AIAA 2004-4465

Improving the Performance of Design Decomposition Methods with POD

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

10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference
August 30 - September 1, 2004/Albany, NY

Improving the Performance of Design Decomposition Methods with POD

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

The use of decomposition methods for multidisciplinary design offer many advantages such as ease of implementation and scalability with increasing numbers of disciplines, but tend to be less computationally efficient than tightly coupled methods. However, by reducing the bandwidth of the interaction between the modules in a decomposition scheme significant gains in computational speed can be achieved. This has been recognized in methods such as Collaborative Optimization (CO), for example, where disciplines don't interact directly with the system level problem but through response surfaces or spline fits. In our past research we have found reduced order models based on Proper Orthogonal Decomposition (POD) to be of some practical use for modeling aerodynamics, but with useful mathematical properties such as incorporating the governing equations of the system into the approximation and its convergence in the limit of a large number of observations of the system. In this work we have implemented POD as a method of reducing the coupling bandwidth between disciplines in a decomposition method called Bi-Level Integrated System Synthesis (BLISS). By using POD the normally high bandwidth interaction between some disciplines, such as aerodynamics and structures where all of the surface pressures and structural displacements need to be exchanged, is replaced by a one time exchange of modes and a per iteration exchange of values that scale the modes. Results from applying this procedure to an aerodynamic panel code and beam structural model have shown that the results using the decomposition method with POD are identical to those computed using a Multidisciplinary Feasible (MDF) method that internally used the coupled adjoint method for gradients. Computational costs are competitive with the combination of BLISS and POD being less than twice as expensive as the MDF method.

Nomenclature

M	number of modes used in approximation
N	number of snapshots
$\mathbf{R}$	autocorrelation tensor
$\mathbf{u}$	arbitrary function to be generated
x	vector of independent variables
λ	eigenvalue
η_i	coefficient of the i-th mode in a function expansion
Ω	domain of interest
$\varphi^j(x)$	j-th POD basis mode
$\langle\cdot\rangle$	averaging operator
$\|\cdot\|$	L^2-norm
Y	vector of interaction variables
Y_i	vector of interaction variables computed by discipline i
$Y_{i,j}$	vector of interaction variables output by discipline j that are inputs to discipline i
X	vector of discipline design variables
X_i	vector of design variables for discipline i
$x_{r,j}$	j^{th} local design variable in discipline r
ΔX_i	vector of design variables in the optimization problem for discipline i
Z	vector of system design variables
A	matrix of interaction variable gradients
I	Identity matrix
Φ	overall system objective function
G_i	vector of constraints in discipline i
L	vector of Lagrange multipliers, lift
ZL	vector of lower bounds on Z
ZU	vector of upper bounds on Z
ΔZ	vector of design variables for system optimization
ΔZL	vector of lower move limits on ΔZ
ΔZU	vector of upper move limits on ΔZ
G_{yz}	vector of constraints more dependent on Y and Z than X
D	drag
$\frac{L}{D}$	lift-to-drag ratio
W_{struc}	structural weight
W_{init}	initial cruise weight
R	range
t	vector of structural thicknesses
α	angle of attack
θ	vector of wing twists
u	vector of structural displacements
Γ	vector of circulations

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

Introduction

DESIGN decomposition methods for MDO such as Collaborative Optimization (CO)[1] or Bi-Level Integrated System Synthesis (BLISS)[2] are appealing because they are relatively easy to implement, allow for more autonomy of disciplines compared to more tightly coupled methods, and can more easily handle large numbers of disciplines. However, their computational cost can be significantly higher than using a tightly coupled method such as Multidisciplinary Feasible (MDF) methods when using higher fidelity analyses such as CFD for aerodynamics, finite element representations of structures, etc. A typical method of alleviating some of the overhead of decomposition methods is to use simpler models of the disciplines such as curve fits, response surfaces, etc. Instead of exchanging pressures and displacements directly, the coefficients in the approximation technique can be used instead. Our past work with Proper Orthogonal Decomposition (POD) indicates it may also be used as a dimensionality or bandwidth reduction technique with the advantage that POD can easily be used in multiple dimensions, guarantees convergence in the limit of a large number of observations, and can incorporate the governing equations of the system. In this work we describe how we are leveraging our past experience with POD as a data representation technique to improve decomposition methods for multidisciplinary design.

POD

Details of POD have been described in our past work[3–5] and only a brief review is presented here. The POD is a procedure for computing the optimal *linear* basis for representing a set of observed data. This data could be obtained from experimental results or, as presented here, by the numerical solution of a system of equations. The projection of a solution of interest onto the basis functions (or modes) provides a finite set of scalar coefficients that approximate that solution. The order of the system is reduced from very large numbers (hundreds of thousands for the external aerodynamics of a full aircraft) to very small numbers (in the hundreds or thousands). 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 $\mathbf{u}(\mathbf{x})$ 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 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

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

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

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

and the notation $(\cdot, \cdot)$ 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 require (see Ref.[6]) 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 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

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

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. They represent the amount of the variation attributed to a particular mode, which leads to a sorting of the basis functions in descending eigenvalue order.

