# EFFICIENT AND ROBUST DESIGN OPTIMIZATION OF TRANSONIC AIRFOILS

by

Changyeol Joh

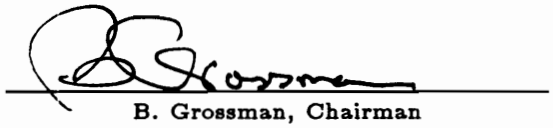
Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Aerospace Engineering

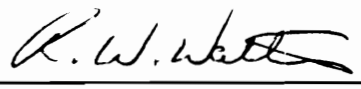
APPROVED:



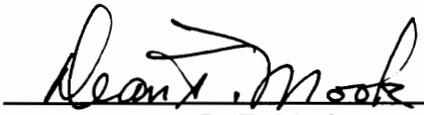
B. Grossman, Chairman

R. T. Haftka

J. A. Schetz



R. W. Walters



D. T. Mook

May, 1991

Blacksburg, Virginia

# EFFICIENT AND ROBUST DESIGN OPTIMIZATION OF TRANSONIC AIRFOILS

by

Changyeol Joh

Committee Chairman: Bernard Grossman

Aerospace Engineering

(ABSTRACT)

Numerical optimization procedures have been employed for the design of airfoils in transonic flow based on the transonic small-disturbance (TSD) and Euler equations. A sequential approximation optimization technique was implemented for solving the design problem of lift maximization with wave drag and area constraints. A simple linear approximation was utilized for the approximation of the lift. Accurate approximations for sensitivity derivatives of the wave drag were obtained through the utilization of Nixon's coordinate straining approach. A modification of the Euler surface boundary conditions was implemented in order to efficiently compute design sensitivities without recreating the grid. Our design procedures experienced convergence problems for some TSD solutions, where the wave drag was found not to vary smoothly with the design parameters and consequently create local optimum problems. A procedure interchanging the role of the objective function and constraint, initially minimizing drag with a constraint on the lift was found to be effective in producing converged designs, usually in approximately 10 global iterations. This procedure was also shown to be robust and efficient for cases where the drag varied smoothly, such as with the Euler solutions. The direct lift maximization with move limits which were fixed absolute values rather than fractions of the design variables, was also found to be a reliable and efficient procedure for designs based upon the Euler equations.

## ACKNOWLEDGEMENTS

My Committee Chairman and Research Advisor, Prof. B. Grossman, has provided me with continuous and firm support throughout my research. I am indebted to his support. I must also deeply appreciate his invaluable support which enabled me to continue the study at VPI & SU and to graduate.

I am very grateful to Prof. R. T. Haftka for sharing his abundant experience in optimization through his teaching and help in research. I specially thank him for enlightening me in optimization area.

I would like to thank Prof. A. Schetz, Prof. W. T. Walters and Prof. D. T. Mook for serving in my Committee and Prof. W. H. Mason for helpful comments for my research. Also, I would like to mention all of my friends and colleagues for their daily help and useful discussions.

I would like to thank my M.S. thesis Advisor, Prof. Paul K. Chang, for introducing me to the field of Aerodynamic Design and strongly encouraging me to continue my education.

Special thanks should be given to my wife, Yunhwa, and son, Hanmiru, for their patience to the lonely and boring life in the U.S. Thanks are due to my parents, whose belief in education got me this far and also to my wife's parents for their support and concern.

This research was funded by the National Science Foundation under grant DMC–8615336, whose support is gratefully acknowledged.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# 1. INTRODUCTION

Efficient transonic flow analysis methods were introduced by Murman and Cole[1] and Jameson[2] in the early 1970's. Since that time, in addition to the evolution of sophisticated analysis procedures for the transonic flow regime, a variety of numerical aerodynamic design methods have been developed. These can be broadly categorized as inverse methods, numerical optimization methods and shock-free design methods, such as the hodograph method and fictitous gas approach. Among these methods, currently two classes including inverse methods and numerical optimization methods are widely used as practical aerodynamic design tools. In the inverse design methods, a distribution of flow variables, usually pressure or velocity, is specified along the arc length of the airfoil, and the solution of the governing equations is used iteratively, to determine the corresponding airfoil shape. In numerical optimization procedures, a sequence of transonic flow analyses is performed over specified geometric shapes and this information is utilized to determine the geometric design variables which optimize a specific objective function, (usually based upon performance).

Before describing the general overview of the present study, a brief review of various methods in these two catagories will be helpful to understand the motivation and the current position of this study in the latter category. Good surveys of various aerodynamic design methods are provided by Holst *et al.*[3] and Slooff[4]. Most of the aerodynamic design methods were reviewed and their usefulness and limitations were summarized. Portions of the review presented here, especially for inverse methods, relies on information from those surveys.

Early work in the development of inverse design methods for the transonic flow regime started with transonic small disturbance(TSD) theory. These include Steger and Klineberg[5] and Shankar *et al.*[6] for two-dimensional airfoil design, and Forsey and Carr reported by Lock[7] and Shankar[8,9] for three-dimensional wing design. Most of the methods were generally not well developed, in particular in the limited geometric capabilities and the almost common lack of control of closure. The *closure problem* in airfoil design refers to the fact that a prescribed pressure or velocity distribution does not necessarily result in a closed airfoil trailing edge. However these methods showed two different basic approaches for the inverse methods, *i.e.* iterative-Dirichlet and iterative-Neuman(residual-correction) types, and provided the background for the next development of inverse design methods with full potential theory.

The inverse design methods based on full potential theory can be distinguished by two approaches. One class of methods follows the classic approach described by Lighthill[10]. In this method the airfoil corresponding to a target pressure distribution is obtained as the solution to a Dirichlet problem. The boundary conditions are linearized about a given approximate geometry being sought and the distribution of surface potential is obtained by integration of the target velocity(pressure) distribution. Then the inverse problem is solved as a sequence of Dirichlet boundary-value problems for the velocity potential with geometry corrections based on calculated normal velocity distribution after each Dirichlet step.

The first reported effort to solve the full potential transonic inverse airfoil problem based on this approach is that of Tranen[11]. He applied the complete equations to the inverse problem by modifying the conformal mapping relaxation solution technique of Garabedian and Korn[12] to use the Dirichlet boundary conditions.

Since this technique maps the computational grid to match the airfoil surface, Tranen was forced to applied the boundary conditions at the surface of some assumed airfoil. As a result the shape computed after the relaxation process converged was not completely compatable with the input pressure distribution. Thus he required a direct analysis to check the resulting pressure distribution for adjusting the surface displacement. The inverse and direct analysis were repeated alternatively to achieve convergence. This method is basically improperly formulated and there is no guarantee for convergence. A similar approach to that of Tranen was taken by Volpe[13] and Arlinger[14] for inverse methods for two-element airfoil systems.

A distinctly different approach has been taken by Carlson[15,16]. Instead of using a body conforming finite-difference mesh in the circle plane, he used Cartesian coordinates in the physical plane with the body not aligned with the mesh. The surface boundary conditions then were applied using special, complicated finite-difference expressions. The mixed boundary value problem was solved with the leading-edge shape or a greater portion of the airfoil fixed and the pressure prescribed over the remaining portion. Unlike the previous methods, this approach determined the airfoil shape simultaneously with the flow-field solution since the geometry was updated periodically during the relaxation sweeps rather than determined after a complete convergence. Airfoil closure was controlled in an *ad hoc* manner by changing the airfoil nose until the desired trailing-edge gap was obtained. The procedure was found to be economical.

As was first described by Lighthill for the incompressible problem, constraints arise for this category of inverse methods. Two constraints arise from the requirement that the airfoil profile be closed(or have a specified trailing-edge thickness). A third constraint requires that the prescribed pressure distribution be compatible with the specified free-stream velocity. Failure in satisfying these constraints

can result in an ill-posed Dirichlet problem and/or failure in obtaining the desired trailing-edge gap. The main advantage of the Dirichlet type methods is rapid convergence, provided appropriate auxiliary algebraic constraints are satisfied. In many published procedures, these constraints are obviously not taken into account.

In all of the early compressible flow methods based on the iterative Dirichlet type approach, as pointed out by Volpe and Melnik[17], the actual pressure computed on the surface of airfoils designed does not generally agree with the target pressure within arbitrary small tolerance, because not all the constraints on the inverse problem are specified. Volpe and Melnik[18] discussed the importance of those constraints rigorously and addressed a method in which the target pressure distributions satisfies all the constraints and, therefore, the problem is well posed and has complete control over closure problem, at least for two dimensions.

The first inverse method based on the iterative Dirichlet approach for three-dimensional potential flows is that of Henne[19]. It can be considered as the three dimension equivalent of Tranen's[11] two-dimensional method and only addressed constraint issues in an *ad hoc* fashion using a modified version of the Jameson-Caughey FLO-22 wing code[20]. Another iterative Dirichlet scheme for three-dimensional inverse design was performed by Shankar[21] which used a modified FLO-30 finite volume wing-fuselage analysis code[22]. Carlson's method[15,16] has been also extended to the three-dimensional transonic wing design in more generalized curvilinear grid system in Refs. 23-25. A modified version of FLO-30 is used and a viscous boundary layer correction is considered.

Another inverse design approach, often called *residual-correction type*, is to use a sequence of Neuman-type *direct* solutions. Here the pressure(or velocity) distribution on an airfoil surface and its force characteristics are computed and then compared to the desired target pressure(or velocity) and/or forces. The differences

between computed and target values, usually called the *residuals*, are then used in some geometry correction procedure to modify the airfoil contour, and this process is repeated as long as desired. The main advantage of the residual-correction type is its simplicity. Only a small investment of effort is required, since any existing transonic analysis code can be used without modification. The main disadvantage is the lack of guarantee that the iteration will converge with the residual reduced to arbitrary small levels. It also used relatively more computation time compared to the iterative-Dirichlet type approach. However, these methods do not necessarily converge for arbitrarily prescribed pressure distributions.

A residual-correction type of approach to the two-dimensional transonic inverse airfoil problem has been developed by Davis[26], which utilized Jameson's full potential code FLO-6. A transonic small disturbance wavy-wall formula is utilized for the surface modification which is driven by the residual. This approach is limited to small local modifications of existing pressure distributions. An advantage is that only modest development effort is needed to obtain a working code since any analysis code can be retained in its original form and only a simple geometry correction routine must be developed.

An approach somewhat similar to that of Davis has been performed at the NLR in the Netherlands, as reported in Ref. 27. This system combines the geometry modification routine based on the two-dimensinal version of the subsonic thin wing inverse code of Ref. 28 and an inverse supersonic wavy-wall formulation with the Boestoel TRAFS[29] analysis method. The method was used for the design of two-dimensional airfoils in transonic flow with control on closure. For the design of an airfoil with a weak shock, approximatly 10 iterations which includes 10 exact flow field analyses by TRAFS is needed to obtain convergence to engineering accuracy.

McFadden[30] also solved the inverse problem with the use of an direct analysis method. A modified version of the Bauer-Garabedian-Korn-Jameson(BGKJ) circle plane relaxation program [31] and a functional relationship involving the mapping and the velocity distribution along the airfoil surface between the physical and computational planes are used in an iterative sequence. The method has basically no control over closure. This method is extended for three-dimensional wings in transonic flows by Garabedian and McFadden[32–34]. The method utilizes a version of the FLO-22 analysis code[20] in combination with a relationship between geometry correction and the pressure residual which considers the effects of curvature and slope modifications. Continuous modifications were made to add the design capability for wing leading-edge regions and to expand the design range of configurations by Malone *et al.*[35,36]

Most of the previous efforts in the inverse methods are based on the full potential analysis. Although the potential-flow design procedures are useful and effective as practical design tools due to their reasonable computational resources needed, they have a limitation of the isentropic flow assumption, which leads to an incorrect treatment of shock waves. Recently, the inverse methods gradually broaden their capability to handle this rotational effects and even viscous effects. One of the inverse methods based on the Euler analysis has been studied by Giles and Thompkins[37–39]. The method describes the Newton solution methods for the Euler equations on an intrinsic streamline grid. Other approaches were taken by the research group in ONERA, with Ref. 40 for two-dimensional cascades and ducts, Ref. 41 for axisymmetric nacelles and Ref. 42 for three-dimensional nozzle and jets. This method uses the pseudo-unsteady approach for solving the Euler equations with a special treatment of the slip conditions on a solid wall based on a

characteristic relation method. The Navier-Stokes equations have been also successfuly implemented in the inverse method in Refs. 43 and 44 which are based on the modified Garabedian-McFadden residual-correction design algorithm.

