A MULTIDISCIPLINARY OPTIMIZATION METHOD FOR DESIGNING INLETS USING COMPLEX VARIABLES

David L. Rodriguez*
Stanford University
Stanford, CA

Abstract

The Blended-Wing-Body is a conceptual aircraft design with rear-mounted, over-wing engines. Two types of installations have been considered for this aircraft: partially buried engines with boundary layer ingesting inlets and the more conventional podded engines with pylons. For both designs, the tight coupling between the aircraft aerodynamics and the propulsion system poses a difficult design integration problem. This paper presents a design method that approaches the problem using multidisciplinary optimization. A Navier-Stokes flow solver, an engine analysis method, and a nonlinear optimizer are combined into a design tool that correctly addresses the tight coupling of the problem. Gradients for the optimizer are computed using the complex variable method. Results from optimization runs on the podded installation are presented.

List of Symbols

$\alpha$ aircraft angle of attack
$C_D$ drag coefficient
$C_L$ lift coefficient
$C_M$ pitching moment coefficient
$\eta_r$ inlet pressure recovery
$m$ air flow rate
$m_a$ required engine air flow rate
$m_a^*$ predicted engine air flow rate
$m_f$ fuel burn rate
$m_{ff}$ air flow rate through fan-face
$P_t$ total pressure
$P_{t*}$ freestream total pressure
$T$ required engine thrust
$X$ geometric design variables

* Research Assistant, Aircraft Aerodynamics and Design Group, Member AIAA.
Copyright © 2000 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

Background

Boundary layer ingestion (BLI) by a propulsion system is a design concept that can improve propulsive efficiency. Ingesting the low momentum flow in the boundary layer reduces the ram drag. This concept is currently being considered for the Blended-Wing-Body (BWB), a large passenger transport configuration (Figure 1) currently being developed by Boeing, NASA, and several universities including Stanford. The BWB airframe and propulsion system are more tightly coupled than the same systems on a conventional aircraft. Hence, conventional design methods are unable to properly integrate these systems and fail to fully exploit the advantages of BLI. This lack of an effective propulsion integration method has been the primary motivation for a current research project at Stanford. Aerodynamic and propulsion system analysis methods are coupled with a nonlinear optimizer creating a more effective tool for designing an inlet/airframe configuration that incorporates the BLI concept.

Podding the engines and installing them on over-wing pylons is the other type of installation being considered for the BWB design. For comparison and to determine the benefits (if any) of BLI, the design tool can be applied to the podded engine design as well. The problem is essentially the same since the coupling between the aerodynamic and propulsion systems still exists. The podded engine design also presents a less complex test for the inlet design tool due to the lack of

Figure 1: BWB conceptual design.
separated flow usually present in a boundary layer ingesting inlet.

Method

The method described here has already been presented in References 1 and 2. However, for completeness the method is again described in this section. Note that although the motivation for the development of this inlet design method are the BLI inlets on the BWB, this method can be effectively applied to a more conventional podded engine design. The examples presented in the results section are in fact on a podded nacelle design.

A BLI inlet exhibits a strong coupling between the airframe aerodynamics and the propulsion system. The more low-momentum flow the inlet ingests, the lower the aerodynamic drag on the airframe and the lower the thrust requirement on the propulsion system. On the other hand, ingesting this boundary layer flow also reduces the inlet pressure recovery which reduces propulsive efficiency. An effective design method must be able to address this trade-off between aerodynamic and propulsive efficiencies and hence must be multidisciplinary in approach. The design method developed for this work addresses this coupling by integrating three basic tools: an aerodynamics analysis method, a propulsion system simulator, and a nonlinear optimizer. Brief descriptions of these tools and the integration techniques follow.