Even if only a small number of modes is necessary to represent a new function which is not a member of the original ensemble, we still have to solve an eigenvalue problem of order equal to that of the original problem (that is the size of the autocorrelation matrix). The computational cost involved in this solution can be mitigated by using iterative techniques to find the eigenvectors corresponding to the largest eigenvalues with highest energy content. A more elegant procedure called the *method of snapshots* due to Sirovich[7] reduces the cost of the solution to dot products of the snapshots with one another and an eigenvalue problem of size equal to the number of snapshots instead of the size of the snapshots.

As an example of applying POD we consider a 1-D panel code for computing the lift distribution over a wing. To begin the POD procedure we need a set of observations of the system. In this case we consider a series of different twist distributions as shown in Fig. 1a. The corresponding lift or circulation distributions are shown in Fig. 1b and are the actual snapshot data for POD. When applying POD to a collection of N snapshots we expect to get N modes. However, as shown in Fig. 1c the results in this case are 2 non-zero modes from the 11 snapshots. These two modes correspond to the well known basic and additional lift distributions. If we use these two modes to represent a solution not present in the snapshots used to generate the modes, we see the approximation when using a single mode and the exact nature of the representation with all modes in Fig. 1d. The ability to exactly represent a solution not used in the data to construct the modes is not uncommon for linear systems but is highly unlikely to occur for non-linear systems. In fact, there is recent evidence that for aerodynamic flow control problems, the POD may not be the most suitable way to represent phenomena that is quite different from that included in the observed snapshots.

BLISS

BLISS is an optimization method that decomposes the problem, typically along disciplinary boundaries. The numerous sub-problems, each with a large number of local design variables, are separated from the system level optimization that has a few global design variables. Each sub-problem concurrently performs an optimization which is formulated to improve the overall system. These sub-problem optimizations alternate with system level optimizations to complete each iteration cycle. An overall objective is improved, or at least constraint violations are eliminated, during each cycle.

Fig. 2 illustrates a generic BLISS problem with 3 disciplines. The design variables for the overall optimization are the system variables, Z, and the discipline variables, X. The interaction variables, Y, are outputs of a discipline that are also inputs to other disciplines. In the case of high-fidelity aeroelastic design these interaction variables would represent the tens of thousands of surface aerodynamic loads and structural displacements. Each interaction variable is assumed to be differentiable with respect to the design variables to at least first order.

In Fig. 3 the overall algorithm is shown. The system analysis is the full multidisciplinary analysis, which is problem dependent and likely to be iterative. After evaluating the state of the system, a judgment can be made to proceed with an optimization cycle or terminate the problem. For the discipline optimizations we need to formulate objective functions for each discipline that result in the minimization of the overall system objective function, denoted by Φ and assumed to be one of the elements of Y.

Total derivatives of Y with respect to $x_{r,j}$ (the j^{th} local design variable in discipline r) are computed by solving the Global Sensitivity Equations (GSE)[8] for a given $x_{r,j}$

$$[A]\left\{\frac{dY}{dx_{r,j}}\right\} = \left\{\frac{\partial Y}{\partial x_{r,j}}\right\}, \tag{5}$$

where A is a square matrix of dimensionality equal to the number of interaction variables and made up of sub-matrices as follows

$$A = \begin{bmatrix} I & A_{1,2} & A_{1,3} \\ A_{2,1} & I & A_{2,3} \\ A_{3,1} & A_{3,2} & I \end{bmatrix}, \tag{6}$$

where I is the identity matrix and $A_{r,s}$ is the matrix of the gradients of interaction variable outputs with respect to interaction variable inputs

$$A_{r,s} = -\frac{\partial Y_r}{\partial Y_s}. \tag{7}$$

The computation of the gradients in Eq. 7 are the parallel discipline sensitivity analyses in Fig. 3.

With the system sensitivity analysis computed from Eq. 5 the overall system objective function can be expanded in a Taylor series as a function of X

$$\Phi = \Phi_0 + \frac{d\Phi}{dX_1}^T \Delta X_1 + \frac{d\Phi}{dX_2}^T \Delta X_2 + \frac{d\Phi}{dX_3}^T \Delta X_3 + \cdots. \tag{8}$$

The change in the overall objective function due to a change in the local design variables is

$$\Delta\Phi = \frac{d\Phi}{dX_1}^T \Delta X_1 + \frac{d\Phi}{dX_2}^T \Delta X_2 + \frac{d\Phi}{dX_3}^T \Delta X_3, \tag{9}$$

and the contribution of the i^{th} discipline is

$$\phi_i = \frac{d\Phi}{dX_i}^T \Delta X_i. \tag{10}$$

This equation is the objective function that the i_{th} discipline should minimize to achieve the minimum overall system objective function which leads to an optimization problem for each discipline, such as for discipline two:

Given:	X_2, Z, and $Y_{2,1}$, $Y_{2,3}$
Find:	ΔX_2
Minimize:	$\phi_2 = \frac{d\Phi}{dX_2}^T \Delta X_2$
Satisfy:	$G_2 \leq 0$

where the constraints are the local constraints, e.g. stress constraints for structures, etc. All of the discipline optimization problems can be carried out in parallel.

