Optimizing Advanced Propeller Designs by Simultaneously Updating Flow Variables and Design Parameters

Magdi H. Rizk*
Flow Research, Inc., Kent, Washington

A scheme is developed for solving constrained optimization problems in which the objective function and the constraint function are dependent on the solution of the nonlinear flow equations. The scheme updates the design parameter iterative solutions and the flow variable iterative solutions simultaneously. It is applied to an advanced propeller design problem with the Euler equations used as the flow governing equations. The scheme's accuracy, efficiency, and sensitivity to the computational parameters are tested.

Nomenclature

c_1 = incrementing factor for optimization scheme [see Eq. (9)]
c_2 = decrementing factor for optimization scheme [see Eq. (9)]
C = positive constant for chord method [see Eq. (7)]
C_p = power coefficient
C_{p0} = desired power coefficient
D = propeller diameter
$\boldsymbol{e}_l$ = unit vector along the P_l axis
E = objective function
f = constraint function
$\boldsymbol{g}$ = solution of the flow-governing equations
$\underline{G}_l$ = lth component of ∇f relative to rotated coordinate system
$\boldsymbol{i}_l$ = unit vector along the $\underline{P}_l$ axis with components defined relative to the unrotated design parameter coordinate system
i_l = unit vector along the $\underline{P}_l$ axis with components defined relative to the rotated design parameter
L = number of design parameters
n_c = number of iterations required for convergence
$\boldsymbol{P}$ = vector of design parameters
$\underline{\boldsymbol{P}}$ = vector of design parameters relative to rotated coordinate system
P_l = lth component of design parameter vector
$\underline{P}_l$ = lth component of design parameter vector relative to rotated coordinate system
r = radial coordinate
R = blade tip radius
R_E = residual Euclidean norm
$\beta_{¾}$ = SR-3 blade angle at 75% blade span
$\beta_{0¾}$ = $\beta_{¾}$ corresponding to the power coefficient C_{p0}
β_0 = unperturbed blade angle distribution
β' = blade angle distribution perturbation
$\delta\underline{\boldsymbol{P}}$ = incremental vector used to update the vector of design parameters
$\delta P_{\max}$ = maximum incremental value allowed in updating the design parameters
ΔN = number of iterative steps at which $\boldsymbol{P}$ is periodically updated
ϵ = small positive incremental value used to perturb the design parameters
η = efficiency
μ = parameter determining the allowable region in design parameter space for searching for the optimum solution [see Eq. (18)]
ψ = flow iterative solution

Superscripts

n = iteration number
$*$ = optimum value

Subscripts

M = coordinate system rotated by the modified scheme
$(\underline{\ })$ = rotated coordinate system

Introduction

SOLUTIONS of constrained optimization problems minimize and objective function E subject to given constraints. In aerodynamic applications, the objective function and the constraint functions f_i, $i=1,2,\ldots$, depend on the flowfield solution $\boldsymbol{g}$. The optimization scheme developed here is applicable to situations in which the flow governing equations are nonlinear equations that are solved iteratively.

Conventional optimization methods (e.g., the steepest descent method and the conjugate gradient method) are iterative procedures that require the evaluation of the objective function many times before the converged optimum solution is determined. Since E and f_i are dependent on the flow solution $\boldsymbol{g}$ in addition to the vector of design parameters $\boldsymbol{P}$, the flow governing equation must be solved each time E and f_i are evaluated. Therefore, the application of conventional optimization schemes to aerodynamic design problems[1-5] leads to two-cycle (inner-outer) iterative procedures. The inner iterative cycle solves the analysis problem for $\boldsymbol{g}$ iteratively, whereas the outer cycle determines the optimum $\boldsymbol{P}$ iteratively. An alternative to this costly procedure is the single-cycle approach, which modifies the iterative procedure for solving the flow governing equations so that $\boldsymbol{g}$ and $\boldsymbol{P}$ are updated simultaneously.[6] Difficulties have been encountered, however, in attempting to apply this approach to advanced propeller design problems.[7] Our objective here is to develop a scheme based on the idea of simultaneously updating the flow variables and the design parameters that can overcome the problems previously encountered.

Received Feb. 27, 1988; revision received Aug. 8, 1988.

*Senior Research Scientist, Applied Mechanics Division; currently with Sverdrup Technology, Inc., Eglin Air Force Base, FL. Member AIAA.

A resurgence of interest in recent years in the turboprop propulsion system has been caused by the projected high fuel costs in the 1990's and the potential savings in fuel consumption that can be achieved with such a propulsion system. Advanced propellers operate at transonic speeds. Therefore, one of the two basic elements required for optimizing the design of these propellers is an analysis code capable of solving the nonlinear flow equations about the propeller so that the compressibility effects are predicted. The second element is an optimization scheme that can be efficiently combined with the analysis code.

Procedures have been developed for designing propellers by combining vortex lattice aerodynamic analysis methods with standard optimization schemes.[8,9] However, the first attempt to optimize propeller designs by using the full potential formulation,[10] which includes the necessary elements for transonic design, encountered difficulties in maximizing the propeller efficiency subject to a given power constraint.[7] The optimization scheme's inaccurate determination of the constraint surface resulted in these difficulties. Thus, in this scheme, efficiency was replaced as an objective function by an approximation, which is valid only under special conditions, and computations were limited to low Mach numbers.

In the present work, an optimization procedure is developed based on the idea of updating the flow variable iterative solutions and the design parameter iterative solutions simultaneously. This procedure has several common elements with the scheme of Ref. 7. However, it is more reliable and thus eliminates the difficulties encountered by the scheme of Ref. 7. Although applied here to the propeller design problem, this optimization scheme is suitable for application to general aerodynamic design problems. The Euler equations are assumed to be the flow governing equations. An implicit approximate factorization scheme[11] is used to compute the flowfield about an advanced high-speed propeller.

Approach

The propeller design problem is cast into an optimization formulation in which the optimum design parameter vector $\boldsymbol{P}^*$ is to be determined such that

$$E(\boldsymbol{P}^*;\boldsymbol{g}) = \min_{\boldsymbol{P}} E(\boldsymbol{P};\boldsymbol{g}) \tag{1}$$

subject to the constraint

$$f(\boldsymbol{P};\boldsymbol{g}) = 0 \tag{2}$$

with the flow variable vector $\boldsymbol{g}$ satisfying the flow governing equation

$$\boldsymbol{D}(\boldsymbol{g}) = 0 \tag{3}$$

subject to the boundary condition

$$\boldsymbol{B}(\boldsymbol{g};\boldsymbol{P}) = 0 \tag{4}$$

Our objective is to maximize the propeller efficiency η. The objective function is therefore defined by

$$E = -\eta$$

The propeller power requirements are constrained to a specified value through the constraint function

$$f = C_p - C_{p0}$$

Equation (3) is the system of Euler equations governing the flowfield, and Eq. (4) is the propeller solid-wall boundary condition. The vector of design parameters $\boldsymbol{P}$ defines the propeller geometrical configuration.