Besides the inverse methods mentioned so far and the numerical optimization methods to be reviewed later, there are several other design approaches in transonic flow regime. One of those is the hodograph method. This method makes use of the fact that the partial differential equations of compressible potential flow are linear when the equations are transformed from the physical plane to the hodograph plane and, thus, linear superposition of fundamental solutions of the hodograph equations can be used for constructing more complex solutions. Since the first successful computation of transonic potential flow about airfoil-like shapes using the hodograph method by Nieuwland[45], many studies, *e.g.*, Refs. 12, 46–52 have been shown that this method could be used to design airfoils with superior transonic performance characteristics with the work of Garabedian and Korn[12] and Boerstoel and Huizing[47] most well known. The methods generally yield excellent results of high accuracy. Very recently a general belief which had been considered as a major drawback of the hodograph method, that the method can only be applied to two-dimensional steady flow problems, appears to be broken and the method was extended to three-dimensional problem in Refs. 53 and 54. Nevertheless, they are not very practical tools for transonic design because they involve complex mappings and require complicated input conditions which are difficult to formulate in general.

The other special approach is the fictitious gas idea by Sobieczky[55–62]. The method is based on the concept of the elliptic continuation of the subsonic part of a mixed subsonic/supersonic flow field into the supersonic zone by utilizing a direct potential-flow analysis with a modified pressure-density relation. The fictitious gas

flow is first calculated for a given body geometry using this modified potential-flow anaysis method and then the correct supersonic flow inside the region of the sonic surfaces is calculated using the solution on the sonic surface as the initial data. In this way the body contour inside the sonic surface could be modified, hence, the method is considered as a shock-free redesign method[59]. This approach is much easier to implement than the hodograph method. However, the important point to note is that the redesign problem does not always have a useful solution and the initial value problem for supersonic domain in three dimensions is considered to be ill-posed[58,61]. The method, generally, is a viable tool for the shock free redesign of a given wing in the final stages of the aerodynamic design process. However, because of the fact that a suitable basic shape is required from the outset, additional tools such as described earlier are necessary to complete the aerodynamic design.

A large branch of design methods in the transonic regime utilizes numerical optimization methods. These methods utilize direct aerodynamic analysis along with a numerical optimization algorithm, to optimize a specified function of aerodynamic performance. The main issues in developing numerical optimization procedures are the selection of design variables, objective function, constraints, particular flow analysis methods and numerical optimization algorithms.

The method of specifying the geometry to be designed and corresponding design variables, is of utmost importance. The choice should be generally directed towards describing a sufficiently wide class of practical solutions and also reducing the computational effort required for exact flow analyses by effectively defining the geometry designed. The first applications of numerical optimization to transonic flow design problems, as performed by Hicks, Murman and Vanderplaats[63–65] using simple polynomial expressions for the geometry definition. In later applications[66–70], the

concept of shape functions describing local geomtry modifications was utilized. A simple polynomial which has the same form of thickness distribution for NACA four-digit airfoils has been used first to define symmetric airfoils by Hicks *et al.*[63] The coefficients of the polynomial are treated as design variables. This idea is extended to the several piecewise polynomial representations for both upper and lower surfaces of the airfoil in Refs. 64 and 65. In Ref. 66, a shape function idea is introduced where the airfoil geometry is represented by the superposition of a baseline airfoil and these shape functions. However, still polynomials are selected to be used as the shape functions. Using arbitrary polynomials to define an airfoil is quite inefficient since the polynomials have basic restrictions in representing various geometric shapes so that the true optimum may be biased. If higher-order polynomials are utilized to avoid this problem, then the computational time will be increased due to the increse of the number of corresponding design variables. The same idea of utilizing simple mathematical bumps, *e.g.*, sine, exponential or polynomials, were employed in three-dimensional wing design in Refs. 69,70 and 79. In Refs. 67 and 68, Vanderplaats takes four existing airfoils as shape functions and the geometry of airfoil is determined by the linear superposition of these four airfoils. The idea is very successful and the results showed that complicated geometries can be generated with much less computational effort mainly due to the reduction of the number of design variables. A more sophisticated and refined idea for shape functions can be found in Refs. 71-73. There, Aidala *et al.*[71] developed several improved shape functions which were more aerodynamic, unlike the physically meaningless bumps or polynomials used in the previous studies. They chose a base airfoil and analyzed it to obtain a baseline pressure distribution. Then this baseline pressure distribution was modified several times in a way that each modification had its own specific aerodynamic function such as smoothing upper surface shock or moving it

forward or aft etc. Using an inverse method, modified airfoil shapes were obtained corresponding to those modified pressure distributions by specifying them as target pressures. Finally, thickness distributions of both the base and modified airfoils were calculated and then substracted each other to result in corresponding shape functions. In application to actual design problem, by selecting necessary shape functions for each particular design problem, they could substantially reduce the computational effort.

In general, the shape functions and design variables are selected carefully for each individual design problem. The formulation of the present study was also set up with considering this fact, thus it does not preclude implementation of more improved shape function idea such as that of Aidala *et al.*[71]

The selection of the objective function and constraints defines the optimization problem. In two-dimensional transonic applications[63–68], typically wave drag has been selected as the objective function, subject to constraints, *e.g.*, minimum cross-sectional area, minimum lift and/or maximum pitching moment. Also lift has been maximized subject to constraints on wave drag and airfoil area[67,68]. The latter was also chosen for the present study since the selection of the same problem will give us a good comparison of performance between different optimization procedures.

Some problems in using numerical optimization methods are associated with nonuniqueness of the solution or local optima. These effects have been partly improved by adding more constraints, using improved objective functions, improved design variables and applying more effective optimization algorithms. In three-dimensional wing design cases, attempts to directly minimize wave drag have been made in Refs. 69 and 74–76. Nonuniqueness effects were present and even more severe than two-dimensional cases. The primary reason for this failure appears to be the inaccuracies in the drag calculations may act on numerically generated *noise*

which relays incorrect gradient information to the optimization process. Some example design problems where inaccurate calculations produced erroneous designs are discussed in Ref. 78. A way to circumvent this problem is to use alternative types of objective functions. One candidate, which is most frequently used for three-dimensional wing design problem, is a pressure distribution type of objective function based on the difference between target pressure and actual computed pressure. However, if this type of objective function is used, one of the important advantages of numerical optimization approach over the inverse method, which is freedom to select a clear design objective for aerodynamic designers instead of handling the more indirect pressure distribution, is lost. Furthermore the inverse method can perform the same work with much less computational effort.

The direct solution for the accuracy problem is to increase the accuracy of computations by developing more efficient algorithms and/or more increasing mesh points. The study of Cosentino and Holst[80] demonstrates clearly that this is true. TWING, the analysis program developed at NASA Ames Research Center by Holst and Thomas[81], uses a fully implicit approximate factorization for solving the full potential equation. This algorithm has a fast convergence rate so that very stringent convergence criterion can be utilized without much increase in computing time. This improves the precision of the wave drag computation dramatically and removes all the previous problems associated with directly minimizing wave drag as the objective function.

Most of previous work in developing numerical optimization procedures are based on the transonic potential-flow analysis. Although recently several applications of the inverse methods with the Euler or Navier-Stokes analysis methods have been addressed, similar efforts have not been implemented using numerical optimization methods in transonic design problems. A major reason for this may

be that the coupling of a more sophisticated transonic flow analysis methods with a numereical optimization algorithm produces prohibitive computational costs. In the present study, two different analysis codes will be utilized. One is the simplest but more approximate transonic small-disturbance(TSD) analysis and the other is a more exact inviscid Euler analysis.

Another issue in developing numerical optimization procedures is the choice of the particular optimization algorithm and the related special optimization techniques which can be more efficient when the algorithm is coupled with complex analysis methods. Unlike the various choices in analysis codes, most of previous work in this area, in particular, with transonic flow analysis codes, has utilized the optimizer CONMIN[82] which is based on the feasible direction method. There may not be any specific reason for this, but only because the program can be easily used and is widely available. So far, no comprehensive studies of performance for various optimizers have been performed for transonic design optimization problems. A few groups are utilizing different optimizer such as QNDIF[83](Quasi-Newton method with DIFferent approximations to the derivatives) by Kennelly[84] and Cosentino and Holst[80] and NEWSUMT-A[87] which is based on the penalty function method with constraint approximations by Joh *et al.*[85] In the present study, the optimizer NEWSUMT-A will be utilized for transonic airfoil optimization based on the successful demonstrations of its implementation in multi-disciplinary wing design problems[89,90]. It may be noted that some exploratory attempts have been made to apply one of the state-of-art optimization algorithms, evolution theory, to a three-dimensional wing design problem as reported by Gregg and Misegades[88]. Results indicate that the algorithm is simple to implement, efficient and relatively insensitive to the number of design variables. However it needs care in determining geometry modifications used during the optimization for a reasonable wing design.

Also the weighting factors to be used in the processing function should be carefully determined to obtain truly optimized wings.

In addition to the selection of a specific numerical optimization algorithm, there are methods to improve the efficiency of these problems by more effectively interfacing an optimization program with an analysis program, as implemented in Refs. 68 and 85. This technique, which is very popular in structural optimization, is called *sequential approximate optimization.* In this technique, the approximate transonic flow analyses are utilized during portions of the optimization process and many of necessary exact analyses can be saved. In Ref. 68, the data base approach can also be used to speed-up the convergence on numerical optimization, where all preceding analysis data are stored and reused to construct higher-order derivative-based approximations to the objective function and constraints. The sequential approximate optimization is also adopted in the present study for improving the efficiency.

After a review of the two major categories in transonic aerodynamic design methods, some comments on the relative strengths and weaknesses of both methods are appropriate. Refs. 77 and 91 address this issue. The major weakness of the inverse methods is that they require the use of considerable design experience in order to effectively specify the pressure distribution and must deal with the *closure problem* associated with the fact that an arbitrary pressure distribution may not result in a closed airfoil shape. Moreover, even if the target pressure distribution results in physically realistic geometry, it takes considerable experience to specify aerodynamically desirable pressure distributions since this prescribed pressure distribution should improve the aerodynamic performance such as reducing drag or increasing lift-to-drag ratio. Another weakness of the inverse methods is that

constraints are difficult to impose. Again, this is due to the fact that aerodynamic designers may not have sufficient intuition and experience in how a specific pressure distribution relates to the constraints. Extensive investigation of inherent problems of inverse methods such as physically unrealistic geometry, existence of solutions and body closure is presented in Refs. 3,4 and 18.

The difficulty of integrating the inverse methods with other engineering disciplines is also one of their major weaknesses. As an example, consider wing design problems involving composite materials and active controls, where there is a close coupling of aerodynamics, structures and control theory. When more than one discipline is involved, experience-based methods, such as inverse approaches, become less attractive. This is in part due to the limited experience of the designer over the wide range of design space of several disciplines, and is also due to the difficulty in developing problem-specific, inverse method software for more than one discipline.

Nevertheless, inverse methods have proved to be very useful, *c.f.*, Ruppert and Goldhammer[92], and most widely used as practical design tools primarily due to their computational efficiency.

Numerical optimization methods, while giving promise to find an optimal airfoil based on specified performance criteria and subject to specified constraints, are less widely used. The methods tend to be computationally costly, and the selection of the appropriate objective function for a two-dimensional design, within a three-dimensional wing flow field is not obvious. Furthermore the methods are sometimes hampered by the appearance of local, rather than global optima.

In general, despite their potential, numerical optimization methods are not commonly used in most aspects of aircraft design, *e.g.*, Ashley[93]. However, in the future this situation will change since the development of new generations of computers will surely make optimization methods more practical.