Returning to Fig. 3, we have completed the discipline optimizations and now proceed with the system level optimization using the design variables Z. To do this we need the derivatives of Φ with respect to Z. There are two options for computing these terms, BLISS/A and BLISS/B. BLISS/A computes these derivatives by a modified GSE.[2]

BLISS/B uses an algorithm that computes $\frac{d\Phi}{dP}$, where P includes Y and Z, from the Lagrange multipliers.[9] For $F = F(P)$ and $G_0 = G_0(P)$ the algorithm provides the following result

$$\frac{dF}{dP}_0 = \frac{\partial F}{\partial P} + L^T \frac{\partial G_0}{\partial P}.$$

For BLISS this leads to the following

$$\begin{aligned} \left.\frac{d\Phi}{dZ}^T\right|_0 \quad = \quad & \left(L^T \frac{\partial G_0}{\partial Z}\right)_1 + \left(L^T \frac{\partial G_0}{\partial Z}\right)_2 + \left(L^T \frac{\partial G_0}{\partial Z}\right)_3 \\ & + \left[\left(L^T \frac{\partial G_0}{\partial Y}\right)_1 + \left(L^T \frac{\partial G_0}{\partial Y}\right)_2 \right. \\ & \left. + \left(L^T \frac{\partial G_0}{\partial Y}\right)_3\right] \frac{dY}{dZ} + \frac{d\Phi}{dZ}^T. \end{aligned}$$

Once we have computed the derivatives using the above method or BLISS/A the overall system level optimization may be performed:

Given:	Z and Φ_0
Find:	ΔZ
Minimize:	$\Phi = \Phi_0 + \frac{dy_{1,i}}{dZ}^T \Delta Z$
Satisfy:	$ZL \leq Z + \Delta Z \leq ZU$
	$\Delta ZL \leq \Delta Z \leq \Delta ZU$

where ZL and ZU are the bounds on the Z design variables and ΔZL and ΔZU are move limits.

In some cases the disciplines may have constraints, denoted G_{yz}, that are more dependent on Z and Y than on X. To satisfy such constraints we must add their estimated values as constraints to the system level optimization

$$G_{yz}^T = G_{yz}^T\big|_0 + \left(\frac{\partial G_{yz}}{\partial Z} + \frac{\partial G_{yz}}{\partial Y}\frac{dY}{dZ}\right)^T \Delta Z \leq 0. \quad (11)$$

As an example of the application of BLISS, we consider a three discipline system with an aerodynamics discipline consisting of a 1-D panel code for computing the lift distribution on a wing, a structures discipline with a single wing spar with a circular cross section modeled by tubular beam finite elements, and a mission performance discipline that computes the takeoff weight to satisfy a range requirement via the Breguet range equation. Fig. 4 illustrates the discretization of the aerodynamics and structures. The design variables are the wing twists and structural thicknesses and the objective to minimize is takeoff weight, with lift equal to weight and stress constraints as shown in Fig. 5. Using the previously described BLISS procedure, the results for the twist distribution and skin thickness distribution are shown in Fig. 7 compared to an optimization performed using the MDF method with coupled gradients and Newton iterations to converge the system. The values of the design variables are indistinguishable and the objective functions matched to within half a pound but the computational time was over 8 times greater for BLISS compared to MDF.

BLISS and POD

The need for and benefits of using approximation methodologies in MDO decomposition methods have been successfully demonstrated for BLISS and other algorithms.[10,11] Our past use of POD has been for solution approximation, i.e. if we have made observations of the solution of a system for a series of different inputs we can approximate what the solution will be for a previously unseen set of input parameters. Thus we have an approximation method with similar characteristics to response surfaces, Kriging, etc. but with the governing equations of the system incorporated and the ability to recover the true solution in the limit of a large number of modes. As is always the case with making use of approximation methods the difficult question of accuracy and what uncertainty is involved to achieve such computational cost savings arises. It could be exact if we happen to be dealing with a simple problem, but it likely has some error which is relatively expensive to estimate.