The goal of the optimization scheme is to determine the values of the design parameters that minimize the objective function E, subject to an equality constraint. A search must therefore be conducted in the design parameter space $\boldsymbol{P}$ for the optimum solution $\boldsymbol{P}^*$. This optimization problem is most conveniently solved in the rotated design parameter space $\underline{\boldsymbol{P}}$, with the $\underline{P}_1$ coordinate normal to the constraint surface and the $\underline{P}_l$ coordinates, where $l=2,3,\ldots,L$, parallel to the constraint surface. For fixed values of the components of $\underline{\boldsymbol{P}}$, let

$$\boldsymbol{g}^{n+1} = \psi(\boldsymbol{g}^n;\underline{\boldsymbol{P}}), \qquad n=0,1,2,\ldots \tag{5}$$

be the iterative solution for the analysis problem, where ψ denotes the solution obtained by applying the iterative scheme for solving the Euler equations, once using $\boldsymbol{g}^n$ as an initial guess. An implicit approximate factorization scheme is used here to solve the Euler equations and is described in Ref. 11. As for the analysis solution, obtaining the optimization solution requires the repeated application of Eq. (5) to update the flowfield. Although $\underline{\boldsymbol{P}}$ is held fixed in the former case, it is allowed to vary in the latter. The scheme used to update $\underline{\boldsymbol{P}}$ follows.

The vector of design parameter $\underline{\boldsymbol{P}}$ is updated every ΔN iterations. Therefore,

$$\underline{\boldsymbol{P}}^{n+1} = \underline{\boldsymbol{P}}^n + \delta\underline{\boldsymbol{P}}^{n+1} \tag{6}$$

where

$$\delta\underline{\boldsymbol{P}}^{n+1} = 0, \qquad (n+1)/\Delta N \neq 1,2,3,\ldots$$

In the iterative steps that satisfy the relation $(n+1)/\Delta N = 1,2,3,\ldots$, the incremental values for the design parameters are given by

$$\delta\underline{P}_1^{n+1} = -\frac{f^n}{|f^n|}[\min(C|f^n|, \delta P_{\max})] \tag{7}$$

$$\delta\underline{P}_l^{n+1} = \min\left(1, \frac{\delta P_{\max}}{|\Delta\underline{P}_l^{n+1}|}\right)\Delta\underline{P}_l^{n+1}, \quad l=2,3,\ldots,L \tag{8}$$

where

$$f^n = f(\underline{\boldsymbol{P}}^n;\boldsymbol{g}^n)$$

$$\Delta\underline{P}_l^{n+1} = \tfrac{1}{2}[c_1(\tau_l^{n+1}+1) + c_2(\tau_l^{n+1}-1)]\delta\underline{P}_l^{n+1-\Delta N} \tag{9}$$

$$\tau_l^{n+1} = -\frac{\Delta E_l^n \delta\underline{P}_l^{n+1-\Delta N}}{|\Delta E_l^n \delta\underline{P}_l^{n+1-\Delta N}|}$$

$$\Delta E_l^n = E(\underline{\boldsymbol{P}}^n + \epsilon\boldsymbol{i}_l^n;\boldsymbol{g}_l^n) - E(\underline{\boldsymbol{P}}^n;\boldsymbol{g}^n)$$

where ϵ is a small positive constant, and $\boldsymbol{i}_l^n$, $l=1,2,\ldots,L$, are the set of orthogonal unit vectors along the axes of the rotated coordinate system $\underline{P}_1^n, \underline{P}_2^n, \ldots, \underline{P}_L^n$. The solution $\boldsymbol{g}_l^n$ is a solution in which the lth component of $\underline{\boldsymbol{P}}$ is perturbed by ϵ.

The incremental displacement in the design parameter space introduced so that the constraint may be satisfied is taken in the direction normal to the constraint surface and is determined by the chord method in Eq. (7). The constant $\delta P_{\max}$ sets an upper limit on the magnitude of this incremental displacement. The incremental displacements given by Eq. (8) are introduced along the coordinate axes, which are parallel to the constraint surface with the purpose of reducing the value of the objective function. The sign of the incremental correction $\delta\underline{P}_l^{n+1}$, where $\delta\underline{P}_l^{n+1}$ is the lth component of the vector $\delta\underline{\boldsymbol{P}}^{n+1}$, is chosen to be opposite to that of $\partial E^n/\partial\underline{P}_l^n$. The magnitude of the increment $\delta\underline{P}_l^{n+1}$ is given by

$$|\delta\underline{P}_l^{n+1}| = c\,|\delta\underline{P}_l^{n+1-\Delta N}|$$

with an upper limit given by $\delta P_{\max}$, where $c>0$. If the signs of $\delta\underline{P}_l^{n+1}$ and $\delta\underline{P}_l^{n+1-\Delta N}$ are in agreement, then the last two

iterative solutions $\underline{P}_l^n$ and $\underline{P}_l^{n-\Delta N}$ fall to one side of the point along the $\underline{P}_l$ direction at which E is a minimum. In this case, c is set equal to the constant c_1, which is greater than 1. Increasing the magnitude of the step size in this manner accelerates the approach toward the point along the $\underline{P}_l$ direction at which E is a minimum. On the other hand, if the signs of $\delta\underline{P}_l^{n+1}$ and $\delta\underline{P}_l^{n+1-\Delta N}$ are not in agreement, then $\underline{P}_l^n$ and $\underline{P}_l^{n-\Delta N}$ fall on opposite sides of the point along the $\underline{P}_l$ direction at which E is a minimum. In this case, c is set equal to the constant c_2, which is less than 1. Decreasing the magnitude of the step size in this manner is necessary for convergence to the point along the $\underline{P}_l$ direction at which E is a minimum.