Numerical optimization methods still retain several important advantages over inverse methods. One of these is their automated design capability which does not require much design experience. Another advantage is that design objectives and constraints can be clearly and freely specified with these methods. For example, objective functions can be selected with constraints suitable for multi-point designs. An example of a two-point design problem directed towards the design of airfoils with low drag creep can be found in Ref. 66. The three-dimensional applications of multi-point designs can be found in Refs. 94-96. The primary geometries are determined by one design point with the optimization used to minimize drag at a second condition. The ability to include various constraints enables the numerical optimization methods handle more complicated design problems, especially those of a multi-disciplinary nature. For example, when the methods are used within a hierarchal framework, *c.f.*, Sobieszczanski-Sobieski[97], they tend to preserve the autonomy of each discipline. Studies in simultaneous aerodynamic-structural design[89,90] bear this out. In Ref. 90, the wing planform shape and the internal structural sizes for a forward-swept, all composite, transport wing are determined by numerical optimization. Through the use of a modular sensitivity analysis, each discipline utilizes the appropriate individual analysis codes, *i.e.*, finite-element for the structure and vortex-lattice for the (low-speed) aerodynamics, while including the complete coupling of the disciplines with a single optimization loop, minimizing the overall wing weight. This research, which only considered planform effects, with a prescribed airfoil shape, formed the motivation for the present work.

We considered the design of transonic airfoils by numerical optimization methods beginning with the work of Hicks *et al.*[63] using a transonic small-disturbance code for the airfoil analysis. The airfoil geometry was approximated by a piecewise

polynomial distribution with 11 coefficients as design variables. They considered several design problems including a transonic lift maximization at $M_\infty = 0.75$, with a constraint on the wave drag, $C_d \leq 0.004$ and a constraint on the non-dimensional airfoil cross-sectional area ratio, $A \geq 0.075$. Results were obtained for this case after an optimization involving 143 iterations.

Computational efficiency issues for airfoil design with optimization has been studied by Vanderplaats and Hicks[67] and Vanderplaats[68]. In Ref. 67, the idea of utilizing *shape functions* for the airfoil geometry was introduced. Designing the airfoil as a superposition of 4 existing airfoil shapes resulted in improved efficiency of the transonic lift maximization problem, requiring 70 complete full-potential global iterations. In Ref. 68, the efficiency issue was further studied using a sequential optimization technique and data base approach. The same transonic lift maximization problem now required 44 global iterations.

The objective of the present effort is to develop efficient numerical optimization procedures for the design of two-dimensional airfoils at transonic speeds, using as few complete transonic analyses as possible. It is also hoped that the methods that are developed here, will be of sufficient computational efficiency to eventually allow their use within the larger framework of multi-disciplinary wing designs. A preliminary study for this effort is reported in Ref. 85, where some special treatments were developed for design optimization based upon the transonic small-disturbance equations. In this study, these ideas will be amplified and improved and the applicability of the methods examined in detail to the more accurate Euler equation analysis.

The detailed formulation of the design problem considered in the present study is described in the following two chapters, the formulation of design problem in

Chapter 2 and formulation for transonic flow analysis in Chapter 3. The design formulation in Chapter 2 discusses the selection of objective function, constraints and specific optimization algorithm(optimizer). The sequential approximate optimization technique is also described in this chapter.

In Chapter 3, the two transonic codes, transonic small disturbance(TSD) and Euler which will be coupled with the optimization algorithm, are described. A detailed discussion is presented about the advantage of the TSD boundary conditions and the necessary modification of the Euler boundary conditions for implementing those methods in the design optimization problem considered here.

In Chapter 4, the approximations of the lift objective function and wave-drag constraint are considered, which play a very important role in the approximate optimization. Several candidates for the approximation methods are tested. Two selected approximations which are the linear approximation for lift and the method of strained coordinates for wave drag are discussed.

The design results based on the TSD method are presented in Chapter 5. The first trial of design with conventional tight move limit strategy which resulted in local optima is investigated. The cause was found to be the noisy behavior of wave drag constraint which resulted from the poor nose-geometry interpolation. A new design strategy is suggested to overcome this local optimum problem, which is interchanging the role of objective function and constraint. A third strategy which is the design with absolute move limits is also tested.

In Chapter 6, the design procedures developed with the TSD method in Chapter 5 are applied and evaluated with the Euler analysis method. Interchanging the role of objective function and constraint is proven to be still an effective strategy for the design based on the Euler solutions where the drag has a smooth behavior with respect to the design parameters. The direct lift maximization with absolute

move limits also works well for designs based on the Euler analysis method. The complete convergence histories for both strategies are presented.

Final conclusions are included in Chapter 7. The important findings and results exhibited during the development of the design optimization procedures in transonic flow regime are summarized in this chapter.

# 2. DESIGN FORMULATION

This chapter describes the specific design problem which will be solved for the present study. The formulation of this design problem includes the selection of design objective function, constraints, design parameters, specific flow analysis method and optimization algorithm. All of these issues are discussed in this chapter except the flow analysis method which will be discussed in Chapter 3. In addition, the implementation of a sequential approximation optimization which is a special technique to improve the efficiency of the design optimization process is also described in this chapter.

## *2.1 Design Problem*

The design problem considered is that of lift maximization subject to constraints on wave drag and airfoil cross-sectional area. The problem is formally stated as

$$\begin{aligned} &\textit{maximize} \quad C_l(\bar{X}) \\ &\textit{such that} \quad C_d(\bar{X}) \leq C_{d_l}\,, \\ &\qquad\qquad A(\bar{X}) \geq A_{min}, \end{aligned} \tag{2.1}$$

where $\bar{X}$ is the vector of design parameters $\bar{X} = (X_1,\ X_2, \ldots, X_N)^T$ specifying the airfoil geometry, $C_d$ is the drag coefficient due to wave drag, $C_{d_l}$ is the prescribed upper limit on wave drag, $A$ is the airfoil cross-sectional area non-dimensionalized by $c^2$, with c the airfoil chord and $A_{min}$ is the minimum required area. The design is performed at a free-stream Mach number of $M_\infty = 0.75$ and zero angle of attack $\alpha = 0$.

This type of problem has been solved first by Vanderplaats and Hicks[67] with a full potential code, requiring 70 complete exact analyses. In Ref. 68, the same problem required 44 exact analyses with a sequential optimization technique and data base approach where all the previous design information is stored and reused for constructing higher-order approximations.

Here, this problem will be solved by utilizing two different codes, one a more approximate transonic small disturbance(TSD) analysis and the second, a more exact inviscid Euler analysis. However, due to the different approximations in the two analyses, particularly the neglect of entropy jumps across the shock waves, the wave drag values are found to be different, with the TSD result at a lower level. In order to develop somewhat similar designs between the Euler and TSD methods it was found necessary to utilize a larger value of $C_{d_l}$ in Eq. 2.1 for the Euler designs. Specifically, a $C_{d_l}$ value of 0.004 for the TSD method and 0.01 for the Euler method were selected. For both analyses, the same value of $A_{min}$, 0.075 was used.

### *2.2 Shape functions and Design Variables*

Vanderplaats and Hicks[67] utilized shape functions to significantly reduce the computational effort of airfoil design optimization, especially when compared to their previous studies using a piecewise polynomial representation[63–66]. The specific form introduced in Ref. 67 is used here, where the non-dimensional airfoil ordinates are considered to be a linear combination of ordinates of existing airfoils as

$$Y = \sum_{i=1}^{N+2} X_i Y_i \left(\frac{x}{c}\right), \tag{2.2}$$

where $Y \equiv y/c$ with $y$ being the airfoil ordinate and $c$ the airfoil chord length. The specified shape functions $Y_i$ are functions of the non-dimensional abscissa $x/c$. The

parameters $X_i$ are the design variables and $N$ is the number of independent design variables. For the design process considered here, four shape functions, $N = 4$, corresponding to the following airfoil shapes were considered:

$$
\begin{aligned}
Y_1 &: NACA\ 2412, \\
Y_2 &: NACA\ 64_1 - 412, \\
Y_3 &: NACA\ 65_2 - 415, \\
Y_4 &: NACA\ 64_2A215.
\end{aligned}
$$

The coordinates are defined at 52 points along the upper and lower surfaces. A cubic spline interpolation based on the arc-length of the airfoil with a periodic boundary condition is used for generating the continuous geometric data of these shape functions. There are two additional shape functions to impose the boundary conditions at the trailing edge of the airfoil. These are $Y_{N+1} = +x/c$ on the upper surface and zero on the lower surface, and $Y_{N+2} = -x/c$ on the lower surface and zero on the upper surface. These six shape functions are presented in Fig. 1. For the design problem based on the TSD analysis, the trailing-edge thickness is specified to be 0.25 percent of the chord length as

$$
\begin{aligned}
(Y_{T.E.}) &= \begin{pmatrix} (y/c)_u \\ (y/c)_l \end{pmatrix}_{T.E.} \\
&= \begin{pmatrix} .00125 \\ -.00125 \end{pmatrix}.
\end{aligned}
\tag{2.3a}
$$

For the design problem based on the Euler method, zero trailing-edge thickness is specified as

$$
(Y_{T.E.}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{2.3b}
$$

These boundary conditions fix the values of the coefficients $X_{N+1}$ and $X_{N+2}$ in terms of $X_1, \ldots, X_N$. If we rewrite Eq. 2.2 in matrix form including the above

boundary conditions at the trailing edge as

$$\overbrace{\begin{pmatrix} \vdots \\ Y_{T.E.} \end{pmatrix}}^{Y} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \\ X_{N+1} \\ X_{N+2} \end{pmatrix}, \tag{2.4}$$

where $Y_{11}$ and $Y_{21}$ contain the first four shape function data and $Y_{12}$ and $Y_{22}$ contain two additional shape function data. $Y_{21}$ and $Y_{22}$ correspond to the ordinates of both upper and lower trailing edge points. Since the vector of design variables $\bar{X} = (X_1, X_2, \ldots, X_N)^T$ is defined by the optimization program, the unknowns are now $(X_{N+1}, X_{N+2})^T$. These can be solved by

$$\begin{pmatrix} X_{N+1} \\ X_{N+2} \end{pmatrix} = [\, Y_{22}]^{-1} (Y_{T.E.}) - [\, Y_{22}]^{-1} [\, Y_{21}] \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{pmatrix} \tag{2.5}$$

Thus the designs considered here utilize four independent design variables. However, the methods implemented here will not preclude the use of additional design variables or different shape functions.

### *2.3 Optimization Method*

When a design optimization is coupled with an expensive numerical analysis code, most of the cost of the optimization is associated with the exact analyses and sensitivity calculations. It is, therefore, relatively less important to consider the efficiency of the optimizer in the selection of a specific optimizer. In general, it is more important to emphasize reliability and robustness. The specific optimizer used for the present study is the general purpose optimization program NEWSUMT-A[87].

Some studies have indicated that the NEWSUMT-A is relatively reliable and robust compared with other optimizers[98].

**NEWSUMT-A**

This program is a general purpose optimization program which is the modified version of NEWSUMT[86] by adding constraint approximations and move-limit strategy. It is based on a quadratic extended interior penalty function and Newton method for unconstrained minimization. The one dimensional minimizations are carried out using the Golden Section method. The program provides the user with several approximation-switching and move limit strategies.

### *2.4 Sequential Approximation Optimization*

The optimization process usually requires evaluating the objective function and constraints hundreds or thousands of times. Even with the most efficient transonic flow analysis code, the cost of the process may be prohibitive if the analysis code and an optimization algorithm are linked together directly so that full analyses are made for all the function evaluations during the design process. Instead, the sequential approximation algorithm[99] is utilized here. This approach replaces the original objective function and constraints with approximations based upon nominal values and derivatives at an initial point. Additionally, move limits are used to prevent the design from moving outside the bound of validity of the approximations. The approximated design problem is given then as :

$$\textit{maximize} \quad C_{l_a}(\bar{X})$$

$$\begin{aligned} \textit{such that} \quad & C_{d_a}(\bar{X}) \leq C_{d_l} \\ & A(\bar{X}) \geq A_{min} \\ & \bar{X}_0 - \Delta\bar{X} \leq \bar{X} \leq \bar{X}_0 + \Delta\bar{X}, \end{aligned} \tag{2.6}$$

where subscript $a$ refer to the approximation, $\Delta\bar{X}$ represents move limits imposed to guarantee the accuracy of the approximation and $\bar{X}_0$ is the initial design. Since $A$, the cross-sectional area, is a linear function of design variables $\bar{X}$, no approximation is needed.