In the case of using POD as a bandwidth reduction technique we have the reverse situation. We know the true solution but it is expensive to communicate in the case of high fidelity computational simulations. We may want to compress that information, through approximation if necessary, for interaction with others while still using the higher order solution internally. Because we continue to maintain the full order analysis the error we are committing in making the approximation is known and can be easily monitored. If the error is deemed excessive we can request refinement, at some increased cost, through updated models from additional observations or other means such as the domain decomposition technique we have previously described.[3]

The overall BLISS procedure is essentially the same when introducing POD models for the interaction variables. An initial set of representative data is necessary to compute the POD modes before we can enter the optimization procedure. These snapshots of the interaction variables could come from running system analyses for a range of design variables, computations of individual disciplines with anticipated ranges of design variables and interaction variables, a database of results from previous optimization runs, or any combination thereof. The important point is that we collect data similar to that we anticipate needing to approximate during the actual optimization to ensure that our approximation model will be as good as possible. From these snapshots we compute the POD models for each interaction variable as described before and distribute the modes to each discipline that normally provides the associated variable as an output or uses it as an input.

Once we have an initial POD model, the BLISS/POD optimization procedure can begin. The system analysis is typically unchanged when using POD so the overall system objective function is computed and the termination criteria is evaluated with no error from the approximation model. Once we reach the discipline sensitivity analysis step the full order, high dimensionality interaction variables computed during the system analysis step that would normally be direct inputs to the disciplines are replaced by their low-dimensional, POD coefficient counterparts, η. Fig. 6 illustrates the changes for the aerodynamics discipline. Note that low-dimensional data such as the structural weight, lift-to-drag ratio, etc. are unaffected by the use of POD. Computation of the A matrix and right hand side of Eq. 5 is essentially the same but instead of having hundreds of thousands of gradients of all of the surface loads with respect to all of the surface displacements in a high fidelity aeroelastic optimization, we have the sensitivities of hundreds of POD coefficients representing surface loads to hundreds of coefficients representing surface displacements.

Once the system sensitivity analysis has been completed, the discipline optimizations are unchanged except that the gradients are now an approximation. However, for a good POD approximation the error in the gradients should be comparable to the truncation error in the Taylor series expansion of the overall objective function. Typically the system variables, Z, tend to be of lower dimensionality compared to the interaction variables, Y, and therefore, the application of POD models does not provide compelling computational cost advantages. In the event that there are variables of significant dimensionality, an analogous procedure to that used in the sub-problems can be applied.

As long as the gradient approximations are sufficiently accurate the BLISS algorithm with POD should converge to the correct optimal and feasible solution. The accuracy of the gradients can be inferred by the measurable accuracy of the POD approximation and a comparison of the estimated change in the objective function from the Taylor series expansion and the real change in the objective function as computed by the system analysis which has no such approximation. If the error is deemed to be too large, the POD basis modes need to be improved and/or augmented.

One option for improving the POD modes is to save the discipline analyses performed during the course of the optimization procedure to compute updated POD models. A limitation of POD is that we need observations of a system that are similar to those we wish to approximate. But a good approximation of a system during a particular iteration of an optimization should obviously be achievable if we use the similar observations from previous iterations. This is not quite true in the case of highly non-linear problems such as those in transonic aerodynamics. It may be difficult, for example, to accurately represent a shock that is moving around over the surface of the wing with a reasonable number of observations. In this case we can choose to represent the small portion of the domain with moving shocks at full fidelity, while still using our reduced order model over the remaining portion of the domain.[3] In any case, we are only using the approximation to reduce the cost of the coupling and the effect on the outcome of the optimization is at worst a slightly worse optimum.

Results

The ability of POD to improve the performance of BLISS was tested using the aerostructural problem previously described. It is a three discipline system with an aerodynamics discipline consisting of a 1-D panel code for computing the lift distribution on a wing that enforces a lift equal weight constraint, a structures discipline with a single wing spar with a circular cross section modeled by tubular beam finite elements that enforces stress constraints and a minimum skin thickness, and a mission performance discipline that computes the takeoff weight to satisfy a range requirement via the Breguet range equation. Optimization problems with varying numbers of elements, n, for the discipline discretizations were run with an MDF method, BLISS, and BLISS/POD to verify accuracy and compare computational costs.

For the BLISS/POD results, the POD snapshots were obtained by running four aeroelastic solutions and using the corresponding circulations and displacements to compute a set of POD modes for each of these variables. In the case of this problem, prior experience with these codes indicated that only a few modes and hence a few snapshots were necessary to get a good POD model. For a general problem this information will not necessarily be known. One way this can be determined without wasting resources computing a large number of snapshots is to compute the eigenvalues as each snapshot becomes available. If there is a very rapid drop in the eigenvalues, that is a strong indication that further snapshots will not improve the model. For the case of the full aeroelastic design of a supersonic business jet we can expect to compute dozens of snapshots before initiating the design procedure, use decomposition methods to handle the moving shocks likely to occur during the design process, and make updates of the POD models from analyses made by disciplines during each iteration of the design cycle.