The aerodynamics method currently being used is an advanced Navier-Stokes method. CFL3D is a cell-centered, finite volume code developed at NASA-Langley. It uses a third-order, upwind-biased spatial discretization for the convective and pressure terms and a central-difference scheme for the shear stress and heat transfer terms. An implicit scheme is used for time-advancement. The code is extremely versatile with its large collection of turbulence models and ability to handle multiblock grids, patched, embedded, and overlapping grids. This versatility along with its popularity and validated accuracy were the primary reasons for selecting this code for the design tool. The turbulence model that was used for all computations discussed in this paper is the one-equation Spalart-Allmaras model. This model was chosen for its simplicity and its applicability to the multiblock grids necessary to analyze a complex configuration such as the BWB with nacelles installed. For the inlet design tool, CFL3D provides predictions of aerodynamic forces (lift, drag, and moment) along with the necessary fan face properties (predicted air flow rate and inlet pressure recovery).

The propulsion system simulator is an engine analysis code developed at NASA-Lewis. The NEPP code allows for 1D, steady-state, thermodynamic analysis of gas turbine engines. The design specifications of each engine component (such as the fan, turbine, burner, and inlet) are specified by the user allowing design and off-design analyses of the engine. The NEPP code was chosen for its speed, ability to handle off-design conditions, and wide usage in industry and government research. For the inlet design tool, the engine simulator takes the required thrust and inlet pressure recovery provided by the aerodynamics analysis and computes the fuel burn rate and the required inlet air flow rate.

The BWB engines studied in this work are advanced ducted propellers (ADP). While the bypass ratio (BPR) of the engine has not been finalized, the range of bypass ratios being studied is wide; BPR’s as large as 22 have been considered. However, for the work presented here, the bypass ratio (and actually the whole engine design) was held constant at 12. Varying the engine design would require including weight in the optimization problem and weight must be assumed constant for the method of this paper to produce reasonable results.

NPSOL is a gradient-based, constrained, nonlinear optimizer developed at Stanford University. An advanced quasi-Newton algorithm is implemented where each major iteration involves the solution of a quadratic subproblem based on the current objective function value, gradient, and Hessian approximation. Constraints can be linear or nonlinear. The NPSOL optimizer was chosen for its robust ability to handle nonlinear problems.

These three tools have been integrated into a multidisciplinary design tool. The basic architecture of this design tool is depicted in Figure 2. Taking computed information from the aerodynamics and propulsion analyses, the optimizer uses a set of user-specified design variables to minimize an objective function. For all the work discussed in this paper, this objective function is the engine fuel burn rate. The set of design variables usually includes geometric variables to modify the inlet/airframe shape, the angle of attack to control the total lift of the airframe, and an inlet back pressure level.
to ensure air flow compatibility between the engine analysis and the flow solver.

This air flow compatibility between the engine analysis and the flow solver is critical to the design methodology. The inlet air flow rate predicted by the flow solver must be the same as the required air flow rate that is computed by the engine analysis to make the problem consistent. This compatibility constraint can be satisfied by using the predicted air flow rate as a design variable. The flow solver can use this predicted air flow rate as an input and modify the corresponding boundary condition to attain this rate. As it minimizes the objective function, the optimizer modifies the value of the predicted air flow rate to ensure that it is equal to the required air flow rate computed by the engine analysis code. However, in practice the design variable used for this compatibility constraint is not the predicted air flow rate but rather a back pressure level.

The boundary condition used in the flow solver to control the air flow rate through the inlet is the standard “engine-outflow” type. At the boundary, the static pressure is specified and the values of the other flow variables are extrapolated from inside the flowfield. In order to obtain a specific flow rate, this back pressure must be determined iteratively which is extremely inefficient computationally. By using the back pressure value itself as the design variable, this iterative process is eliminated and the optimizer has direct control of the boundary condition. As before, the back pressure is modified to ensure the air flow compatibility constraint is satisfied as the objective function is minimized.

As shown in Figure 2, the design variables are passed from the optimizer to the flow solver. After modifying the inlet/airframe shape and the computational grid accordingly, the flow solver computes values for required aircraft thrust and inlet pressure recovery. These values are passed to the engine simulator where the fuel burn rate and required air flow rate are computed. These values are then in turn passed back to the optimizer. The lift and moment coefficients along with the predicted air flow rate are also passed back to the optimizer. The optimizer uses these computed values to determine a new set of design variables by trying to minimize the objective function and satisfy the constraints. The process repeats until an optimal solution is found.