Each approximate optimization problem is solved until an optimum is found. Then a new approximation is constructed there based on the new exact analysis and sensitivity data, and the design optimization process is repeated until convergence is achieved. The flowchart for the design process employing the sequential approximation optimization is presented in Fig. 2. An approximate optimization is typically referred to as an optimization cycle, and this is also the terminology used here.

The remaining work to implement a sequential approximation algorithm includes the selection of exact analysis methods and development of accurate approximation methods. In the next chapter, the transonic flow analysis codes used for evaluating the objective function and constraints will be described. An important part of implementing a sequential approximation algorithm involves the approximation of the objective function and constraints. It has been found that these approximations play a crucial role in the design process. The procedures developed for approximating the lift and drag appear in detail in Chapter 4.

# 3. ANALYTICAL FORMULATION

When analysis methods are considered to be selected for coupling with an optimization approach, a trade off is often made between the complexity of the mathematical model for the flow of interest and the cost of calculation. In this chapter, some consideration on this trade off is made through a brief review of all the numerical transonic flow analysis methods. Two specific analysis methods selected for the present study based on the transonic small-disturbance equation and the Euler equations, are discussed in more detail especially in connection with the nature of boundary conditions when those methods are coupled with an optimization algorithm. An approximate treatment of boundary conditions for the Euler analysis method in order to alleviate the computational effort caused in coupling with a design optimization is also presented.

## *3.1 Governing Equations*

The Navier-Stokes equations are generally accepted as the basic equations governing most fluid dynamic phenomena. The equations are capable of representing mathematically the physical phenomena encountered in transonic flows, including mixed subsonic-supersonic flow, shock waves, boundary layers, and separation. However, if shock waves are relatively weak so that the flow is well behaved and attached, then there are no significant viscous effects due to flow separation and it is possible to neglect the viscous terms entirely. This approximation leads to the Euler equations. The Euler equations in Cartesian coordinates can be written in vector form as

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0, \tag{3.1}$$

where $Q$ represents the conservative quantities and $F$ and $G$ are flux vectors in the $x$- and $y$- directions respectively,

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_0 \end{pmatrix}, \quad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u h_0 \end{pmatrix}, \quad G = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho v h_0 \end{pmatrix}, \tag{3.1a}$$

and for a perfect gas

$$\begin{aligned} p &= (\gamma - 1)\left(\rho e_0 - \frac{\rho\left(u^2 + v^2\right)}{2}\right), \\ h_0 &= e_0 + \frac{p}{\rho}, \end{aligned} \tag{3.1b}$$

where $p, \rho, u, v, e_0$ and $h_0$ represents the pressure, density, Cartesian velocity components, total energy and total enthalpy and $\gamma$ represents the ratio of specific heats. As boundary conditions, the tangency condition of no flow through a solid body and characteristic-based far field conditions are imposed to solve the Euler equations. Additionally the Kutta condition is also imposed to model the real viscous flow for lifting body cases. The Euler equations are the most accurate mathematical form of equations in the inviscid flow. The flow can be rotational.

If the flow is adiabatic and non-heat conducting, then the only mechanism for producing entropy changes in an inviscid fluid is through the presence of a shock wave. However, as long as the shock wave is sufficiently weak, approximately with a local Mach number smaller than 1.3 upstream of the shock wave, the entropy production can be negligible. Furthermore, if the flow at infinity is uniform, then by Crocco's theorem the assumption of isentropy leads directly to the irrotational flow. The main advantage of irrotational flow is that a velocity potential can be

introduced so that a system of Euler equations reduces to a single equation with the only unknown being the velocity potential. Once the velocity potential is known, all the other flow quantities can be calculated according to the relations expressed with the velocity potential.

Final simplification for transonic flow can be made by assuming small disturbances and a Mach number close to unity. For the flow over a thin airfoil where the free stream is only slightly disturbed, the transonic small disturbance(TSD) equation can be systematically derived from the full potential equation. Suppose that the body profile is given in the form of $y = Y(x) = \delta f(x)$, where $\delta$ is a small parameter. If we expand the solution in powers of $\delta$ under the assumption that $1 - M_\infty^2 \simeq O(\delta^{2/3})$ in the transonic range where $M_\infty \simeq 1$, and retain only the leading term, we obtain the TSD equation.[100] The typical form of the TSD equation may be written either in physical variables

$$\left[(1 - M_\infty^2)\Phi_x - \frac{(\gamma+1)}{2} M_\infty^K \Phi_x^2\right]_x + \Phi_{yy} = 0, \tag{3.2a}$$

or in similarity variables

$$\left[K\phi_x - \frac{(\gamma+1)}{2}\phi_x^2\right]_x + \phi_{\tilde{y}\tilde{y}} = 0, \tag{3.2b}$$

where $\Phi, \phi$ are, respectively, the physically and transonically scaled pertubation potentials defined as

$$\Phi^* = cU_\infty(x + \Phi + \ldots) = cU_\infty(x + \delta^{2/3} M_\infty^{-n}\phi + \ldots), \tag{3.3}$$

and $\Phi^*$ is the total potential, $\delta$ is usually taken as the maximum thickness of airfoil, $c$ is the airfoil chord, $M_\infty$ is the free-stream Mach number, $x$ and $y$ are the Cartesian coordinates and

$$\tilde{y} = \delta^{1/3} M_\infty^m y, \tag{3.4}$$

$$K = \frac{1 - M_\infty^2}{M_\infty^{2m}\,\delta^{2/3}}, \tag{3.5}$$

where $K$ is the transonic similarity parameter. Both the exponents $m$ and $n$ basically modify or tune-up the small-disturbance theory to provide more accurate approximation of solutions to the full potentail equation over a wide range of $M_\infty$ and $\delta$. This TSD equation is the simplest equation that can describe a transonic flow. In addition to the governing equation, the boundary conditions at both surface and far field must also be applied. The linearized form of the surface boundary condition applied to the axis $y = 0$ instead of the airfoil surface is consistent with the TSD approximations. For the lifting case, the circulation must be imposed and determined by satisfying the Kutta condition.

For the present study, two different analysis methods are selected. The first is a more approximate and consequently more simple and efficient numerical flow anlysis method, which is based upon the transonic small disturbance equation. The other is the more exact inviscid flow analysis method based upon the Euler equations for the validation of the procedures developed first. Although the actual results of our design studies will depend upon the specific form of the analysis method chosen, it is expected that most of the conclusions on relative computational efficiency obtained with a simple numerical analysis method can be applied to more complicated analyses.

### *3.2 Transonic Small Disturbance(TSD) Analysis*

Since our primary emphasis for the first effort was placed on the efficiency, the transonic small disturbance method was selected. One reason to use the simple transonic analysis code at first is basically to relieve the computationl burden since it usually requires huge number of repeated runnings of the analysis code to develop a design optimization procedure. The efficiency of the TSD analysis over full potential theory is mainly due to its linearized surface boundary condition. Since

this boundary condition is applied to the $x$-axis not curved surface of the body, a complicated body conforming grid system need not to be generated to apply the boundary condition. The other reason is that it is also hoped that the methods developed here would be sufficiently efficient as to be employed within the much more complex framework of a multi-disciplinary aircraft design. Thus efficiency is the most crucial factor. Although the TSD method is the simplest theory for transonic flow, it still retains the capability to simulate the most important features of the transonic flow such as mixed nonlinear type of flow, shock waves and even wave drag prediction. Clearly, Eq. 3.2b has the form locally either of hyperbolic type, representing supersonic flow $\phi_x > K/(\gamma+1)$, or of a elliptic type, representing subsonic flow $\phi_x < K/(\gamma+1)$. If we introduce conservative discretizations into Eq. 3.2, the solution will satisfy the weak form and consequently discontinuities in the solution will be correctly captured. In the exact invscid flow theory(Euler equations), it is well known that wave drag results from the entropy production by the shock waves. In the TSD equation, it is also possible to approximate the wave drag by first-order entropy production through weak shocks from the first-order theory [101]. Thus the TSD equation can efficiently simulate the mixed elliptic -hyperbolic type flow with shock waves through which the wave drag can be evaluated.

The TSD equation contains some sacrifices in accuracy due to the assumption of thin airfoils during the derivation. For example, in the immediate neighbourhood of blunt leading edges, the solution will be inaccurate because the pertubation velocity is of the same order as the free stream velocity. With this assumption, there is some simplification of the governing equation. In addition, much more attractive advantage accrues in the application of boundary conditions. The surface boundary condition on the airfoil surface $y = Y(x)$ is expanded to be

$$\frac{\partial \phi}{\partial y}(x,0) = \frac{dY}{dx}, \qquad 0 \le x/c \le 1\,. \tag{3.6}$$

This surface boundary condition for two-dimensional problems is applied along the $x-$axis not on the actual surface of the airfoil. Thus the numerical problem is greatly simplified since complex body aligned mappings will not be needed for the application of boundary conditions. A simple Cartesian grid system is enough to perform the TSD calculation. This results in significant reductions in computational effort particularly in the design problems with changing body geometries. The initial grid system can be fixed and utilized repeatedly throughout the whole design process for different body geometries designed consecutively. In particular the computer program TSFOIL[102,103] is used for the TSD analyses.

### *3.3 TSD Simulation - TSFOIL*

#### *3.3.1 Description of TSFOIL*

TSFOIL[102,103] is a program for solving the two-dimensional transonic small disturbance equation for lifting airfoils. It predicts main nonlinear features of transonic flow and also has various numerical options including wave drag calculation. The similarity notation(Eq. 3.2b) is adopted to properly scale the variables and to provide consistency with the theory developed. Three scaling rules by Cole, Spreiter and Krupp are provided in the program; The Krupp scaling is used actually for the present study, where $K = 7/4$, $n = 3/4$ and $m = 1/2$. A successive line over relaxation(SLOR) algorithm is used to solve the system of finite difference equations which are in the form of fully conservative to guarantee that the correct shock jumps are calculated. The default Krupp mesh is used, which is a Cartesian mesh with the clustering near the leading and trailing edge of the airfoil as shown in Fig. 3. Mesh refinement technique($20 \times 14$ on the coarse, $39 \times 28$ on the medium and $77 \times 56$ on the final mesh) is used to accelerate the convergence.

An exact analysis of the transonic flow around an airfoil requires approximately 10 – 15 CPU seconds on IBM 3090 with optimization level 3. It was not attempted to enhance the convergence of TSFOIL by converting the solution algorithm to AF(Approximate Factorization)[104,105] or adopting the multigrid techniques[106,107]. More details about TSFOIL can be found in Refs. 102 and 103.

### *3.4 Analysis Using the Euler Equations*

The assumption of potential flow implies that the flow is irrotational. This is not strictly correct when shock waves are present. An exact description of transonic inviscid flow requires the solution of the Euler equations subject to appropriate surface and far field boundary conditions.

The treatment of the far field condition is usually based on the introduction of Riemann invariants for a one dimensional flow normal to the boundary[108]. Let subscripts $\infty$ and $I$ denote far field and interior of the computational domain. Assuming that the flow is subsonic at infinity, we introduce fixed and extrapolated Riemann invariants

$$R_\infty^- = \bar{u}_\infty - \frac{2\,a_\infty}{\gamma - 1}, \tag{3.7}$$

$$R_I^- = \bar{u}_I + \frac{2\,a_I}{\gamma - 1}, \tag{3.8}$$

corresponding to incoming and outgoing characteristics. These may be added and substracted to give

$$\bar{u} = \frac{1}{2}(R_I^+ + R_\infty^-), \tag{3.9}$$

$$a = \frac{\gamma - 1}{4}(R_I^+ - R_\infty^-), \tag{3.10}$$

where $\bar{u}$ and $a$ are the actual normal velocity component and speed of sound to be specified in the far field. At an outflow boundary, the tangential velocity component

and entropy are extrapolated from the interior, while at an inflow boundary they are specified as having free stream values. These four quantites provide a complete definition of the flow in the far field. If the flow is supersonic, all the flow quantities are specified at an inflow boundary, and they are extrapolated from the interior at an outflow boundary.