There are many ways to compare computational effort - numbers of iterations, numbers of discipline analyses, etc. Ultimately we are interested in the time to reach a solution, so wall clock time was used to compare cost in this work. The BLISS based computations did *not* take advantage of the opportunities for parallelism in the discipline sensitivity analyses and optimizations so there are further improvements possible. Computational costs for computing the snapshots and modes for POD were included for BLISS with POD. The results are summarized in Table 1. The accuracy of BLISS and BLISS/POD is compared using the MDF method as a reference solution. Times are non-dimensionalized with respect to the time for the MDF problem. The results of the case for $n = 40$ are shown in Fig. 8.

Table 1: Comparison of MDF, BLISS, and BLISS/POD

n = 10

	MDF	BLISS	BLISS/POD
Objective	15918.3	15918.8	15918.0
$\|t - t_{MDF}\|$	0.	$1.0e10^{-5}$	$7.8e10^{-6}$
$\|\theta - \theta_{MDF}\|$	0.	$8.2e10^{-6}$	$7.2e10^{-6}$
Time	1.00	8.42	3.76

n = 20

	MDF	BLISS	BLISS/POD
Objective	15868.2	15868.3	15868.2
$\|t - t_{MDF}\|$	0.	$9.6e10^{-6}$	$1.5e10^{-5}$
$\|\theta - \theta_{MDF}\|$	0.	$9.6e10^{-6}$	$1.3e10^{-5}$
Time	1.00	8.95	2.63

n = 40

	MDF	BLISS	BLISS/POD
Objective	15845.0	15845.4	15845.3
$\|t - t_{MDF}\|$	0.	$1.8e10^{-5}$	$2.0e10^{-5}$
$\|\theta - \theta_{MDF}\|$	0.	$1.7e10^{-5}$	$1.6e10^{-5}$
Time	1.00	8.77	1.94

The first observation is that the results of the different optimization methods are essentially the same. Given that the methods all work, we can consider other criteria such as ease of implementation and computational cost. The computational cost of the MDF method was always the lowest. For BLISS, the computational cost was greater by a factor of approximately 8.5 for all cases. By using POD, the relative computational cost is decreasing as the size of the problem increases. This makes sense since a fixed number of POD modes is necessary regardless of the underlying discretization. For a large enough problem there would be a crossover and BLISS would be less costly with POD than MDF even with the extra overhead of computing snapshots and modes, conversions between full order and modal coefficient representations, etc.

For the ease of implementation, adding disciplines to BLISS is relatively straightforward. The additional entries are placed in the GSE and there is no requirement to determine which disciplines affect each other. This is in contrast to more efficient methods for computing sensitivities of coupled systems such as a coupled adjoint which do require careful attention to the interactions between disciplines.

In the case of the demonstration problem the linearity of the problem leads to optimal results that are identical using BLISS/POD with no special intervention required in terms of monitoring the accuracy of the approximations. This cannot be expected to hold true for more general problems with higher fidelity models which will require measurement of the approximation error and updates of the POD model to achieve acceptable optimization results.

Conclusions and Future Work

The use of POD as a bandwidth reduction technique in the BLISS decomposition method for MDO has been demonstrated. By replacing high dimensional quantities such as displacements and loads in an aeroelastic problem with POD modes and modal coefficients, of which only the modal coefficients need to be exchanged, significant computational cost savings can be achieved. The use of this technique in a representative aeroelastic design problem achieved computational costs very similar to MDF for moderate size problems.

Further work will verify the BLISS/POD methodology in high-fidelity multidisciplinary design optimization where the computational cost savings are more compelling and the challenges of monitoring accuracy and updating the POD models as necessary are greater. A similar optimization to the one demonstrated here will be done for a supersonic business jet using an Euler solver for the aerodynamics and a realistic finite-element representation of the wing structure in the pyMDO[12] framework. This framework, which has CFD and CSM solvers, load transfer algorithms, mesh warping routines, a geometry perturbation module, optimization packages, and visualization interfaces in it is allowing us to more efficiently research high-fidelity MDO problems such as the algorithm described in this paper.

Acknowledgments

The authors acknowledge the support of the Department of Energy as part of the ASC program under contract number LLNL-B341491.

References

[1] R.D. Braun. *Collaborative Optimization: An Architecture for Large-scale Distributed Design.* PhD thesis, Stanford University, 1996.

[2] J. Sobieszczanski-Sobieski, J.S. Agte, and R.R. Sandusky Jr. Bi-level integrated system synthesis (bliss). Technical Report TM-1998-208715, NASA, 1998.

[3] P.A. LeGresley and J.J. Alonso. Dynamic domain decomposition and error correction for reduced order models. *AIAA paper 03-0250*, 41st AIAA Aerospace Sciences Meeting & Exhibit, Reno, Nevada, January 2003.

[4] 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.

[5] 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.

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

[7] 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.