Several nonlinear constraints are necessary to complete the problem. First of all, the lift of the aircraft is constrained be equal to the cruise lift coefficient. To trim the aircraft, a pitching moment constraint must also be introduced. As discussed earlier, the air flow compatibility constraint is necessary. Geometric constraints on the wing and nacelle (such as those necessary due to the aircraft structural design) manifest themselves as a set of linear constraints on the geometric design variables used in the problem.

**Force Computation**

Normally the force computation process with a CFD method is straightforward: integrate the pressure and skin friction forces on all solid walls. However, for the case with an engine, the correct method is not so clear. Accurately analyzing the flow within the engine is obviously not an option due to its complexity and therefore the engine must modeled. The typical way of modeling the engine is to introduce a flow exit-plane in the inlet and a flow entrance-plane at the engine exhaust region. This engine modeling is represented in Figure 3. Special boundary conditions are applied at the engine entrance and exit to control the properties of the flow through these planes. Computing forces on this configuration is not straightforward due to the fact that some flow exits the computational domain at the fan face and some flow enters at the engine exit plane; but it is still possible. Consider the control volume shown in Figure 3 which includes all the solid surfaces and the engine fan-face and exhaust planes. To compute the total force acting on this control volume, the pressure, shear forces, and momentum flux at the control volume boundaries must be integrated. Note that the momentum flux through the solid walls is zero and the shear forces (which are parallel to the control volume boundary) on the fan-face and engine exhaust planes are negligible.

For steady, unaccelerated cruise flight, the integration of these forces on this control volume should be zero. However, in the inlet design method, the engine simulator must be supplied with a gross thrust to balance the drag and thrust in the design problem. To compute the gross thrust of the engine using the CFD method, the force due to the flow entering the computational domain at the engine exit plane must be determined. This means the momentum flux and pressure must be integrated on this boundary face. Since the total force integration is zero, the force acting on all solid surfaces and the fan face is equal (except in sign) to the gross thrust and is the total drag (including any ram drag) of the aircraft. Therefore, the gross thrust requirement of the engine can also be computed by integrating the pressure and skin friction on all solid surfaces and the pressure and momentum on the fan-face.
Computation of Fan-face Total Pressure Recovery

The method of computing pressure recovery is an issue of much debate. The propulsion analysis method requires a single value that represents the total pressure losses at the fan face. Since the total pressure can vary over the fan-face area, the only solution is some sort of integrated quantity. When the total pressure distribution is not constant over the fan face, the correct integration scheme is not obvious. Many methods exist including area integration, mass flow integration, momentum integration, and even various forms of energy integration. All these methods can give very different results for a fan face with a widely varying total pressure distribution, like that found in a BLI inlet for example. The method chosen for this inlet design method is based on the work by Livesey. This integration method computes pressure recovery by assuming a constant entropy flux and weighting the integration appropriately:

\[ \eta_r = \frac{1}{P_{\infty}} \exp \left( \frac{1}{n_{ff}} \int_{\text{fan face}} (\ln P_t) \, dm \right) \]  

Note that momentum is not conserved with this method. While other methods which do conserve momentum exist, these methods either do not conserve other important properties (such as mass flow rate) or involve some sort of conservative mixing process which reduces the total energy of the flow. While it is still unclear whether the method of Equation 1 is the correct method (if there even is a correct method), Livesey provides some argument for selecting this method. This integrated quantity is therefore the value of pressure recovery that is sent to the engine simulator in the inlet design method. Note that the issue of a widely varying total pressure distribution is not as much of a problem for the podded nacelle case since the air ingested is predominantly uniform. However, the integration method of Equation 1 is included in the method so that future BLI inlet optimization work can be performed.

Gradient Computation

Two methods of computing gradients (sensitivities) for the optimizer for presented in this section: finite differences and the complex variable method. Although complex variables were used for all optimization work in this paper, the finite difference method is presented for comparison.

Finite Differences

When analytic gradient computation is not possible or practical, the simplest and most common method for computing sensitivities is the method of finite differences. The method is straightforward and easy to implement. Perturbations in each design variable are taken and the perturbed objective and constraint values computed. A forward difference gradient computation (based on a truncated Taylor series expansion) is given by

\[ f'(x) \approx \frac{f(x + \Delta x) - f(x)}{\Delta x} \]  

which is well known to be first order accurate in \( \Delta x \). For \( N \) design variables, the forward finite difference method requires \( N+1 \) objective function computations to compute all the necessary gradients. Ideally the step size is chosen to be very small to provide the most accu-
rate gradient possible. However, this is not always possible.