The surface boundary conditions for the Euler equations, representing no flow through the solid surface may be expressed as

$$\bar{V} \cdot \hat{n} = 0, \tag{3.11a}$$

or

$$v(x, Y) = Y'(x)\, u(x, Y), \qquad 0 \leq x/c \leq 1, \tag{3.11b}$$

where $\bar{V}$ is the velocity, $\hat{n}$ is the unit normal vector at the surface and a prime denotes differentiation with respect to $x$. The only contribution to the flux balance comes from the pressure. The other three conditions are usually

$$\frac{\partial p}{\partial n} = 0, \tag{3.12}$$

$$\frac{\partial \rho}{\partial n} = 0, \tag{3.13}$$

$$\frac{\partial h_o}{\partial n} = 0. \tag{3.14}$$

Sometimes the normal pressure gradient can be estimated from the integration of the normal momentum equation[109] and zero entropy gradient can be used instead of Eq. 3.13. These surface boundary conditions for Euler equations are applied on the curved surface of the body, not just on the slit lying in the $x-$axis as was done for the TSD analysis. This requires that the discretization of the field must conform to the surface boundary so that the boundary conditions can be accurately imposed. It introduces, besides the complexity of the equations, a more complicated problem, where a boundary-conforming grid system should be generated.

### *3.5 Modified Euler Boundary Conditions for Airfoil Design*

#### *3.5.1 Background*

As mentioned in the previous section, the Euler equations require a body conforming grid system. This adds more complexity to the design problem with the Euler method compared to the one with the TSD method in two aspects. First, in each design iteration, new sets of grid systems corresponding to both the nominal and perturbed geometries of the design are generated for exact analyses and sensitivity calculations. This is not a trivial work especially for the design optimization problem where the body geometry is constantly changing. Secondly, since the grid system is changing due to the change of body geometry, it would be complicated to calculate the sensitivities of solution at fixed grid points of computational domain. The latter is very true particularly in the implementation of the method of strained coordinates which needs sensitivities of the solution on the body surfaces for the Euler analysis method(discussed in section 4.3). The computational burden of regenerating the grid for each geometry is reduced by assuming that design changes proceed slowly and for a specified number of cycles, consider the grid to be fixed to a baseline airfoil geometry. Then the surface boundary condition would have to be altered to allow a small amount of mass transpiration through the surface to approximately account for the changing geometry. This procedure fits in well with the sequential approximate optimization algorithm used in the design process, which imposes move limits on the design. For each optimization cycle a baseline geometry will define the grid and the grid will be kept fixed throughout the approximate optimization cycle. As will be discussed later in Chapters 5 and 6, approximately 10 cycles will be used to obtain converged designs, so that the grid will be changed only a corresponding 10 times. This greatly reduces the computational effort for the Euler designs. The details of the derivation of the boundary condition now follows.

### *3.5.2 Modification of the Euler Boundary Conditions*

Let the subscripts $b$ and $p$ refer to the baseline airfoil and perturbed airfoil respectively. The exact surface boundary condition for the baseline airfoil is

$$\bar{V}_b \cdot \hat{n}_b = 0, \tag{3.15}$$

and for the perturbed airfoil

$$\bar{V}_p \cdot \hat{n}_p = 0. \tag{3.16}$$

Now we try to solve the flow around the perturbed geometry using the baseline grid. Since we utilize the grid system which was generated according to the surface condition of Eq. 3.15, we need some values for the flux across the original baseline grid surface which will not be zero.

If we denote the airfoil geometry function as $Y(x)$, then the body surface equation for the baseline airfoil is written by

$$F(x,y) = y - Y_b(x), \tag{3.17}$$

and the normal velocity at $Y_b$ becomes

$$\begin{aligned} V_{n_b} &= \bar{V}_b \cdot \hat{n}_b \\ &= \frac{1}{\sqrt{1+Y_b'^2}} (v_b - u_b Y_b'), \end{aligned} \tag{3.18}$$

since the unit normal vector at the surface $Y_b$ defined by

$$\begin{aligned} \hat{n}_b &= \frac{\nabla F}{|\nabla F|} \\ &= \frac{1}{\sqrt{1+Y_b'^2}} (Y_b'\hat{i} + \hat{j}), \end{aligned} \tag{3.19}$$

where $\hat{i}$ and $\hat{j}$ are unit vectors in the Cartesian coordinates. On the other hand, Eq. 3.16 can be written for the perturbed airfoil $Y_p$ as

$$\frac{v_p}{u_p} = Y_p'(x)\,. \tag{3.20}$$

Now, if we expand the velocity at the baseline airfoil surface about the perturbed airfoil surface, then

$$v_b = v_p + \left(\frac{\partial v}{\partial y}\right)_p (Y_b - Y_p) + \dots\,, \tag{3.21a}$$

$$u_b = u_p + \left(\frac{\partial u}{\partial y}\right)_p (Y_b - Y_p) + \dots\,. \tag{3.21b}$$

Substituting these equations into Eq. 3.18 and neglecting higher order terms, we can obtain an approximate expression for the normal velocity at the surface $Y_b$ as

$$V_{n_b} = \frac{u_p}{\sqrt{1+Y_b'^2}}\,(Y_p' - Y_b')\,. \tag{3.22}$$

The tangential velocity at the surface $Y_b$ is

$$\begin{aligned} V_{t_b} &= \bar{V}_b \cdot \hat{t}_b \\ &= \frac{1}{\sqrt{1+Y_b'^2}}\,(u_b + v_b Y_b')\,, \end{aligned} \tag{3.23}$$

where $\hat{t}_b$ is the unit tangential vector at the surface $Y_b$ defined by

$$\hat{t}_b = \frac{1}{\sqrt{1+Y_b'^2}}\,(\hat{i} + Y_b'\hat{j})\,. \tag{3.24}$$

Similarily, using Eqs. 3.20, 3.21 and neglecting higher order terms, we can approximate the tangential velocity at the surface $Y_b$ by

$$\begin{aligned} V_{t_b} &= \frac{1}{\sqrt{1+Y_b'^2}}\,(u_p + v_p Y_b') \\ &= \frac{u_p}{\sqrt{1+Y_b'^2}}\,(1 + Y_b' Y_p')\,. \end{aligned} \tag{3.25}$$

Eliminating $u_p$ from Eq. 3.22 and Eq. 3.25 leads to the following modified boundary condition at the surface $Y_b$,

$$V_{n_b} = \frac{V_{t_b}}{1 + Y_b' Y_p'} (Y_p' - Y_b') . \tag{3.26}$$

Thus, the modified boundary conditions which account for mass transpiration at the surface of a baseline geometry may be written as:

$$\begin{aligned} \rho &= \rho_b , \\ \rho V_t &= (\rho V_t)_b , \\ \rho V_n &= (\rho V_n)_b , \\ \rho e_0 &= \left[ \frac{p}{\gamma - 1} + \frac{\rho (V_n^2 + V_t^2)}{2} \right]_b , \end{aligned} \tag{3.27}$$

where $p_b$ is evaluated from integration of the normal momentum equation, $\rho_b$ and $(\rho V_t)_b$ are extrapolated from field points and $(\rho V_n)_b$ calculated from Eq. 3.26. The specific analysis code used for the Euler calculations is FLOMG[110,111].

## *3.6 Euler Simulation - FLOMG*

### *3.6.1 Description of FLOMG*

FLOMG is a program for solving the two dimensional Navier-Stokes equations developed by Swanson and Turkel[110,111]. However, here, it was utilized only in the inviscid, Euler solver mode. The finite volume formulation for space discretization is used with the addition of artificial dissipative terms. The program adapted Jameson's multistage time-stepping scheme[112] in conjunction with multigrid algorithms[113] to produce a rapid convergence. Various devices are employed in the program to improve the rate of convergence such as a local time step with constant CFL number throughout whole computational domains, an enthalpy damping

in which a forcing term is proportional to the difference of the total enthalpy between local and free stream and an implicit residual smoothing technique [114]. In the analyses for the present study, a W-cycle full multigrid method is used with five multigrid levels($14 \times 2$, $28 \times 4$, $56 \times 8$, $112 \times 16$ and $224 \times 32$ cells). Fifty multigrid cycles are first run on a mesh with $112 \times 16$ cells. This is used as the starting point for a multigrid run of 200 cycles on the $224 \times 32$ cell mesh. The grid system used for the present study is the C-type hyperbolic grid (Fig. 4). A CFL number of 7.5, no enthalpy damping and the five stage Runge-Kutta type time-stepping are used also. The CPU time taken in an exact analysis for our flow problem with Mach number 0.75 and zero angle of attack is approximately 40 seconds on a CRAY-2S.

### *3.6.2 Implementation of Modified Euler Boundary Conditions in FLOMG*

The technique of applying the surface boundary conditions in FLOMG is based upon reflection. In implementing reflection, the body surface lies on the cell interfaces between the first cells above the surface and some fictitious cells interior to the body(Fig. 5a). The body surface values of the flow variables are evaluated as the average of the values in the cells on the two sides of the face. Values of the primitive variables at cell centers of the sublayer are assigned using the reflection process. It is also assumed that the density and tangential velocity are even functions of the normal distance above the body while the normal velocity is assumed to be an odd function of the normal distance. The density and tangential velocity at the sublayer cell centers are set equal to their respective values at the first cell centers above the body while the normal velocity is set equal to the negative of its value at those cell centers. In FLOMG, the normal pressure gradient is evaluated by the integration of the normal momentum equation[109] as

$$[(x_\xi^2 + y_\xi^2)p_\eta]_{body} = [\rho(uy_\eta - vx_\eta)(vx_{\xi\xi} - uy_{\xi\xi}) + (x_\xi x_\eta + y_\xi y_\eta)p_\xi]_{j=2} \tag{3.28}$$

where $(\xi, \eta)$ is the generalized curvilinear coordinate system and $x_\xi, y_\xi, x_\eta, y_\eta, x_{\xi\xi}$ and $y_{\xi\xi}$ are metric derivatives between the curvilinear and Cartesian coordinate systems. The reflection process for implementing the modified Euler surface boundary conditions based on Eqs. 3.26–3.28 is

$$\begin{aligned} \rho_1 &= \rho_2 \,, \\ (\rho V_t)_1 &= (\rho V_t)_2 \,, \\ (\rho V_n)_1 &= -(\rho V_n)_2 + 2(\rho V_n)_b \,, \\ (\rho e_0)_1 &= \left[ \frac{p}{\gamma - 1} + \frac{\rho(V_n^2 + V_t^2)}{2} \right]_1 \,, \end{aligned} \tag{3.29}$$

where

$$\begin{aligned} p_1 &= 2\, p_b - p_2 \,, \\ (\rho V_n)_b &= \frac{(\rho V_t)_2}{1 + Y_b' Y_p'} \left( Y_p' - Y_b' \right) . \end{aligned} \tag{3.29a}$$

To evaluate $p_b$ we first evaluate $p_p$ by applying Eq. 3.28 to the surface of perturbed geometry $Y_p$, since this equation is only valid for the solid surface. Then the pressure at the baseline grid surface $Y_b$ can be obtained by using Taylor series expansion as

$$p_b = p_p + p_\eta |_p \left( \eta_b - \eta_p \right) . \tag{3.30}$$

This value of $p_b$ is needed to balance the fluxes in the evaluation of quantities at cell center 2 in Fig. 5c.

# 4. LIFT AND DRAG APPROXIMATIONS

The transonic airfoil design problem considered here is expected to be highly nonlinear. This is because the lift and wave drag will be nonlinear functions of the design variables even if the aerodynamic analysis is linear, but the transonic flow analysis considered here is highly nonlinear due to the presence of shock waves. As mentioned in Chapter 2, the sequential approximation optimization is a very good choice for this type of a nonlinear constrained optimization problem which involves expensive objective function and constraint evaluations. However, in order to successfully implement the sequential approximation optimization technique for the design problem considered here, the selection of sufficiently accurate approximation methods for both the objective function and constraints is necessary. In this chapter, several approximation methods for lift and wave drag are tested. The linear approximation for lift and the method of strained coordinates for wave drag are finally selected. In particular, the basic idea and specific implementaion procedures of the method of strained coordinates are described in detail.