The updated components of the design parameter vector $\underline{\boldsymbol{P}}^{n+1}$ are used to calculate the new flow iterative solution $\boldsymbol{g}^{n+1}$, given by

$$\boldsymbol{g}^{n+1} = \psi(\boldsymbol{g}^n; \underline{\boldsymbol{P}}^{n+1}) \tag{10}$$

and the perturbed solutions $\boldsymbol{g}_l^{n+1}$, $l=1,2,\ldots,L$, given by

$$\boldsymbol{g}_l^{n+1} = \psi(\boldsymbol{g}_l^n; \underline{\boldsymbol{P}}^{n+1} + \epsilon \underline{\boldsymbol{i}}_1^{n+1}) \tag{11}$$

Although the optimization procedure is most suitably conducted in terms of the transformed parameters $\underline{P}_l$, $l=1,2,\ldots,L$, the flow solution is computed in terms of the physical design parameters P_l, $l=1,2,\ldots,L$. In order to express the transformed design parameters in Eqs. (10) and (11) in terms of the original design parameters, it is necessary to use the transformation equation, which relates these two sets of parameters. This equation is

$$\boldsymbol{P}^{n+1} = T^{n+1}\underline{\boldsymbol{P}}^{n+1}$$

where the orthogonal transformation matrix T^{n+1} is given by

$$T^{n+1} = [\boldsymbol{i}_1^{n+1}\ \boldsymbol{i}_2^{n+1}\ldots\boldsymbol{i}_L^{n+1}]$$

The unit vector $\boldsymbol{i}_1^{n+1}$ is normal to the constraint surface at $\underline{\boldsymbol{P}} = \underline{\boldsymbol{P}}^n$ and is given by

$$\boldsymbol{i}_1^{n+1} = \nabla f(\underline{\boldsymbol{P}}^n; \boldsymbol{g}^n) / |\nabla f(\underline{\boldsymbol{P}}^n; \boldsymbol{g}^n)| \tag{12}$$

where an estimate for $\underline{G}_l^n$, the lth component of ∇f, is given by

$$\boldsymbol{G}_l^n = [f(\underline{\boldsymbol{P}}^n + \epsilon \boldsymbol{i}_l^n; \boldsymbol{g}_l^n) - f(\underline{\boldsymbol{P}}^n; \boldsymbol{g}^n)]/\epsilon \tag{13}$$

The Gram-Schmidt orthogonalization process, which uses a set of L linearly independent vectors to construct a set of L orthonormal vectors, is used to construct the unit vectors $\boldsymbol{i}_l^{n+1}$, $l=2,3,\ldots,L$ along the rotated axes $\underline{P}_l^{n+1}$, $l=2,3,\ldots,L$. The following equation is used for this purpose:

$$\boldsymbol{i}_l^{n+1} = \boldsymbol{I}_l^{n+1} / |\boldsymbol{I}_l^{n+1}|, \qquad l=2,3,\ldots,L$$

where

$$\boldsymbol{I}_l^{n+1} = \boldsymbol{i}_l^n - \sum_{k=1}^{l-1} (\boldsymbol{i}_l^n \cdot \boldsymbol{i}_k^{n+1}) \boldsymbol{i}_k^{n+1}$$

In the initial iterative step, the vectors $\boldsymbol{i}_l$ are given by $\boldsymbol{i}_l = \boldsymbol{e}_l$, $l=1,2,\ldots,L$ where $\boldsymbol{e}_l$, $l=1, 2, \ldots, L$ are the set of orthogonal unit vectors along the axes of the coordinate system $P_1, P_2, \ldots, P_L$.

Although the flow variable vector $\boldsymbol{g}$ is updated each iterative step, the coordinate system in the design parameter space is rotated every ΔN iterations. The unit vectors $\boldsymbol{i}_l$, like the vector of design parameters $\boldsymbol{P}$, are updated only in the iterative steps that satisfy the relation $(n+1)/\Delta N = 1,2,3,\ldots$.

The optimization scheme described above requires that $L+1$ iterative problems be solved in parallel. In addition to the main solution, L perturbed solutions are computed in which each of the design parameters in the transformed space $\underline{P}_1, \underline{P}_2, \ldots, \underline{P}_L$ is perturbed. The computational costs and the computer memory requirements are therefore proportional to $L+1$. A modification to this scheme, which requires that only L iterative solutions be obtained, is now introduced. In the modified procedure, the perturbation solution associated with the perturbed design parameter in the direction of the $\underline{P}_1$ axis, normal to the constraint surface, is not computed. This solution was used in Eq. (13) to compute $\underline{G}_1^n$, which is required for the calculation of the vector $\boldsymbol{i}_1^{n+1}$, which determines the direction normal to the constraint surface in Eq. (12). In the absence of this solution, a new procedure for rotating the design parameter space must be defined. The procedure is first explained for the case of a two-design-parameter problem and is then extended to the general multi-design-parameter problem.

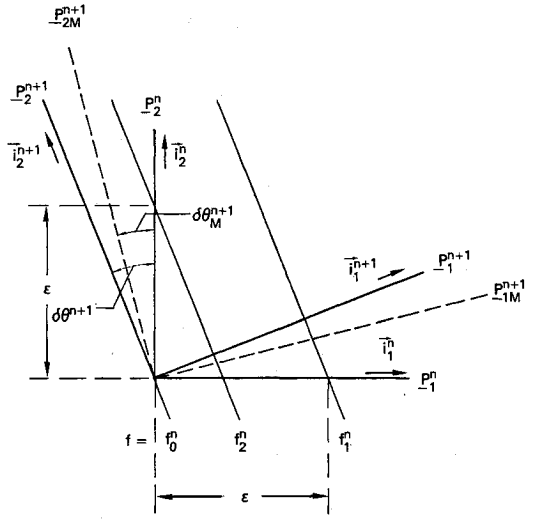

Fig. 1 Two-dimensional design parameter space.

Figure 1 shows the design parameter space for a two-design-parameter problem. In the figure, the constraint function values f_0^n, f_1^n, f_2^n are defined as follows:

$$f_0^n = f(\underline{\boldsymbol{P}}^n, \boldsymbol{g}^n)$$

$$f_1^n = f(\underline{\boldsymbol{P}}^n + \epsilon \boldsymbol{i}_1^n, \boldsymbol{g}_1^n)$$

$$f_2^n = f(\underline{\boldsymbol{P}}^n + \epsilon \boldsymbol{i}_2^n, \boldsymbol{g}_2^n)$$

In the modified procedure, the chord method, used in Eq. (7) to satisfy the constraint condition, is used to rotate the design parameter space. The rotation angle $\delta\theta_M^{n+1}$ given by

$$\delta\theta_M^{n+1} = \tan^{-1}[C(f_2^n - f_0^n)/\epsilon] \tag{14}$$

is used to rotate the coordinate system, where the subscript M indicates that the modified scheme is used. The angle $\delta\theta_M^{n+1}$ is now compared to the corresponding rotation angle $\delta\theta^{n+1}$ used in the original scheme and given by

$$\delta\theta^{n+1} = \tan^{-1}\left[\frac{f_2^n - f_0^n}{\epsilon}\frac{\epsilon}{f_1^n - f_0^n}\right] \tag{15}$$

This comparison shows that the term $f_1^n - f_0^n$ in the original scheme is replaced by ϵ/C in the modified scheme. Therefore, the modified scheme may be viewed as the original scheme with the exception that the exact value for $\boldsymbol{i}_1$ is replaced by an approximate estimate in which the gradient of f in the direction of the $\underline{P}_1$ axis, $\underline{G}_1$, is not calculated but is estimated using the same proportionality constant used in the chord method of Eq. (7). Thus,

$$\underline{G}_1 = 1/C \tag{16}$$

This is applicable for both the two-design-parameter problem and the general multi-design-parameter problem.