While the scheme is easy to implement, the finite difference method can be problematic when the accuracy of the objective function is not satisfactory. In order for the gradient computation to be accurate, the objective function computation must be very accurate since the difference is that of two large numbers when compared to the step size (\( \Delta x \)). Unfortunately, CFD codes are notorious for not converging to highly accurate solutions, especially when separated flow is present. Even if the CFD code could be converged to high accuracy, such convergence is extremely costly in terms of computation time which is highly undesirable in an optimization scheme. In order to be effective and efficient, the step size must be chosen to filter out the error of the CFD code and yet still produce accurate gradients. Too small a step size allows noise in the CFD solution to corrupt the gradient, while too large a step size will produce inaccurate results for functions whose gradients vary greatly with the design variable.

This strong dependence on step size is illustrated in the following example. The clean BWB wing (no engines) with a relatively coarse computational grid was used to compute the gradient of the lift coefficient with respect to angle of attack (\( \partial C_L/\partial \alpha \)). The convergence of the gradient is computed by applying the finite difference after each iteration of both solutions of the CFD code and hence in “lock-step.” Three different step sizes were used to illustrate how important the selection of step size truly is as shown in Figure 4. Note that as the step size gets smaller to try and improve accuracy, the inaccuracies in the CFD code cause the gradient computation to fail. Therefore, in order to make finite differencing a productive method, the step size for each design variable in a problem must be chosen appropriately which can be difficult and tedious.

**Complex Variables**

Computing gradients with complex variables is not a new concept. Lyness and Lyness and Moler proposed this concept back in 1967. This makes the complex variables method even more attractive. The method is based on a Taylor series expansion of a function which is perturbed in the imaginary dimension:

\[
f(x + ih) = f(x) + ihf'(x) - h^2f''(x) + \ldots\quad (3)
\]

Rearranging the terms of Equation 3 to group the real and imaginary parts and factoring out the imaginary step \( ih \) gives

\[
f(x + ih) = [f(x) - h^2f''(x) + \ldots] + ih[f'(x) - h^2f''(x) + \ldots]\quad (4)
\]

The real part of this equation is the value of function to order \( h^2 \) and the imaginary part is the first derivative of the function also to order \( h^2 \). What this allows is the computation of the objective function and a gradient at the same time and to the same accuracy. Since there is no subtraction required to compute the gradient, the step size, \( h \), can be as small as possible without needing the extreme accuracy required in the finite difference method. This property makes the complex method attractive because there are virtually no step size issues that must be considered as with the finite difference method.

Applying this method to a CFD code is rather straightforward. The CFD code is converted to use complex arithmetic instead of the real arithmetic. As a demonstration, a complex version of CFL3D was used on the clean BWB wing computational grid, and the gradient of lift with respect to angle of attack was once again computed. The convergence history of the gradient using the complex method is shown in Figure 5. Note that the convergence of the gradient is virtually identical to that of finite differences with a proper step size (Figure 4) for this case.

![Figure 5: Convergence of the gradient of lift with respect to angle of attack on clean BWB wing using different finite differences and the complex variable method.](image)

On the programming side, the complex method is very attractive because changes can be made to the code directly without having to run the code through a differentiation processor. Other methods, such as automatic differentiation, require that a code be post-processed each time a change is made. The resulting differentiated code is also difficult to very difficult to read and can be highly inefficient computationally.

On the other hand, the gradient computation using complex variables does come at a price, namely more computation time. A CFD code running with complex arithmetic requires about three times as many floating point operations as the standard code and the code still has to be run once for each design variable. So while computing the objective and gradients only takes \( N \)
solutions, each solution takes 3 times longer than the solutions in the finite difference scheme. Then again, the solutions in the finite difference scheme must be very tightly converged to obtain gradients with any kind of accuracy due to the subtractive errors. With no subtractive errors present, the complex variable solutions sometimes often do not have to be as tightly converged to produce gradients of the same accuracy as the finite difference method, though this does not seem to be the case in Figure 5. This is because the lift coefficient is easily converged in the clean BWB case to high accuracy. Based in the author's experience, however, for more complex flows the complex variable method can converge gradients quicker than the finite difference method.