## *4.1 Linear Approximation of Lift*

A linear approximation for the lift, which is the simplest derivative-based approximation and most frequently used in the approximate optimization is considered first. If the linear approximations are employed for both objective function and constraints, the problem considered here turns to a sequential linear programming(SLP) problem. This would be very attractive in the aspect of utilizing a very robust and efficient optimizer since LP packages are usually more reliable than the nonlinear optimization packages and also readily available to most computer

users through system library packages now. A large part of the computational cost associated with the optimization process itself can be saved, in comparison with nonlinear programming, even though the total design cost can not be predicted before performing actual designs, since it also should account for the number of design iterations required for convergence. The reformulated linear problem is also much easier to handle in aspects such as checking the convergence or controlling move limits, etc. Consequently, if the approximation is proven to be accurate in the wide range of the design space, then the convergence of the sequential linear programming will be very efficient. However this would not be true for the problem considered here because the problem involves highly nonlinear transonic flow with shock waves, which especially affects the wave drag constraint.

The linear approximation for lift is expressed by Taylor series expansion as

$$C_l = C_l^0 + \sum_{i=1}^{4} \frac{\partial C_l}{\partial X_i} (X_i - X_i^0), \tag{4.1}$$

where $C_l^0$ is the lift coefficient evaluated with the initial design parameters $X_1^0, \ldots, X_4^0$. The sensitivity derivatives, $\partial C_l / \partial X_i$ is evaluated using forward finite differences here as

$$\frac{\partial C_l}{\partial X_i} \simeq \frac{C_l(\ldots, X_i^0 + \Delta X_i, \ldots) - C_l(\ldots, X_i^0, \ldots)}{\Delta X_i}, \qquad i = 1, \ldots, 4 \tag{4.1a}$$

where $\Delta Xi$ is a small perturbed amount of $X_i$, which is taken as 0.01. The effectiveness of the lift approximation based on both the TSD and Euler solutions are indicated in Tables 1a and 2a. In both tables, $C_l^0$ and $\partial C_l / \partial X_i$ for $i = 1, \ldots, 4$ are numerically evaluated using Eq. 4.1a. Eq. 4.1 is used to approximate $C_l$ at a number of different values of design variables ranging from 0.5% to 2.0% of the initial design. Comparisons between the approximate lift coefficient and the exact

values for the TSD solutions indicate that the approximation is well-behaved, with errors of less than 2% when the design variables vary by as much as 2%. Note the 2% change in the design variables corresponds to a 4.2% increase in the maximum thickness of the airfoil. The linear approximation based on the Euler solutions shows much higher accuracy, with errors of less than 0.5% when the design variables vary by as much as 2%. Thus the linear approximation is selected to be utilized for the lift approximation.

### *4.2 Linear and Higher Order Approximation of Drag*

#### *4.2.1 Linear Approximation of Drag*

The linear approximation for the wave drag is also denoted by Taylor series expansion

$$C_d = C_d^0 + \sum_{i=1}^{4} \frac{\partial C_d}{\partial X_i} (X_i - X_i^0) . \tag{4.2}$$

The good agreement shown in the lift approximation is not repeated for the drag approximation. In Tables 1b and 2b, the results of four $C_d$ approximations labelled $C_d - 1, \ldots, C_d - 4$ varying over the same range of design variables as in Table 1a or 2a are presented. The linear drag approximation based on the TSD solutions , denoted $C_d - 1$ in Table 1b, does not correlate closely with the exact results, with errors of as much as 25% when the design variables change by 2%. The linear drag approximation using the Euler solutions shows better correlations with the exact results in Table 2b. However the approximation still does not seem to have small errors enough to be employed in the approximate optimization, which are as much as 15 % in 2% changes of design variables. This result was not unexpected.

### *4.2.2 Higher Order Approximation of Drag*

It is natural to consider higher order derivative-based approximations to obtain higher approximation accuracy. The quadratic approximation for wave drag is obtained by including the quadratic terms in the Taylor series expansion

$$C_d = C_d^0 + \sum_{i=1}^{4} \frac{\partial C_d}{\partial X_i}(X_i - X_i^0) + \frac{1}{2}\sum_{i=1}^{4}\sum_{j=1}^{4} \frac{\partial^2 C_d}{\partial X_i \partial X_j}(X_i - X_i^0)(X_j - X_j^0)\,. \tag{4.3}$$

However the cost of evaluating all the second-derivative sensitivities increases dramatically for the design problem considered here (in each design iteration, 15 exact analyses are needed to construct a complete quadratic approximation).

Vanderplaats[68] overcame this computational burden by employing a global sense of approximation reusing all of the previous design information. His approximation method is basically different from the usual derivative-based approximation in terms of how the sensitivity derivatives are evaluated. He considers Eq. 4.3 as a quadratic polynomial approximation, with the sensitivity derivatives in Eq. 4.3 considered as unknown coefficients of the quadratic polynomial. The previous analysis data are used with curve-fitting procedures to determine the sensitivity derivatives in Eq. 4.3.

These approximate polynomials are also updated during the optimization. At the initial stage of the design, where the number of analyses is less than the number of unknowns in Eq. 4.3, only fewer coefficients are calculated. As the design iteration proceeds, if more than 15 designs are available, a weighted least squares fit is used in which designs nearest the nominal are weighted most heavily.

In the present study, the idea of polynomial-based global approximations also will be considered, but with different forms. The previous design information will not be used here.

The next approach considered is a global sense of approximation based on the quadratic polynomial of $C_d$ in terms of $C_l$.

$$C_d = A_0 + A_1\, C_l + A_2\, C_l^2\,, \tag{4.4}$$

where the parameters $A_0$, $A_1$ and $A_2$ were determined from a least-squares fit to $C_l$ and $C_d$ data about the initial design $\bar{X}^0$ and four perturbed designs about the initial state of each design variable. This method, denoted $C_d - 2$ in Tables 1b and 2b, actually degraded from the approximation in Eq. 4.2.

Next, the effect of shock movement on the drag coefficient was attempted to be considered. A linear approximation of the shock location was assumed, $x_s$ as

$$x_s = x_s^0 + \sum_{i=1}^{4} \frac{\partial x_s}{\partial X_i}\,(X_i - X_i^0)\,, \tag{4.5a}$$

along with a globally quadratic drag approximation

$$C_d = A_0 + A_1\, C_l + A_2\, C_l^2 + A_3\, x_s\,. \tag{4.5b}$$

The coefficients $A_0, \ldots, A_3$ were determined as in Eq. 4.4. This method, denoted $C_d - 3$ in Tables 1b and 2b, also did not produce acceptable approximations.

### *4.3 Drag Approximation Using Coordinate Straining*

Since the simplest methods do not appear to be adequate for the approximation of wave-drag in the nonlinear transonic regime, and since shock-wave movement obviously must play an important role in the design process, another approximation method employing the *method of strained coordinates* has been implemented. The approach utilized here was introduced by Nixon[115,116] for perturbations of transonic flows with shock waves, and has been applied for airfoil approximations by Stahara *et al.*[117] In this method, the perturbations are made in a *strained* coordinated system where the shock remains fixed. This approach is utilized to approximate the flow velocities. The wave drag is then calculated from the exact expression with the approximate flow velocities.

*4.3.1 Linear Approximation of Solution*

The basic hypothesis underlying the procedure is that a range of solutions in the vicinity of a previously calculated or base solution can be approximated to first order accuracy in the incremental change of the varied parameter by the relation

$$Q \simeq Q_0 + Q_p \Delta X , \tag{4.6}$$

where $Q_p$ is a linearized unit perturbation solution, $Q$ is the approximate solution for conditions differing from those of the base solution $Q_0$ by $\Delta X$ which is an amount of change in the parameter $X$. If the parameter $X$ corresponds to some arbitrary flow quantity or geometry, then $Q_p$ is a sensitivity derivative of the solution with respect to these quantites. In the present study, we change the geometry of the airfoil which is defined by four geometric parameters during the optimization and calculate the corresponding solution. Thus Eq. 4.6 can be extented to the four parameter pertubation with the form of

$$Q = Q^0 + \sum_{i=1}^{4} \frac{\partial Q}{\partial X_i} (X_i - X_i^0) . \tag{4.7}$$

Comparing with Eq. 4.1 or Eq. 4.2 shows that this is a linear approximation of solution instead of more global quantities $C_l$ or $C_d$. Usually in the structural optimization, it is also a popular alternative to find this kind of *intervening variables* that make the approximated function more linear when the direct linear approximations are not accurate satisfactorily[118].

This simple procedure, on the other hand, only works directly for continuous flows for which the perturbation changes does not alter the solution domain. For discontinuous flow such as the present transonic flow including shock waves, *coordinate straining* is necessary to account additionally for movement of discontinuities due to the parameter change.

### *4.3.2 Coordinate Straining*

The concept of employing coordinate straining to remove nonuniformities from pertubation solutions of nonlinear problems is well established and originally suggested by Lighthill[119]. The first application of the concept of coordinate straining to the present transonic flow perturbation was done by Nixon[115,116]. The fundamental idea of solution approximation using coordinate straining is illustrated in Fig 6. The two different graphs can be regarded as related nonlinear flow solutions separated by a nominal change in some geometric or flow parameter. We observe that the perturbation between two solutions is small everywhere except in the region between the two shock waves(the top picture in Fig. 6). This clearly invalidates the pertubation technique around that region. Thus, the procedure is to strain the coordinates of one of the two solutions in such a fashion that the shock waves align and then determine the sensitivity of the solution(the second picture). The approximate solution is, then, extrapolated based on both the sensitivity and the given amount of pertubation through another coordinate straining (the third picture). In the figure, only a single shock wave is considered. However the procedure can be straightforwardly extented to the multiple shock waves as well as high gradient locations such as stagnation points, maximum suction pressure points, etc.

### *4.3.3 Approximation Procedure with the TSD Simulation*

When the TSD analysis is involved in the approximation with the method of strained coordinates, the flow velocities are first approximated and then the wave drag is calculated from the exact expression with the approximate flow velocities. Coordinate straining uses the axial position of the shock wave on the airfoil, $x_s$ and the height of the tip of the shock wave above the airfoil, $y_t$. When $x_s$ changes by

$\Delta x_s$ and $y_t$ changes by $\Delta y_t$, the coordinates $(x, y)$ of each point in the flow field are changed by $\Delta x$ and $\Delta y$ given by

$$\Delta x = s(x)\,\Delta x_s\,, \tag{4.8a}$$

$$\Delta y = \frac{y}{y_t}\,\Delta y_t\,. \tag{4.8b}$$

For the TSD analysis, following Ref. 114, the straining function $s(x)$ is taken as

$$s(x) = \frac{x(c-x)}{x_s(c-x_s)}\,, \tag{4.9}$$

which is valid for $0 \leq x \leq c$ and is equal to zero otherwise.