In the optimization scheme developed here, corrective increments are applied to the design parameter solutions every few iterations of updating the flow solutions. For convergence to occur, the signs of the increments must be chosen correctly to allow the iterative solution to approach the desired solution. The magnitudes of the increments are dependent on the computational constants c_1, c_2, and C. Because the design parameters are updated frequently during the iterative process, we are not concerned with determining the incremental step sizes that lead to the highest short-term convergence rate. In fact, this may be difficult to define, since the flow variable solutions are continuously changing during the iterative process. Our aim is to achieve design parameter convergence over a long term defined by the number of iterations required for the flow solution convergence. A wide range of incremental step sizes should produce the desired convergence properties over many iterations, even though convergence properties over a few iterations may differ. These comments apply to both of the schemes described above for determining the design parameter space rotation. The direct procedure for determining the design parameter space rotation in the original scheme is replaced by an iterative procedure in the modified scheme. Since this rotation is updated frequently during the iterative process, this replacement should have no substantial effect on the overall convergnece of the solution.

A potential problem exists when the modified scheme is used for rotating the design parameter axes. This problem is now discussed, and suggestions for overcoming it are then presented.

In the first $\Delta N-1$ iterative steps of solving the problem, the coordinate system in the design parameter space coincides with the original unrotated design parameter space $P_1, P_2, \ldots, P_L$. At the ΔNth iterative step, a new rotated coordinate system is determined. When Eq. (13) for determining $\underline{G}_1^{\Delta N-1}$ is used, we are guaranteed that the vector $\underline{i}_1^{\Delta N}$ points in the direction in which the constraint function increases. Consequently, the use of Eq. (7) will cause the iterative solution to approach the constraint surface. When Eq. (13) is replaced by Eq. (16) for determining $\underline{G}_1^{\Delta N-1}$, there is a possibility that the computed vector $\underline{i}_1^{\Delta N}$ points in the direction in which the constraint function decreases. In this case, the assumption that C is positive is wrong, and using it will cause the solution to diverge. This occurs if the vector $\underline{e}_1$ is nearly in the direction of $-\nabla f^{\Delta N-1}$, that is, if the quantity

$$-(\nabla f^{\Delta N-1}/|\nabla f^{\Delta N-1}|)\cdot \underline{e}_1$$

is close to unity. The probability of this occurring is approximately 1:4 in a two-design-parameter problem and is reduced further as the number of design parameters increases. There are two suggested approaches for overcoming this problem. In the first approach, the initial few iterations are performed using the original scheme for determining $\underline{G}_1^n$ by Eq. (13) in order to determine the correct initial directions for the P_1 axis. This may then be updated using the modified scheme, Eq. (16), in the rest of the computation. Realizing that the probability for the potential problem to occur is small, the second approach uses the modified scheme from the beginning of the computation. If divergence does occur, then the constraint function is redefined to be equal to the negative of the original constraint function, and the problem is solved again.

Results

The optimization procedure described above, combined with the Euler analysis code developed by Yamamoto et al.,[11] was used to find the twist distribution for the blades of the eight-bladed SR-3 propeller with the objective of maximizing its efficiency under the constraint of a desired power coefficient given by $C_{p0}=1.7$. The computations were performed for a freestream Mach number of 0.8 and an advance ratio of 3.06. Let $\beta_{0¾}$ be the blade angle at 75% blade span corresponding to the desired power coefficient. We take the blade angle distribution $\beta_0(r)$ corresponding to this propeller as our base configuration. A perturbation $\beta'(r)$ to the blade twist distribution $\beta_0(r)$ was computed so that the propeller efficiency is maximized subject to the power constraint. The perturbation twist distribution is given by

$$\beta'(r)=P_1+P_2\left(\frac{r-R/2}{R/2}\right)+2P_3\left[\left(\frac{r-R/2}{R/2}\right)^2-\frac{1}{2}\right] \tag{17}$$

where P_1, P_2, and P_3 are the components of the vector of design parameters P, and R is the propeller radius.

Experimentation with the propeller analysis code indicated that the flow iterative solution diverges when the blade tip angle exceeds a certain limit. To exclude the region leading to the divergence from our search in the design parameter space, the following redefinition of the objective function was introduced:

$$E=-\eta+\max[0.0, 0.1(\sqrt{P_2^2+P_3^2}-\mu)] \tag{18}$$

where μ determines the allowable search region. As the value of μ increases, the allowable search region also increases. The value of μ was taken to be equal to 5.0 unless otherwise specified.

The mesh used in the following computations consists of 45 points in the axial direction, 21 points in the radial direction, and 11 points between adjacent blades in the circumferential direction. Computations are initialized by the SR-3 flow solution, which corresponds to a 54.9-deg angle at 75% blade span. This initial solution was intentionally chosen not to be a close approximation of the desired solution. In all of the following computations, the modified coordinate rotation scheme, which determines $\underline{G}_1$ by Eq. (16) instead of Eq. (13), is used unless otherwise specified. Also, unless otherwise specified, the initial iterative guesses for the design parameters are set equal to zero, and the computational parameters c_1, c_2, C, δP_2^0, δP_3^0, $\delta P_{\max}$, ϵ, and ΔN are given, respectively, by 1.2, 0.6, 3.0, 0.5, 0.5, 1.0, 0.0001, and 40. The computations were performed on the NASA Lewis Cray X-MP computer.

