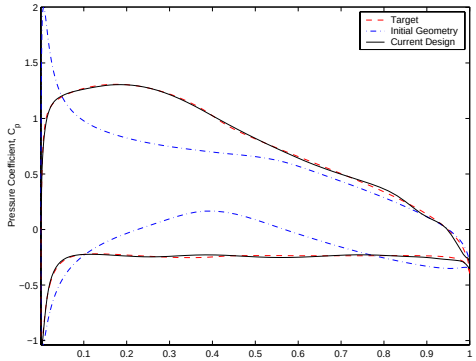
a) Geometry at Design Iteration 20



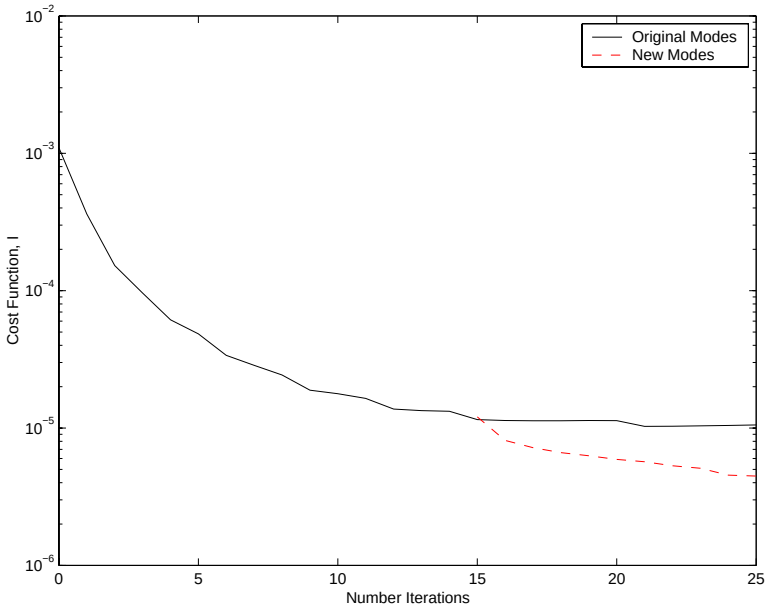
b) Pressure at Design Iteration 20



c) Geometry at Design Iteration 25

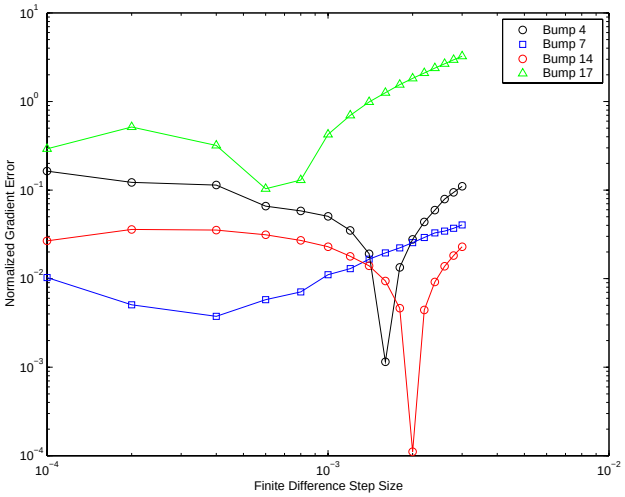


d) Pressure at Design Iteration 25

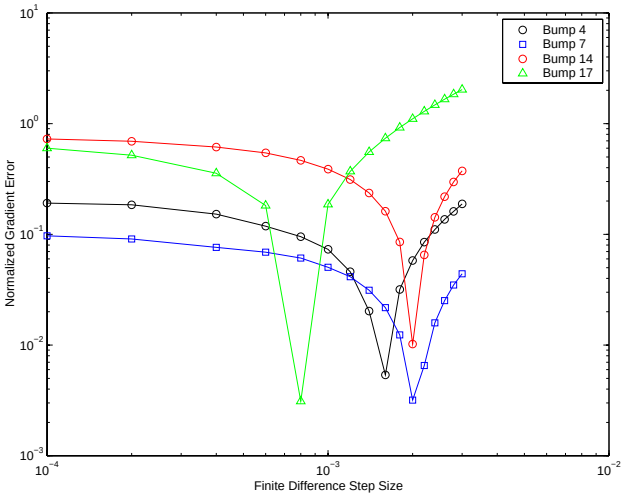


e) Cost Function Convergence

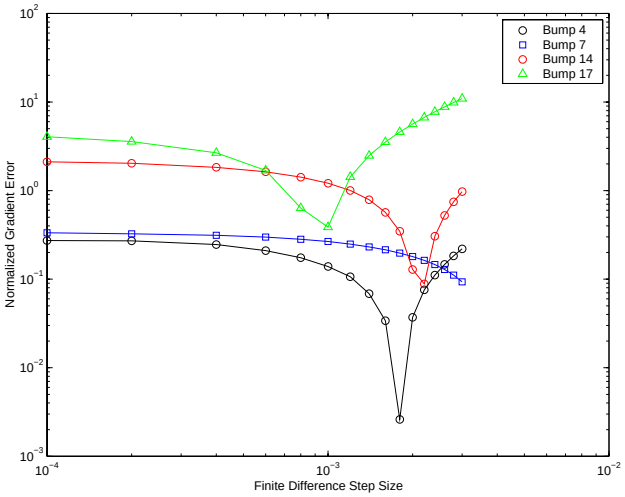
Fig. 6 Inverse Design Iterations and Convergence for NACA 4410 Target Pressure, $M = 0.50$**.**



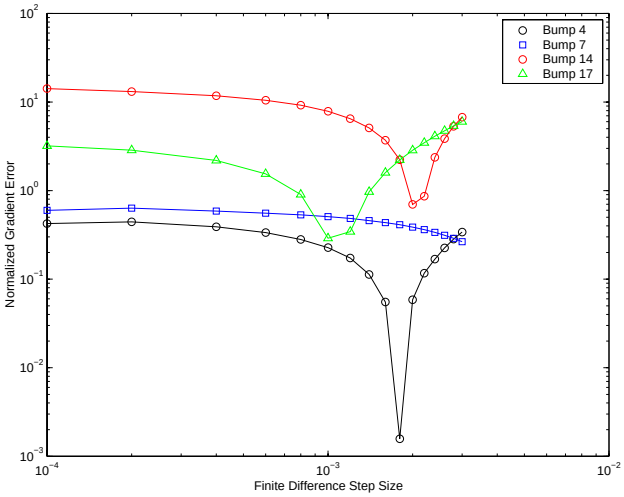
a) Gradient Error at Design Iteration 0



b) Gradient Error at Design Iteration 5



c) Gradient Error at Design Iteration 10



d) Gradient Error at Design Iteration 15

Fig. 7 Gradient Analysis for NACA 4410 Target Pressure, $M = 0.50$**.**