The strained coordinates are used first in the process of calculating finite-difference sensitivity derivatives according to the following steps:

1. The i-th design variable is perturbed by $\Delta X_i$ and the corresponding $\Delta x_s$ and $\Delta y_t$ are calculated from a new solution of the flow field and used to approximate the shock sensitivities

$$\frac{\partial x_s}{\partial X_i} \simeq \frac{\Delta x_s}{\Delta X_i}\,, \tag{4.10a}$$

$$\frac{\partial y_t}{\partial X_i} \simeq \frac{\Delta y_t}{\Delta X_i}\,. \tag{4.10b}$$

2. The flow-field sensitivities are approximated, using $\Delta x$ and $\Delta y$ from Eqs. 4.8a and 4.8b, in terms of the axial velocity for the nominal design $u^0$ and the axial velocity for the perturbed flow field $u$ as

$$\frac{\partial u}{\partial X_i}(x,y) = \frac{1}{\Delta X_i}[u(x+\Delta x, y+\Delta y) - u^0(x,y)]\,. \tag{4.11}$$

The strained coordinates are used again in approximating the flow at a new design point $\bar{X}$ as follows: First the new shock location and shock-tip position are calculated by a linear approximation

$$\Delta x_s = \sum_{i=1}^{4} \frac{\partial x_s}{\partial X_i}\,(X_i - X_i^0)\,, \tag{4.12a}$$

$$\Delta y_t = \sum_{i=1}^{4} \frac{\partial y_t}{\partial X_i}\,(X_i - X_i^0)\,. \tag{4.12b}$$

Values of $\Delta x$ and $\Delta y$ are calculated from Eqs. 4.8a and 4.8b and then the axial velocity $u$ is estimated from

$$u(x+\Delta x, y+\Delta y) = u^0(x,y) + \sum_{i=1}^{4} \frac{\partial u(x,y)}{\partial X_i} \left(X_i - X_i^0\right). \tag{4.13}$$

The pressure coefficient may then computed from the velocities, which for the transonic small-disturbance scaling used here, is

$$C_p(x,y) = -2\,\delta^{2/3} M_\infty^{-3/4}\, u(x,y)\,, \tag{4.14}$$

where $\delta$ is the nominal airfoil thickness, and $M_\infty$ is the free-stream Mach number. Finally the wave drag coefficient is determined from a contour integral of the jump in pressure across the shock, which may be written as[102]:

$$C_d = -\delta^{5/3} M_\infty^{-3/4} \frac{(\gamma+1)}{\gamma} \int_{shock} [u]^3\, dy\,, \tag{4.15}$$

where $\gamma$ is the ratio of specific heats and $[u]$ is the *jump* in $u$ across the shock.

### *4.3.4 Approximation Procedure with the Euler Simulation*

For the Euler analysis a different implementation of the method of strained coordinates is utilized. First, since the wave drag is calculated by integrating surface pressures, only the solution at the body surface needs to be approximated. The coordinate straining in $y$-direction is not necessary. Furthermore, accurate solutions for the Euler wave drag were obtained by approximating the surface pressures directly, rather than computing from the flow field. Thus the surface pressure sensitivity was calculated as

$$\frac{\partial p}{\partial X_i}[x, Y_b(x)] = \frac{1}{\Delta X_i}\left(p[x+\Delta x, Y_b(x+\Delta x)] - p^0[x, Y_b(x)]\right)\,, \tag{4.16}$$

where $\Delta x$ is the amount of coordinate straining. The approximate surface pressure is then estimated from

$$p[x + \Delta x, Y_b(x + \Delta x)] = p^0[x, Y_b(x)] + \sum_{i=1}^{4} \frac{\partial p[x, Y_b(x)]}{\partial X_i} (X_i - X_i^0) . \qquad (4.17)$$

The pressure on the perturbed geometry can be approximated by using Taylor series expansion about this pressure on the baseline geometry and then the wave drag coefficient is determined from the integration of the approximate surface pressure force in $x$ direction.

The non-uniqueness of the straining function has been discussed in Refs. 115–117. Here, several tests was made to select the best straining function in terms of approximation accuracy for the Euler analysis. The linear piecewise continuous straining appeared to be most accurate and well-behaved. This has been also pointed out with the full potential method in Ref. 117. The linear piecewise continuous function is given by

$$s(x) = \begin{cases} x/x_s & ; 0 \le x \le x_s , \\ (c - x)/(c - x_s) & ; x_s \le x \le c , \end{cases} \qquad (4.18)$$

where $x_s$ is the location of shock wave.

### *4.3.5 Approximation Results*

The results of the coordinate-straining approximation on the drag coefficient with the TSD solutions are tabulated in Table 1b, under the heading of $C_d - 4$. It is seen to significantly improve the wave drag approximation, with errors of less than 3% for changes in design variables of 2%. The effect of this approximation on the airfoil pressure distribution is shown in Fig. 7a. The baseline surface $C_p$ distribution for design variables $\bar{X}^0 = (0.5,\ 0.5,\ -0.5,\ 0.5)^T$ along with the approximate coordinate-straining result and the exact result for a 2% change in design variables

to $\bar{X} = (0.51,\ 0.51,\ -0.49,\ 0.51)^T$ are plotted. The results with the Euler solutions are presented in Table 2b. Here errors are slightly larger than the case of the TSD solutions, which is less than 7% for changes in design variables of 2%. The approximation of pressure distribution with the Euler solutions is plotted in Fig. 7b, which also has a good agreement with the exact pressure distribution. The coordinate straining approximations have been utilized in the design results which will be presented in the following two chapters.

*4.3.6 Discussion of Wave Drag from the TSD and Euler Codes.*

In the transonic flow regime, the inviscid drag consists of wave drag and induced drag. In the two-dimensional airfoil design problem based on the inviscid analysis, wave drag is the only component of drag. The current Euler analysis code, FLOMG evaluates the wave drag by integrating the pressure distribution on an airfoil. For the Euler equations, it is well known that this wave drag results from the entropy production by the shock waves[120]. For the full-potential flow which includes isentropic shock waves, wave drag is known to be the consequence of normal momentum jump across the shock waves[121]. It is possible to calculate wave drag in the TSD theory by estimating the first-order entropy production through a weak shock. This has been done by introducing a systematic asymtotic expansion procedure to the Euler equations in Ref. 122. The wave drag evaluation in the TSFOIL program is also based on this procedure. Thus, the actual wave drag values calculated from the Euler and TSD codes are much different, where the small disturbance solution usually predicts a considerably lower wave drag[102,122]. This difference can be considered as mainly due to the higher-order effect to the approximation of the entropy production across the shock waves. Table 1b and 2b also demonstrates the lower prediction of wave drag by the TSD solution while the lift calulations are somewhat similar. This affects the selection of wave drag constraint value in the present design problem, which was already mentioned in Chapter 2.

# 5. DESIGN BASED ON THE TSD EQUATIONS

In this chapter, several optimization strategies are considered for the transonic airfoil design problem of maximizing lift with constraints on wave drag and airfoil cross-sectional area as given by Eq. 2.1. The transonic small-disturbance analysis(TSFOIL) is involved in all of the designs. The design condition is at a free-stream Mach number of $M_\infty = 0.75$ and zero angle of attack, $\alpha = 0$. The minimum non-dimensional area is taken to be $A_{min} = 0.75$ and the wave drag constraint is taken to be $C_{dl} = 0.004$.

In all of the designs, a sequential approximation optimization algorithm is utilized. From a baseline analysis using the TSFOIL program and sensitivity computations, the lift coefficient with Eq. 4.1 and the wave-drag coefficient using Eqs. 4.8–4.15 are approximated. For the case of 4 design variables, this step requires 5 exact analyses from TSFOIL. The sensitivities are fixed until the process using the optimization code NEWSUMT-A converges. This step constitutes one complete design cycle. Then a new baseline analysis is performed to compute new sensitivities and the optimization process is repeated. This process continues until a global convergence is achieved. Note that NEWSUMT-A requires numerous approximate drag analyses, which involve coordinate straining. No additional exact analyses are required during the optimization. As a result, the bulk of the computational time is associated with the 5 calculations by TSFOIL for the baseline analysis and sensitivities per design cycle.

The various optimization strategies implemented are discussed next.

### *5.1 Strategy A: Approximate Optimization with Tight Move Limits*

The first optimization strategy employed consists of imposing tight move limits in the approximate optimization procedure. The move limits constrain the maximum amount that any of the design variables may change during a design cycle. These limits are set in order to keep the approximations of the objective function and constraints within a desired accuracy. The results of applying this strategy to the design problem starting from an initial design vector of $\bar{X}^0 = (0.5,\ 0.5,\ -0.5,\ 0.5)^T$ are tabulated in Table 3a. "Initial interpolations" in the title of Table 3a implies that the default cubic spline interpolation routine in TSFOIL was used to generate the continuous geometric data for airfoils designed during optimization process. It is specially noted here because this interpolation routine will be replaced with a new one later due to its poor performance around the leading edge. More details are discussed in Section 5.3. Here initially 5% move limits were imposed in order to keep the error in the drag approximation to within 10%. The solution ceased to improve after 27 iterations and the move limits were tightened to 2.5%. At 60 iterations the move limits were further reduced to 1.25% and the solution was considered to be converged. As can be seen from the table, the airfoil lift coefficient was increased from 0.530 to 0.559 while the wave-drag coefficient was reduced from .0058 to the prescribed .0040 and the airfoil area ratio remained at the prescribed value of 0.075. Although the optimization appeared to be successful, the convergence required too large a number of global iterations.

Next, the same strategy was employed, but from a different set of initial conditions. The results for $\bar{X}^0 = (0.8,\ -0.4,\ 0.7,\ -0.3)^T$ are tabulated in Table 3b. As can be seen, a completely different design was obtained. This design was only able to achieve a $C_l$ of 0.4055 while maintaining a $C_d$ of 0.0040 and an $A$ of 0.075.

In order to examine whether this second solution was simply a local maximum, the following was considered: We denote the first converged solution as $\bar{X}^1$ and the second as $\bar{X}^2$, and defined an intermediate design state

$$\bar{X} = \bar{X}^2 + \xi(\bar{X}^1 - \bar{X}^2)\,, \tag{5.1}$$

where the parameter $\xi$ may be considered the proportional distance of the intermediate design between $\bar{X}^2$ and $\bar{X}^1$. Both approximate and exact value were computed for the lift and wave-drag coefficients for intermediate designs with $0 \leq \xi \leq 1$. The results are plotted in Fig. 8. From the upper chart in Fig. 8, it can be seen that the lift coefficient is well-behaved between design state 2 and state 1, exhibiting no local maxima or minima. In the lower chart, it was found that the culprit is the drag-coefficient constraint, which exhibits a very *noisy* behavior about $C_d = 0.0040$. Thus we can see that if we are at design state 2, the optimizer would prevent us from moving toward state 1, since that would be a direction of increasing drag. An optimization strategy considered to alleviate this problem is discussed in the following section.

### *5.2 Strategy B: Drag Minimization followed by Lift Maximization*

A useful strategy to avoid the problems associated with the drag constraint was found and consisted of interchanging the roles of the drag contraint with the lift objective function. It was noticed that design problems of minimizing drag with a constraint on the lift were well behaved. In order to solve the design problem formulated in this study, the following strategy was adopted: first the wave-drag coefficient was minimized with a constraint on the lift coefficient of $C_l \geq 0.5$ and cross-sectional area ratio $A \geq 0.075$; then, when $C_d$ was below 0.004 it was reverted

back to the original design formulation of maximizing $C_l$ with constraints of $C_d \leq 0.004$ and $A \geq 0.075$.

The results of this strategy were very good. It was found that during the drag minimization phase, very large move limits, as large as 500% could be used without any adverse effects. During the lift maximization phase, move limits of 5% were imposed. This strategy seemed to be robust, and the solutions did not depend upon the initial data. The first case, starting from an initial design vector of $\bar{X}^0 = (0.5,\ 0.5,\ -0.5,\ 0.5)^T$ is tabulated in Table 4a. It took 4 design cycles to reduce the drag coefficient from 0.0059 to 0.0020 while keeping the lift coefficient fixed at 0.5 and the area ratio at 0.075. Then it is seen to take 3 more iterations, for a total of 7 iterations to increase the lift coefficient to 0.596 while keeping $C_d \leq 0.004$ and $A \geq 0.075$. The second case of $\bar{X}^0 = (0.8,\ -0.4,\ 0.7,\ -0.3)^T$ was tabulated in Table 4b. This case took more iterations than the first since the initial design was farther apart from the optimum, but it ended up with the nearly the same final design to the first case.