The optimization procedure was applied to two-design-parameter problems and to three-design-parameter problems. For the two-design-parameter computations, the values of P_3 in Eqs. (17) and (18) are set equal to zero. Results for the two-design-parameter problem are presented, followed by those for the three-design-parameter problem. For the initial flow solution, which corresponds to a $\beta_{¾}$ value of 54.9 deg, the value of C_p was 1.1. Also, the value of $\beta_{0¾}$ was determined to be 58.067 deg.

The design parameters predicted by the optimization scheme are given by $P_1^*=-2.83$ deg, $P_2^*=5.51$ deg. The predicted solution does satisfy the power constraint. The value of C_p corresponding to this solution is 1.6999. The objective function E was reduced from the value -0.839 in the case of the original design, with $P_1=P_2=0.0$, to the value -0.908 in the case of the optimized design. The value of the efficiency was increased from 0.839 for the original design to 0.910 for the optimized design.

The computed value of efficiency, which corresponds to the optimized design, is approximately 5% higher than expected. Toward the end of this study it was discovered that an approximate formulation used in the analysis code to integrate the aerodynamic forces near the blade base was the cause of this overprediction. The main portion of the results presented here was obtained using the approximate formulation for computing the performance. These results are presented first. They are then followed by results obtained by using an accurate formulation for computing the performance. Although there may be no interest in the first set of solutions for the purpose of improving the propeller design, these results are valid for the purpose of testing the optimization scheme. In this case

$-\eta$ is viewed as an objective function without a physical meaning being attached to it. In the second set of results it was necessary to use an accurate formulation for computing the performance in order to show the required blade shape modifications for improved performance and the corresponding increase in performance obtained by optimization.

The iterative histories of the design parameters are shown in Fig. 2, and the iterative histories of C_p and η are shown in Fig. 3. From these figures two distinct stages in the convergence process of the solution may be identified. In the first stage, relatively rapid changes in the values of $\boldsymbol{P}$, C_p, and η occur as they approach the converged values of the solutions. At the end of this stage, these parameters are close to their final values. In the second stage, minor adjustments take place as the parameter solutions converge to their final values.

The residual R_E is a measure of the convergence of the flowfield solution. Figure 4 compares the residual history for the design problem, in which $\boldsymbol{g}$ is updated in addition to $\boldsymbol{P}$, to the residual history for the regular analysis problem, in which $\boldsymbol{g}$ only is updated while $\boldsymbol{P}$ is held fixed. The figure indicates that modifying the propeller geometry, as the iterative solutions for the flow variables are updated, in the design problem does not negatively affect the rate of convergence of the flowfield solution in comparison to the analysis problem. In fact, the following results of our computations show that the convergence of the flowfield solution is accelerated when the design parameters are updated to satisfy the power constraint or to satisfy the conditions of the optimization problem. For a regular analysis problem with $\boldsymbol{P}$ set equal to $\boldsymbol{P}^*$, the number of iterations required for convergence was 4710. Throughout this report, convergence is assumed to be achieved when the magnitude of the residual R_E is reduced to the value of 10^{-7}. For a constrained solution in which the second component of the design parameter vector P_2 was set equal to the value P_2^*, while the first component was updated throughout the iterative process, so that the constraint $C_p = C_{p0}$ would be satisfied, convergence was attained after 4040 iterative steps, indicating an increased convergence rate relative to the regular analysis problem. For the design problem, in which both P_1 and P_2 were updated in a manner that allows the constraint $C_p = C_{p0}$ to be satisfied and the objective function E to be minimized, the number of iterative steps required for convergence was further reduced to 3250.

On the average, 0.972 s of CPU time were required for the iterative step in the design problem, and 0.403 s of CPU time were required for the iterative step in the analysis problem. Therefore, the average design iterative step required slightly more than double the CPU requirements for the analysis iterative step. In the design problem, two analysis problems are solved in parallel. The additional CPU requirement for the design problem is mainly due to generating a new computational mesh whenever the design parameters are updated.

For a regular analysis problem, the computational mesh is generated only one time at the beginning of the computation. For a design problem, however, it is necessary to regenerate the computational mesh whenever the design parameters are updated. In the present computations, this was done once every 40 iterative steps. The cost of mesh generation relative to the cost of solving the flow equations was acceptably low. As the value of ΔN decreases, however, a point may be reached at which the cost of generating the mesh becomes excessively high, and it may represent a substantial fraction of the total computational cost. In this case, a possible alternative to regenerating new meshes, whenever the design parameters are updated, is the use of approximate meshes that are generated by linearly combining $L+1$ reference meshes. The reference meshes may be updated every $K\Delta N$ iterative steps, where $K>1$. The need for making this approximation does not arise here, as the propeller analysis code used here has relatively slow convergence properties, and, therefore, the appropriate ΔN value is relatively large. However, the use of accelerating schemes, such as the multigrid scheme, would allow the ΔN value to be sufficiently low to require the use of the mesh approximation discussed above.

We have performed a single computation using the exact formulation for calculating $\dot{i}_1$, as given by Eq. (12) with $\bar{G}_1$ computed by Eq. (13). This formulation requires solving $L+1$ problems in parallel instead of L problems, in the case of the approximate formulation given by Eq. (16). The average iterative step for this computation required 1.474 s of CPU time. The number of iterations required for convergence was

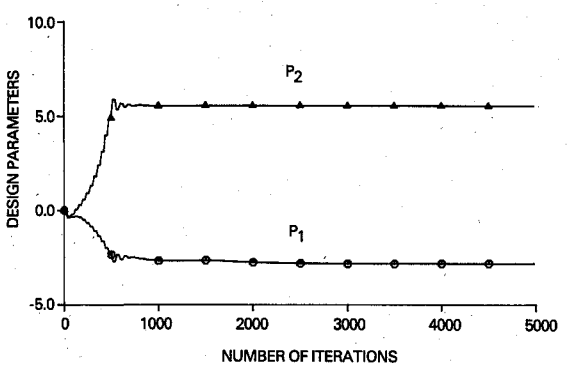

Fig. 2 Design parameter iterative histories for the two-design-parameter problem.

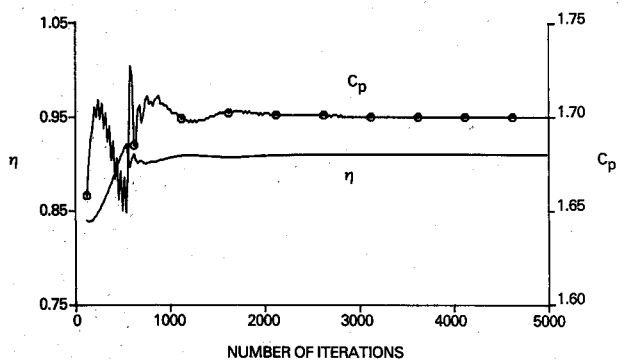

Fig. 3 C_p and η iterative histories for the two-design-parameter problem.