There are other “tricks” that can speed up the computation of gradients using the complex step method. While these techniques have not been researched rigorously, the general experience of the author during the course of the development and application of the inlet design method has shown them to be effective. First of all, the complex part of the solution will usually converge in fewer time steps if the real part of the solution is already converged to the correct solution. Therefore, one useful technique is to first run the standard CFD code (which is much faster than the complex code) to obtain the real part, and then run the complex method to compute gradients. This has been found to speed up the entire process tremendously. Also, the imaginary parts of the solutions for each design variable can be saved for subsequent design iterations reducing convergence times even more, particularly when the optimizer makes only small changes in the design variables, such as when the method has nearly reached the optimal solution.

The Design Problem

Ultimately, the inlet design method will be applied to both the podded nacelle and BLI inlet designs of the BWB. At the time of publication of this paper, only the podded design problem was successfully completed. The BLI inlet design is underway but no results are available. The results from the podded nacelle design are presented here showing that the inlet design method is also applicable to a more conventional type of inlet design.

Configuration Definition

As detailed in Reference 12, the original BWB design had BLI inlets. However to determine the true advantages of the BLI installation, a podded installation must also be studied as a baseline for comparison. Using the baseline wing of Reference 12, a podded configuration was identified. The inlet design method was then applied to this configuration. Since no flow separation is present in the inlets of the podded design, this provided a less challenging test problem for the method before the more challenging BLI inlet design is attempted.

The baseline podded configuration is shown in Figure 6 in 3-view format. The pylons were designed to have a reasonable thickness and are symmetric. The engines are canted at 5° angle of attack from the wing reference plane and have no cant in the yaw direction. The thickness and in particular the radius of curvature of the inlets were selected using conventional design methods.

Some views of the computational grid used for the optimization work on this configuration are given in Figure 7. The computational grid was created by removing a block of a baseline wing grid and inserting nacelle grids. The interfaces between the wing and nacelle grids are not point-match and therefore the patching algorithm included in CFL3D was used for boundary conditions. The total number of points in this grid is just over 476,000. Obviously this is a very coarse grid for such a complex configuration. But to keep computation times reasonably low, optimization was performed on this grid.

Objective Function and Constraints

The objective function for this optimization case is the fuel-burn rate. This objective will allow the inlet design method to optimally integrate the propulsion system and the airframe inasmuch as the design variables will permit. The aircraft weight is assumed constant. The lift and air flow compatibility constraints are necessarily enforced in this problem. Of course, in this case, there are two air flow compatibility constraints since there are two engines. Also included in this case is a constraint which makes certain the pitching moment does not become lower than -0.17, based on the pitching moment of the baseline wing alone design. Ideally this
pitching moment should be zero meaning the aircraft is properly trimmed. However, the baseline wing design used in this example has a high node-down pitching moment which could not be corrected without major redesign work. Therefore instead of trimming the aircraft which would result in a large drag penalty, the pitching moment constraint makes certain the pitching moment does not get any worse in the inlet integration process. Also, constraints on the nacelle thickness and inlet leading edge radius were not necessary since no thickness design variables were included in this problem.

**Design Variables**

Ideally, a large number of design variables (~30-50) would be used to optimize complex geometry like the podded BWB. However, to make computation times manageable in a research atmosphere, only 18 were selected and are listed in Table I. The angle of attack and inlet back pressure variables are necessary to help enforce the lift and air flow compatibility constraints. A wing twist variable (washout) was also added to help enforce the pitching moment constraint. The variable adds a linear twist variation to the outboard section of the wing which is effective in controlling the pitching moment of a swept wing. Only the outboard wing twist is affected so that the cabin section is not warped.