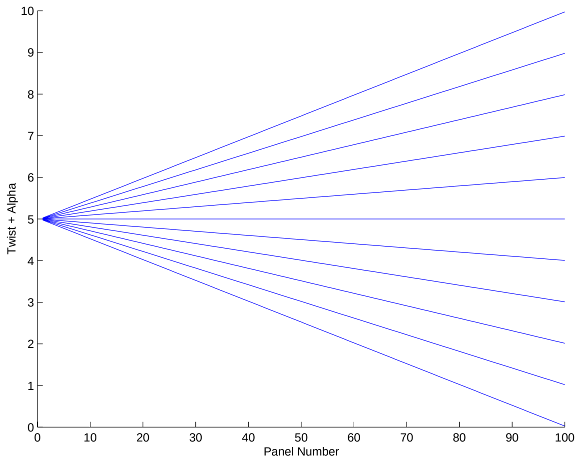
[8] J. Sobieszczanski-Sobieski. Sensitivity of complex, internally coupled systems. *AIAA Journal*, 28:153–160, 1990.

[9] J.-F.M. Barthelemy and J. Sobieszczanski-Sobieski. Optimum sensitivity derivatives of objective functions in nonlinear programming. *AIAA Journal*, 21:913–915, 1982.

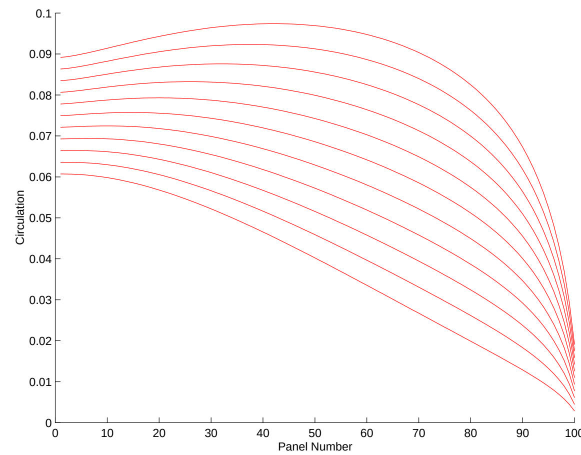
[10] S. Kodiyalam and J. Sobieszczanski-Sobieski. Bilevel integrated system synthesis with response surfaces. *AIAA Journal*, 38:1479–1485, 2001.

[11] I.P. Sobieski and I. Kroo. Collaborative optimization using response surface estimation. *AIAA Journal*, 38:1931–1938, 2000.

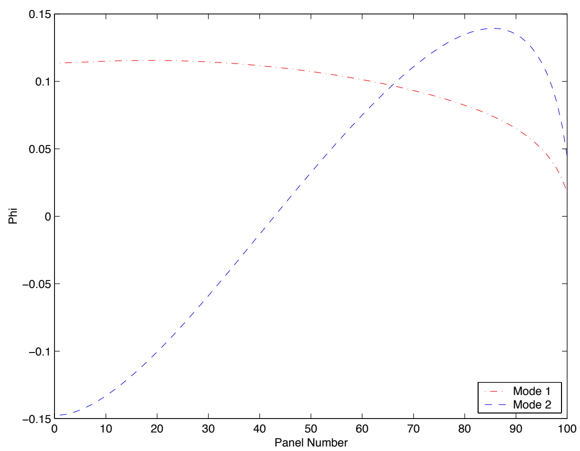
[12] J.J. Alonso, P.A. LeGresley, E.V. Weide, J.R.R.A. Martins, and J.J. Reuther. pyMDO: A framework for high-fidelity multi-disciplinary optimization. *AIAA paper 04-4480*, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York, August 2004.



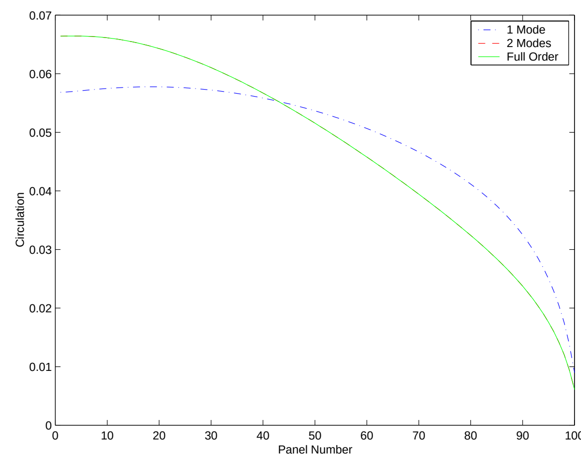
a) Varying Twist Distributions for Computing Snapshots



b) Snapshots: Circulation Distributions for Each Twist Distribution



c) Basis (Modes) Computed using POD



d) Approximate Solution Using the POD Modes

Fig. 1 Application of POD to a Panelcode.

Fig. 2 Interaction of Disciplines in BLISS.