Table 1 Objective function at the optimum solution and perturbed solutions for the two-design-parameter problem

P_1	P_2	E_3
−2.83	5.51	−0.90773
−2.73	5.31	−0.90730
−2.93	5.71	−0.90728

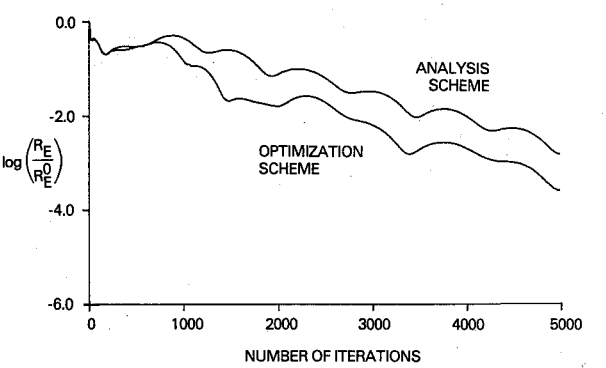

Fig. 4 Residual iterative histories for the analysis problem and the two-design-parameter optimization problem.

3425. Comparing these values to the corresponding values for the approximate formulation indicates that there is a strong advantage in using the approximate formulation over the exact formulation.

To verify that the computed solution is indeed the optimum solution, solutions were computed which were slightly perturbed from the optimum predicted solutions but which satisfied the power constraint. Table 1 compares the values of the objective function for the solution predicted by the optimization scheme, shown in the first row, to those for the perturbed solutions, shown in the second and third rows. It is apparent from the table that perturbing the design parameters causes the value of the objective function to increase. Therefore, the design parameters predicted by the optimization scheme do indeed minimize the value of the objective function.

The sensitivity of the scheme's convergence to the initial iterative guesses of the solution and to the computational parameters was tested by recomputing the problem defined above with perturbed initial conditions and computational parameters. Table 2 shows the number of iterative steps n_c required for convergence when different values are used for the initial iterative solutions and the computational parameters. It is clear from the table that the convergence properties of the scheme are weakly sensitive to the values of the initial conditions and the computational parameters. Needless to say, there is an optimum set of values for these parameters which maximizes the convergence rate of the scheme for a given problem. However, within a relatively wide range of these parameter values, good convergence is achieved. This is due to the frequent updating of the design parameters in the course of solving the problem. The CPU requirement for the average iterative step is approximately the same for all the cases solved, except for the case in which $\Delta N=25$. The CPU requirement for the average iterative step in this case is given by 1.078 s, in comparison to approximately 0.972 s for the other cases. This is due to the increased frequency of generating the computational mesh in the case with $\Delta N=25$. Figures 5–7 show the iterative histories for P_1, P_2, η, C_p, and R_E for the case in which the initial iterative guesses for the design parameters P_1^0 and P_2^0 were perturbed. Among all of the perturbed computations, the rate of convergence for this case was affected the most.

Table 2 Effect of perturbing the initial conditions and the computational parameters on the scheme's convergence for the two-design-parameter problem

P_1^0	P_2^0	ΔN	C	c_1	c_2	n_c
0.0	0.0	40	3.0	1.2	0.6	3250
3.0	−5.0	40	3.0	1.2	0.6	3690
0.0	0.0	25	3.0	1.2	0.6	3376
0.0	0.0	40	4.5	1.2	0.6	3252
0.0	0.0	40	6.0	1.2	0.6	3250
0.0	0.0	40	3.0	1.5	0.6	3333
0.0	0.0	40	3.0	1.2	0.4	3120
0.0	0.0	40	3.0	1.5	0.4	3281

The computations performed above for the two-design-parameter problem were performed with a value of 5.0 for μ. To perform computations that allow both parabolic and linear modifications to the blade angle distributions, it was necessary to reduce the value of μ to 4.0. The three-design-parameter optimization computations were solved using this value for μ. The main two-design-parameter computation was also repeated using this value for μ to allow a comparison between the two-design-parameter and the three-design-parameter results.

The optimum values of the design parameters for the two-design-parameter problem with $\mu=4.0$ were found to be given by $P_1^*=-2.35$ deg and $P_2^*=4.56$ deg. The value of C_p corresponding to this solution is 1.6999. The objective function E was reduced from the value -0.839 in the case of the original

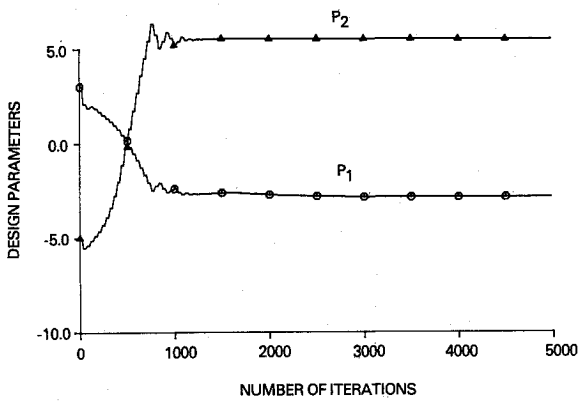

Fig. 5 Design parameter iterative histories for the two-design-parameter problem with perturbed initial conditions.

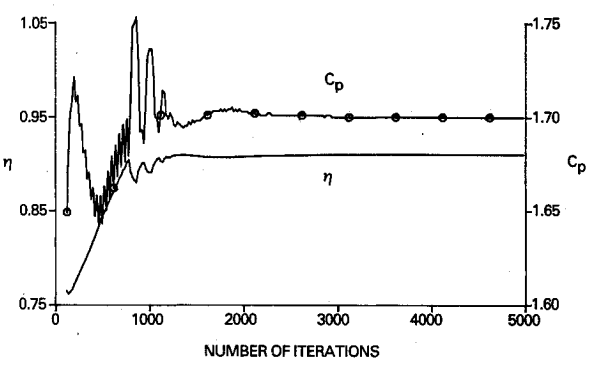

Fig. 6 C_p and η iterative histories for the two-design-parameter problem with perturbed initial conditions.