The remaining variables control the shape and orientation of the nacelles themselves. These include the inlet length, inlet camber (which controls the inlet area distribution and outer shape of the nacelle), the nacelle pitch and yaw, the pylon height, and the pylon camber. Note that no thickness variables were included. Since the baseline nacelles were designed to have the minimum thickness and inlet lip radius, thickness variables were deemed mostly ineffective for this case. This may not be entirely true since thickness variables would

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>angle of attack</td>
<td>entire aircraft</td>
</tr>
<tr>
<td>2</td>
<td>inlet back pressure</td>
<td>centerline engine fan-face</td>
</tr>
<tr>
<td>3</td>
<td>inlet back pressure</td>
<td>outboard engine(s) fan-face</td>
</tr>
<tr>
<td>4</td>
<td>linear wing twist</td>
<td>outboard wing section</td>
</tr>
<tr>
<td>5</td>
<td>inlet length</td>
<td>centerline nacelle</td>
</tr>
<tr>
<td>6-7</td>
<td>inlet/nacelle camber</td>
<td>centerline nacelle</td>
</tr>
<tr>
<td>8</td>
<td>pylon height</td>
<td>centerline nacelle/pylon</td>
</tr>
<tr>
<td>9</td>
<td>nacelle pitch</td>
<td>centerline nacelle</td>
</tr>
<tr>
<td>10</td>
<td>inlet length</td>
<td>outboard nacelle</td>
</tr>
<tr>
<td>11-14</td>
<td>inlet/nacelle camber</td>
<td>outboard nacelle</td>
</tr>
<tr>
<td>15</td>
<td>pylon height</td>
<td>outboard nacelle/pylon</td>
</tr>
<tr>
<td>16</td>
<td>nacelle pitch</td>
<td>outboard nacelle</td>
</tr>
<tr>
<td>17</td>
<td>nacelle yaw</td>
<td>outboard nacelle/pylon</td>
</tr>
<tr>
<td>18</td>
<td>pylon camber</td>
<td>outboard pylon</td>
</tr>
</tbody>
</table>

Table I: Design variables used in podded BWB optimization.
allow for the control of inlet area distribution and the outer nacelle contours, which are important for controlling the shock strength and location on the nacelle outer surface. However, computation time limits forced the omission of these variables.

To demonstrate the geometric changes induced by the geometric design variables (4-18), each variable was perturbed separately and the resulting geometries shown in Figure 8. The perturbations are slightly exaggerated for clarity. The inlet length variables simply scale the nacelle length. It simply moves the inlet leading edge forward or backward and stretches or compresses the nacelle surface accordingly. The camber variables are based on cosine functions that range over a specified circumferential angle. The magnitude of the perturbation peaks at the center of the range and fades to zero at the edges of the range. The pylon height variable scales the pylon so that the sweep of the pylon remains constant. The nacelle pitch and yaw variables pivot the entire nacelle around a specified point. The pylon camber variable applies a sine function to the camber of the outboard pylon.

Results and Discussion

Two cases were completed on the BWB podded design. The first case, which included only the first three design variables, was completed to provide a true baseline value and to analyze the cruise performance of the baseline geometry. The second case includes all the variables and provides the optimal configuration shape (limited by the design space spanned by the design variables). These two cases are presented here and compared.

Analysis of Baseline Podded BWB Design

Only the first three design variables (angle of attack and two back pressures) are included in this case. Likewise, only the lift and air flow compatibility constraints are included in this problem. This optimization is performed to determine the performance of the baseline podded BWB configuration at cruise conditions which will provide a baseline for comparison once the optimization with all design variables is completed. The constraints were satisfied very quickly (3 design steps). More results from this baseline run are presented in the comparison figures in the next section. The performance of this baseline configuration is summarized in Table II.

Optimization of Podded Baseline BWB Design

The 18 design variables of Table I were used to minimize the fuel-burn rate of the baseline podded BWB configuration subject to the lift, moment, and air flow compatibility constraints. The final results are compared to the baseline results of Section in Table II. The optimization history is given in Figure 9. With the 18 design variables, the inlet design method was able to reduce the fuel-burn rate by almost 10%. This performance improvement was entirely due to the reduction in drag of over 23 counts since the pressure recovery of both inlets actually increased.