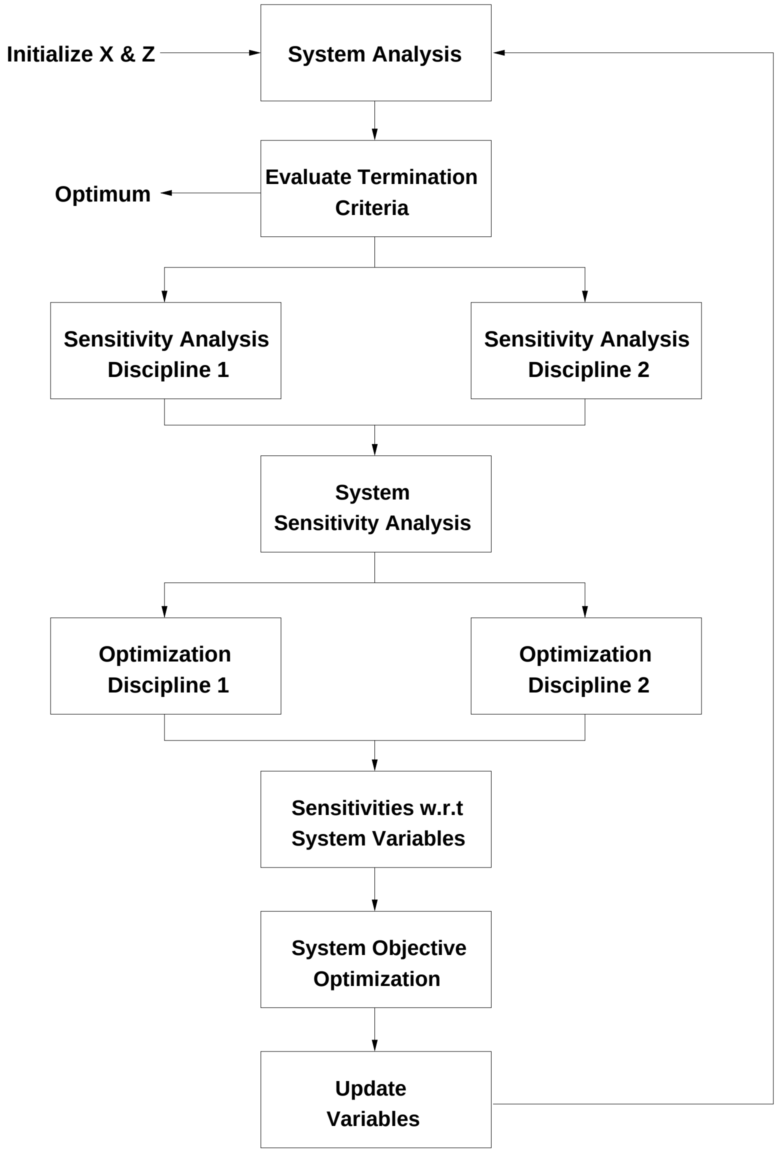


Fig. 3 Flowchart for BLISS Procedure.

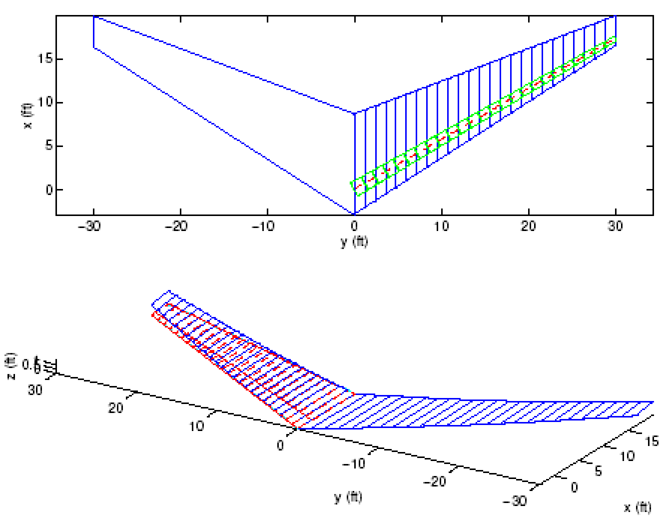


Fig. 4 Low Fidelity Aeroelastic Model.

Fig. 5 Interaction of Disciplines in BLISS for an Aeroelastic Optimization.