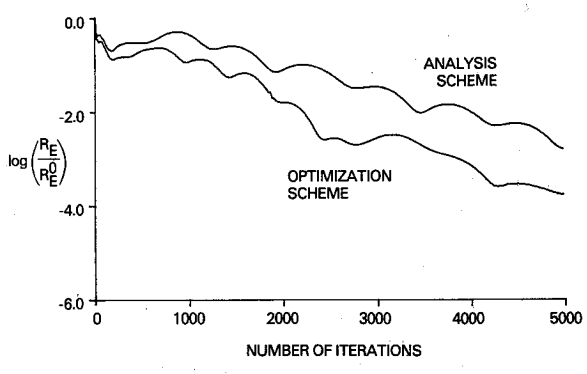

Fig. 7 Residual iterative histories for the analysis problem and the two-design-parameter optimization problem with perturbed initial conditions.

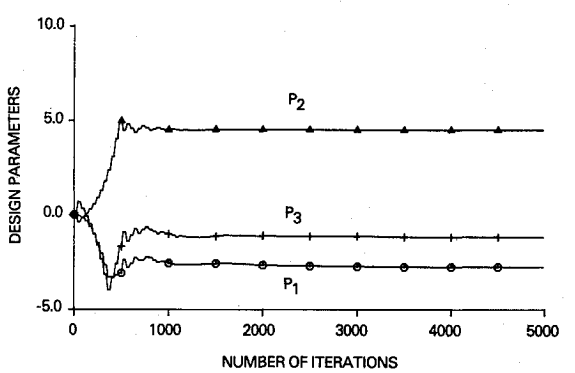

Fig. 8 Design parameter iterative histories for the three-design-parameter problem.

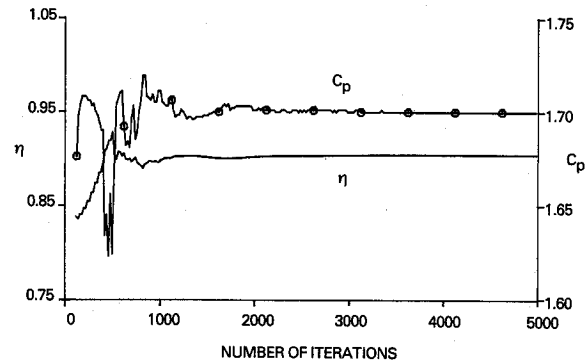

Fig. 9 C_p and η iterative histories for the three-design-parameter problem.

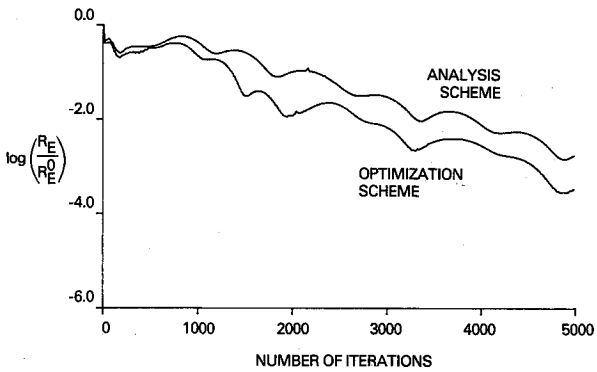

Fig. 10 Residual iterative histories for the analysis problem and the three-design-parameter optimization problem.

Table 3 Objective function at the optimum solution and perturbed solutions for the three-design-parameter problem

P_1	P_2	P_3	E
−2.77	4.50	−1.20	−0.90026
−2.87	4.50	−1.45	−0.90011
−2.87	4.69	−1.20	−0.89986
−2.67	4.50	−0.93	−0.90012
−2.67	4.30	−1.20	−0.89988

design, with $P_1 = P_2 = 0.0$, to the value -0.897 in the case of the optimized design. The value of η was increased from 0.839 for the original design to 0.900 for the optimized design. As expected, the magnitudes of both E and η determined with $\mu = 4.0$ are less than those determined with $\mu = 5.0$. As the value of μ decreases, the restriction on the allowable search region in the design parameter space increases. In the two-design-parameter problem, 3235 iterative steps were required for convergence. The CPU requirement per iterative step was 0.972 s. The optimum values of the design parameters for the three-design-parameter problem with $\mu = 4.0$ were found to be $P_1^* = -2.77$ deg, $P_2^* = 4.50$ deg, and $P_3 = -1.20$ deg. The corresponding values of C_p, E, and η are given by 1.6999, -0.900, and 0.905, respectively, indicating a design superior to that achieved by using only two design parameters. The number of iterative steps required for convergence was 3228, and the CPU requirement per iterative step was 1.459 s. The iterative histories for P_1, P_2, P_3, η, C_p, and R_E are shown in Figs. 8–10.

To verify the accuracy of the computed solution, several solutions were computed which were slightly perturbed from the optimum predicted solution but that satisfied the power constraint. Table 3 compares the values of the objective function for the solution predicted by the optimization scheme, shown in the first row, to those for the perturbed solutions shown in the following rows. It is apparent from the table that perturbing the design parameters causes the value of the objective function to increase. Therefore, the design parameters predicted by the optimization scheme do indeed minimize the value of the objective function.

Computations were performed using the accurate formulation for computing the propeller performance. In these computations it was found that η responds to changes in the design parameters at an iteratively much slower rate than that associated with the first set of computations. For that reason it was necessary to reduce the value of c_1 to 0.98. All other computational parameters were set equal to their same values used in the first set of computations. In this set of computations, it was determined that $\beta_{0¾} = 57.648$ deg. The value of C_p for the initial flow solution, which corresponds to a $\beta_{¾}$ value of 54.9 deg, was 1.2. By optimizing the blade shape for the two-design-parameter problem, the value of the efficiency was increased from 0.8229 for the original design to 0.8233 for the optimized design. For a regular analysis problem with $\boldsymbol{P}$ set equal to $\boldsymbol{P}^*$, the number of iterations required for convergence was 4320. A comparison of this number with the number of iterations required to solve the optimization problem, 3260, shows that the cost of solving the optimization problem is approximately twice the cost of solving a regular analysis problem.

In the computations presented above, the effect of varying the linear term of Eq. (17) on the propeller efficiency was investigated. In order to investigate the effect of varying the quadratic term in Eq. (17) on the propeller efficiency, a computation was performed in which P_3 was allowed to vary while P_2 was set equal to zero. In this case the design parameters predicted by the optimization scheme were given by $P_1^* = -0.79$ deg and $P_3^* = -2.07$ deg. The value of C_p corresponding to this solution was 1.7000, and the value of η was 0.82549. The number of iterations required for convergence was 3980. A comparison of the values of η for the two cases in which (P_1^*, P_2^*) and (P_1^*, P_3^*) were the design parameters shows that the introduction of a quadratic perturbation to the twist distribution is more effective in increasing the efficiency than the introduction of a linear perturbation.