Figure 8: Effects of perturbed design variables on podded BWB geometry.
A comparison of the baseline and optimized geometries is given in Figure 11. The major difference is the height of the centerline pylon. The optimizer has moved the nacelles as far apart as possible with the selected design variables to relieve the shocks that form in the channel. The reduction of shock strength is better illustrated the pressure contour plots in Figure 12. The channel between nacelles is clearly a major source of drag and therefore the optimizer did what it could to improve the flowfield in this region. Since the spanwise location of the nacelles was held fixed due to engine-out performance constraints, the best the optimizer could do was raise the centerline nacelle as high as possible and still not violate the moment constraint. The centerline nacelle was raised higher than the outboard nacelles because then only one pylon’s wetted area would increase where if the outboard nacelles were raised higher, two pylons’ wetted areas would increase. Of course, the structural weight of the pylon may be the true limiting factor in the pylon height, but since structural analysis was included in the method at this point, the pylon weight penalty was not addressed.

The pressure contours (Figure 12) also show that the pressure gradients in the vicinity of the inlets have been reduced. The nacelle yaw, pitch, and inlet camber have been tailored to allow the flow to be ingested cleanly and therefore with minimal drag penalties. The yaw of the outboard nacelles has also been increased to further alleviate the shock strengths in the channel. This could cause problems in the engine-out flight condition since the moment are of the outboard nacelle has been increased. Further design work may have to include a yaw constraint to address this.

**Conclusions**

An inlet design method has been developed by integrating a Navier-Stokes flow solver, an engine simulator, and a nonlinear optimizer. This multidisciplinary design method correctly addresses the coupling between the propulsion system and the airframe. The method was applied to a podded nacelle BWB configuration but is also appropriate for a BLI inlet design. The method proved to be effective for the podded case in improving the design by significantly reducing the selected objective function and still satisfying all necessary constraints. The complex variable method for computing the necessary gradients proved to be very successful and efficient for practical use. The method has successfully identified the baseline spacing between the nacelles is a major source of drag.

**Future Work**

The next step is to apply this method to a BLI configuration like that shown in Figure 1. This should prove to be a more challenging problem because of the almost certain presence of separation in the inlets. However, if successful, the method should reduce this

<table>
<thead>
<tr>
<th>Performance Parameter</th>
<th>Baseline</th>
<th>Optimized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel Burn Rate (lb/hr)</td>
<td>32467</td>
<td>29336</td>
</tr>
<tr>
<td>Lift Coefficient</td>
<td>0.5000</td>
<td>0.5000</td>
</tr>
<tr>
<td>Pitching Moment Coefficient</td>
<td>-0.1676</td>
<td>-0.1700</td>
</tr>
<tr>
<td>Drag Coefficient</td>
<td>0.03074</td>
<td>0.02842</td>
</tr>
<tr>
<td>Centerline Engine Mass Flow Error</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Centerline Engine Inlet Pressure Recovery</td>
<td>0.9963</td>
<td>0.9920</td>
</tr>
<tr>
<td>Outboard Engine Mass Flow Error</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>Outboard Engine Inlet Pressure Recovery</td>
<td>0.9957</td>
<td>0.9926</td>
</tr>
</tbody>
</table>

**Table II: Performance comparison of baseline and optimized podded BWB configuration.**

The next step is to apply this method to a BLI configuration like that shown in Figure 1. This should prove to be a more challenging problem because of the almost certain presence of separation in the inlets.
separation since it is a source of poor pressure recovery and drag. One other constraint which must be included in the BLI case is a distortion constraint. Engine companies require that inlets provide at most a specified level of distortion and BLI inlets are notorious for violating this limit. By including distortion as a constraint in the optimization problem, the inlet design method should be able to reduce if not eliminate the violation of this distortion limit.

Acknowledgments

The author would like to acknowledge Bob Liebeck of Boeing for providing financial support and for aiding in identifying the problem that motivated this research. Professor Ilan Kroo of Stanford University provided a great deal of guidance in this research as well. NASA Ames provided most of the computer time on their SGI Origin 2000 supercomputers. Bob Biedron of NASA Langley also contributed to applying the complex variable method with the CFL3D code.

References