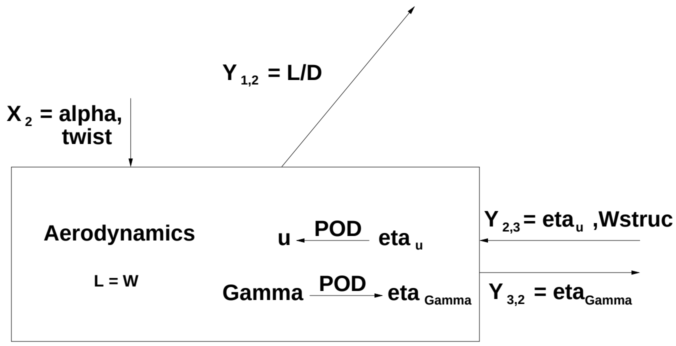
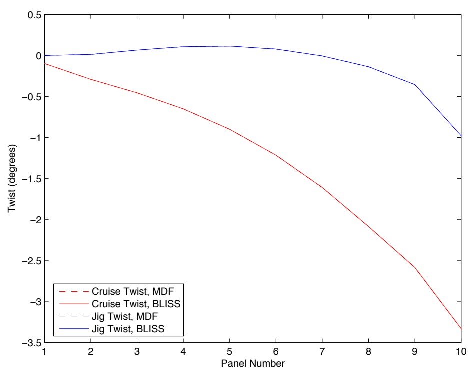
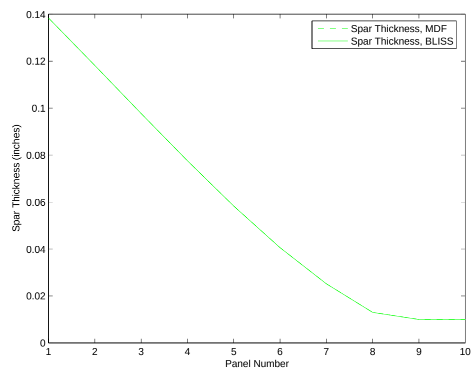


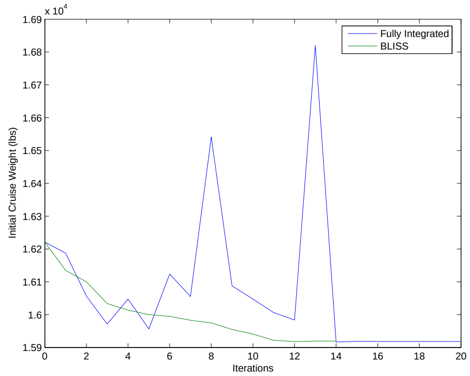
Fig. 6 Aerodynamics Discipline in BLISS/POD.



a) Twist Distribution

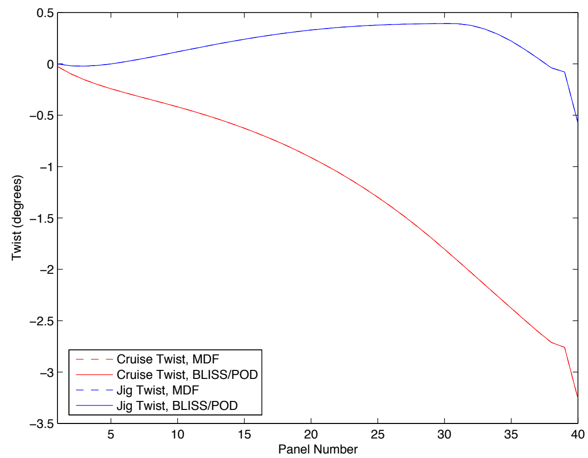


b) Thickness Distribution

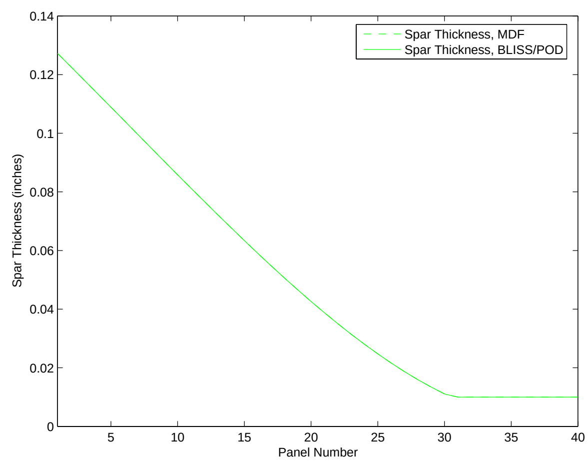


c) Convergence History

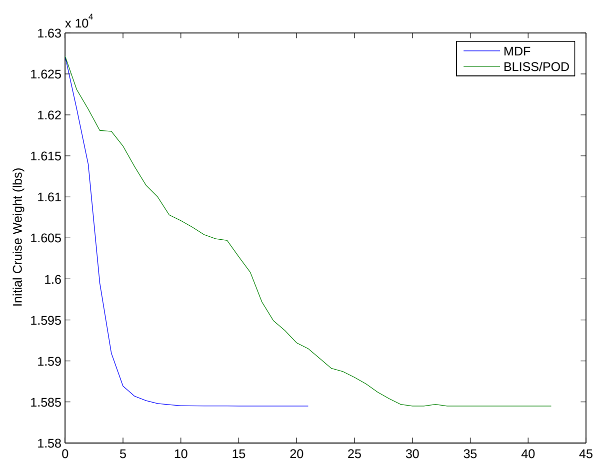
Fig. 7 Aeroelastic Design with BLISS.



a) Twist Distribution



b) Thickness Distribution



c) Convergence History

Fig. 8 BLISS/POD Results, $n = 40$.