Finally, the optimum values of the design parameters for the three-design-parameter problem were found to be $P_1^* = -3.34$ deg, $P_2^* = 3.92$ deg, and $P_3 = -3.23$ deg. The corresponding values of C_p and η are given by 1.7000 and 0.83291, respectively. It is apparent that using a combination of linear and quadratic perturbations in the blade angle distribution is much more effective for improving the efficiency than using only one of these distributions. Relative to the original SR-3 design, using both perturbed distributions increased the propeller efficiency by 0.0100. This is compared to values of 0.0026 for the quadratic distribution alone and a value of 0.0004 for the linear distribution alone. The number of iterative steps required for convergence was 4380, compared to 4460 for the regular analysis problem.

The iterative histories for $\boldsymbol{P}$, η, C_p, and R_E in the second set of computations, not presented here, are similar to those of the first set of computations (Figs. 2–10) and may be found in Ref. 12.

Figure 11 compares the optimum blade angle perturbations from the SR-3 baseline design predicted for the cases of linear, quadratic, and combined linear and quadratic shape functions. Curve C, which gives the blade angle perturbation distribution for maximum improvement in efficiency, shows that the efficiency of the SR-3 propeller can be improved by reducing the blade angle distribution both at the hub and at the tip. This explains the observed weak sensitivity of the propeller efficiency to linear variations in the blade angle distribution. The use of a linear shape function allows an increase in the blade angle at either the tip or the hub positions and a decrease in the blade angle at the other position. Therefore, the positive effect on efficiency resulting from the perturbed blade angle distribution at one of these positions tends to cancel the negative effect resulting from the perturbed blade

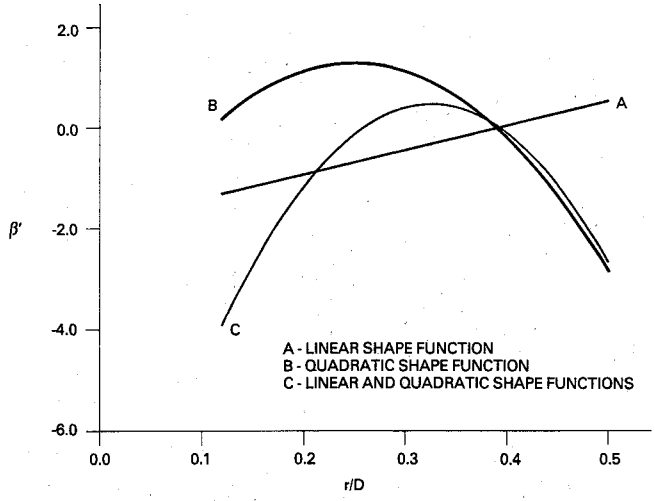

Fig. 11 Optimum blade angle perturbations.

angle distribution at the other position, leading to the apparent insensitivity of the efficiency to linear variations in the blade angle distribution. The maximum improvement in efficiency obtained here resulted from the use of linear and quadratic shape functions. Further improvement may be obtained by using other shape functions.

Conclusions

In this paper, we developed a scheme for solving constrained optimization problems in which the objective function and the constraint function are dependent on the solution of the nonlinear flow equations. The scheme updates the design parameter iterative solutions and the flow variable iterative solutions simultaneously, thereby eliminating the need for the costly inner-outer iterative procedure associated with the use of conventional optimization schemes. The scheme was applied to the problem of optimizing an SR-3 advanced propeller design with the Euler equations assumed to be the flow governing equations. The optimized design caused a 1.2% increase in the propeller efficiency.

Computations were performed to test the scheme's efficiency, accuracy, and sensitivity. The results indicate that the cost of solving an optimization problem with L design parameters is approximately equal to L times the cost of solving a regular analysis problem. The scheme is highly accurate in determining the solution of the constrained optimization problem. The convergence rate of the solution is weakly sensitive to variations in the computational parameters and the initial iterative guesses for the design parameters.

The main emphasis in this paper was to test the idea of updating design parameters and flow variables simultaneously. The particular scheme presented is applicable to optimization problems with a single equality constraint. However, this scheme may be extended to solve more general optimization problems with multiple constraints and inequality constraints.

Acknowledgments

This work was sponsored by NASA Lewis Research Center under Contract NAS3-24855.

References

[1]Hicks, R. M., Murman, E. M., and Vanderplaats, G. N., "An Assessment of Airfoil Design by Numerical Optimization," NASA TM X-3092, July 1974.

[2]Haney, H. P., Johnson, R. R., and Hicks, R. M., "Computational Optimization and Wind Tunnel Test of Transonic Wing Designs," *Journal of Aircraft,* Vol. 17, July 1980, pp. 457–463.

[3]Hicks, R. M., "Transonic Wing Design Using Potential-Flow Codes—Successes and Failures," Society of Automotive Engineers, Paper 810565, April 1981.

[4]Cosentino, G. B. and Holst, T. L., "Numerical Optimization Design of Advanced Transonic Wing Configurations," AIAA Paper 85-0424, Jan. 1985.

[5]Davis, W., "TRO-2D: A Code for Rational Transonic Aero Optimization," AIAA Paper 85-0425, Jan. 1985.

[6]Rizk, M. H., "The Single-Cycle Scheme: A New Approach to Numerical Optimization," *AIAA Journal,* Vol. 21, Dec. 1983, pp. 1640–1647.

[7]Rizk, M. H. and Jou, W.-H., "Propeller Design by Optimization," AIAA Paper 86-0081, Jan. 1986.

[8]Chang, L. K. and Sullivan, J. P., "Optimization of Propeller Blade Twist by an Analytical Method," *AIAA Journal,* Vol. 22, Feb. 1984, pp. 252–255.

[9]Miller, C. J. and Sullivan, J. P., "Noise Constraints Effecting Optimal Propeller Designs," Society of Automotive Engineers, Paper 85-0871, April 1985.

[10]Jou, W.-H., "Finite Volume Calculation of Three-Dimensional Flow Around a Propeller," *AIAA Journal,* Vol. 21, Oct. 1983, pp. 1360–1364.

[11]Yamamoto, O., Barton, J. M., and Bober, L. J., "Improved Euler Analysis of Advanced Turboprop Propeller Flows," AIAA Paper 86-1521, June 1986.

[12]Rizk, M., "Aerodynamic Optimization by Simultaneously Updating Flow Variables and Design Parameters with Application to Advanced Propeller Designs," Flow Research, Rept. 447, July 1988.