In order to illustrate that these designs are not anomalies, four other different initial design vectors were considered, far removed from the case discussed in the previous paragraph. Those are $\bar{X}^0 = (1.0,\ 0.0,\ 0.0,\ 0.0)^T$, $\bar{X}^0 = (0.0,\ 1.0,\ 0.0,\ 0.0)^T$, $\bar{X}^0 = (0.0,\ 0.0,\ 0.8,\ 0.0)^T$ and $\bar{X}^0 = (0.0,\ 0.0,\ 0.0,\ 1.0)^T$. (Note that an initial design vector of $(0.0,\ 0.0,\ 1.0,\ 0.0)^T$ corresponds to a thick airfoil which caused convergence problems with TSFOIL). The design results for these cases are tabulated in Tables 4c–4f. In all cases convergence to nearly the same design result was obtained in 9–11 global iterations. A plot of this very efficient convergence history with the strategy B is presented in Fig. 9 for the case of $\bar{X}^0 = (1.0,\ 0.0,\ 0.0,\ 0.0)^T$.

*5.3 Designs with Smooth Leading Edge Geometries.*

After successful design results using the strategy B were obtained, a careful investigation of the TSD solutions was made to determine the cause of the noisy drag calculation. It was found that the spline interpolation routine in the TSFOIL program generated an irregular airfoil leading-edge geometry. This poor nose geometry resulted because the interpolation was carried out separately for the upper surface and lower surface of the airfoil. Although this should not be important in the TSD solutions, which lose their validity at the leading edge, it clearly affected wave drag calculations and generated noise. The original routine was replaced with a more effective interpolation based on a periodic cubic spline interpolation of the arc-length of the airfoil and consequently fairly smooth nose shapes were generated.

Next, using this new geometry interpolation, new designs were attempted again with the strategy A which directly maximized the lift with tight move limits. The results for the first two initial conditions are tabulated in Tables 5a and 5b. For both initial conditions more than 40 design iterations were performed. However the global convergences were not achieved and design processes still proceeded. The tight move limits slowed down the convergence of the optimization process in these cases. Fig. 10 shows the objective function and constraint behavior for the intermediate designs between the 40th designs of both initial condition cases. The noise of wave drag constraint was reduced considerably but not removed completely. Thus, the design process may eventually encounter a local optimum. We did not try to reach full convergence for these cases because the design process already took more than 40 design cycles which are too many iterations for the purpose of the present study.

The designs with the strategy B were also attempted using the new interpolations. Tables 6a–6f represent the design results. Here, 200% move limits were used

for the first initial condition of $\bar{X}^0 = (0.5,\ 0.5,\ -0.5,\ 0.5)^T$ initially during the drag minimization phase, but the same move limits of 500 % were used for the other five cases. During the lift maximization phase, move limits of 20% were imposed instead of 5% used before. Again, this was just for getting faster convergence since the small move limits in the lift maximization simply slowed down the convergence of optimization process. In all cases, convergence to nearly the same design result was obtained in 5–13 design cycles. The convergence histories for the last four initial condition cases were plotted in Figs. 11a–11d. For the same cases, the pressure coefficients and the corresponding airfoil shapes of the initial and final designs were plotted in Figs. 12a–12d. The designs appear to be physically reasonable, with a weakened shock wave and lift increased through aft camber. Generally, the design results of this strategy were still very good with new geometry interpolations. The solutions did not depend upon the initial data and the strategy seemed to be robust and efficient. This will be examined in detail again for the designs with the Euler equation analysis in the next chapter.

### *5.4 Strategy C: Approximate Optimization with Absolute Move Limits*

On the basis of the success of the previous design strategy, the first design strategy of approximate optimization with tight move limits was reconsidered. This is the conventional strategy, where the move limits are imposed in order to keep the approximations with very high accuracy. This strategy, did not work when a constraint has *noise*, as observed the wave drag to be, resulting in solutions trapped in local optima. However, as can be seen from Tables 5a and 5b, this strategy works much better for the design problems with reduced noise in the wave drag constraint. The problem in these design cases is that slow convergence resulted simply from the tight move limits themselves. Therefore, it was suggested to increase move

limits to get faster convergence. This already has been demonstrated in the designs with the strategy B which is the drag minimization followed by lift maximization during the lift maximization phase, as shown in Tables 6a–6f. On the other hand, further experience in the designs using the direct lift maximization with large move limits indicated that if the percentage value of move limits were imposed on the present design problem, the design process was not very robust and sometimes easily produced slow convergence even with large percentage move limits. The problem is due to design variables which need to change their sign. Some examples showing this difficulty are presented in Ref. 85 where some of the design variables approached zero without changing their sign due to using percentage based move limits. Thus it was suggested to use move limits which were fixed absolute values rather than percentages of the design variables. This idea was tested and found to be effective to produce more reliable designs. Remaining work was to determine how large move limits could be imposed in order to achieve as fast convergence as possible. After several tests with different values of move limits, it was found that an absolute move limit of 0.5 could be used initially without any adverse effect. The move limits were then reduced to 0.25 and/or 0.125 when no further improvement in the objective function was obtained. The design results using this procedure for the last four initial condition cases are tabulated in Tables 7a–7d. The second case shown in Table 7b corresponding to initial condition $\bar{X}^0 = (0.0,\ 1.0,\ 0.0,\ 0.0)^T$ could not be started with absolute move limits. For this case a constraint relaxation strategy was adopted whereby the full $C_d \leq 0.004$ constraint was replaced by the constraint $C_d \leq 0.024$ for the first iteration. Some convergence difficulties with TSFOIL using this approach were experienced in the vicinity of the airfoil leading edge. When we proceeded design process by one more iteration following the last designs shown in Tables 7a–7d, the exact analysis for the new design could not be

performed due to the convergence problem of TSFOIL. Thus the design process with the strategy C ceased actually before the global convergence was achieved. However, based on these design iterations obtained by far, the solution was close to convergence. The convergence histories for the four design cases are plotted in Figs. 13a–13d. This strategy appears to be efficient but slightly less robust than the strategy B. This strategy will be examined in more detail with the Euler analysis method in the next chapter.

For the TSD designs, each exact airfoil analysis using the program TSFOIL required 10–15 CPU seconds on the IBM 3090 at V. P. I. & S. U., with 5 analyses needed per design cycle. The approximate optimization using the program NEWSUMT-A required 10–12 CPU seconds on the same computer. Thus for a typical case requiring say 8 design cycles, the total design process required approximately 600 CPU seconds, with 500 CPU seconds for the exact airfoil analyses and 100 CPU seconds in the optimization program.

# 6. DESIGN BASED ON THE EULER EQUATIONS

In the previous chapter, several design strategies have been considered for the problem of lift maximization with wave drag and area constraints based on the TSD analysis. Among those, two successful strategies, strategy B, drag minimization followed by lift maximization and strategy C, approximate optimization with absolute move limits, are applied to the same type of design problem with the Euler analysis method(FLOMG) in this chapter. The objective of this work is to validate the strategies developed with a simple transonic analysis method using more accurate anaysis method. Most of the conclusions on computational efficiency obtained from the design based on a simple numerical analysis method are expected to hold for the design with a more sophisticated analysis even though the actual design results will be depend on the specific form of analysis method employed.

The designs are performed at the same conditions of a free-stream Mach number of $M_\infty = 0.75$ and zero angle of attack, $\alpha = 0$. The minimum non-dimensional area is also taken to be $A_{min} = 0.75$. However, the wave drag constraint value is changed to 0.01 for the Euler design due to the differences in wave drag prediction between the TSD and Euler methods. The original value of 0.004 was found to be too stringent for the design problem with the Euler analysis. In the next section, we will discuss the selection of the wave drag constraint value. Global convergence will be closely examined for the Euler designs, with strategy B and strategy C in the subsequent sections.

*6.1 Selection of Wave Drag Constraint Value.*

At first, designs using the Euler analysis method were attempted for the exact same design problem as the TSD design where the wave drag constraint was $C_d \leq$ 0.004. The designs seemed to have much smaller values of $C_l$ compared to the previous TSD designs. This was found due to the fact that the wave drag constraint value of 0.004 was too stringent for the Euler designs. In Table 8, a sample of some of these results presented where the exact Euler analyses were performed to obtain the values of lift and drag coefficients for several intermediate design cases taken from the TSD designs in Table 6c. It can be seen that the TSD analysis predicts much lower wave drag than the Euler while the lift has similar values. This has been mentioned previously in Chapter 4 and it is due to the fact that the entropy jumps across the shock waves in TSD theory are neglected. In order to develop somewhat similar designs between the Euler and TSD methods, it was found necessary to utilize a larger value of the wave drag constraint for the Euler designs. The value of 0.01 selected for the Euler designs, roughly produces airfoil designs with the same optimum $C_l$ as was achieved for the TSD designs.

*6.2 Application of the Strategy B*

The strategy B was applied to the designs with the Euler analysis method. Table 9a represents the complete design history for the first case starting from $\bar{X}^0 = (1.0,\ 0.0,\ 0.0,\ 0.0)^T$. During the initial drag minimization phase, the lift constraint of $C_l \geq 0.5$ and 500% move limits were used, which were same as the corresponding TSD case. It took four design cycles to reduce the drag coefficient approximately by half from 0.0118 to 0.0057. In the lift maximization phase, 50% move limits were utilized initially, which were larger than the value of 20% used

in the TSD design. The move limits were selected on the basis of getting faster convergence. A large improvement in the lift coefficient to a value of $C_l = 0.7136$ was achieved with about 2% violation of wave drag constraint in the first four iterations. After that the move limits were reduced by half twice and slightly improved design of $C_l = 0.7084$ with 0.26% violation of wave drag, the 11th design, was obtained in four more iterations. Actually it is difficult to know whether this design is a real improvement without considering both the objective function improvement and constraint violations simultaneously. Usually when a design is in the initial stage, the sufficiently large improvement in the objective function can be achieved to still make improvement even with accounting for the loss due to the the violation of the constraints. The true improvement of the designs are relatively easy to determine in this case. However, as the design approaches to the optimum, the improvement in the objective function becomes smaller, so that even small violation of the constraints have a large effect against the improvement of the objective function. In this case, the true improvement of designs is hard to determine without estimating the true objective function improvement which already accounted for the loss due to constraint violations.

The equivalent lift coefficient, $C_{l_{equ.}}$ in Table 9a is based on this idea. It is a relevent estimated lift coefficient when all of the violated constraints are brought to be critical. Lagrange multipliers are used to estimate these equivalent lift coefficients, since they represent the amount of change in the objective function due to the unit change in a constraint at an optimum. Kuhn-Tucker conditions are used to compute the Lagrange multipliers. Details about computing the equivalent lift coefficient are described in Appendix A.

Finally very tight move limits of 2% were imposed and the design improvement was ceased in a total 25 iterations which resulted in $C_l = 0.7143$. For most design

purposes, the design achieved after 8 iterations corresponding to the 50% move limits considered to be acceptable as a final design. However, we felt that it was important to take this design to convergence in order to compare this strategy with other strategies. The complete convergence details are in Table 9a.

We consider the effect of varying initial conditions using this strategy and the results are presented in Tables 9b–9d. We did not attempt to take any of these cases to complete converge. The convergence history for the first 8 global iterations of the case in Table 9a is plotted in Fig. 14a. The other three initial-condition designs were also run with 50% move limits and the results were tabulated in Tables 9b–9d. The strategy of drag minimization followed by lift maximization was found to be as robust and efficient for the design based on the Euler method as it was for the TSD method. As shown in Tables 9a–9d, four different initial conditions yielded nearly the same final design in 8–12 design cycles. The convergence histories for the other three design cases are plotted in Figs. 14b–14d. The pressure distributions and corresponding airfoil shapes of the initial and final designs are shown in Figs. 15a–15d.

One important issue in the previous TSD designs with the strategy B was the noisy behavior of wave drag constraint. Here this issue was tested again with the Euler solutions. The initial and 8th designs from the Table 9a were taken to generate intermediate designs using Eq. 5.1. The exact Euler analyses were performed for these intermediate designs and the results were plotted in Fig. 16. Here, unlike the TSD solutions, we cannot find any noise in the wave drag calculations.

*6.3 Application of the Strategy C*

Next, it was attempted to directly maximize the lift with move limits which were fixed absolute values of the design variables, as utilized in the strategy C for the TSD designs. The complete design history for the first case, starting from $\bar{X}^0 = (1.0,\ 0.0,\ 0.0,\ 0.0)^T$ is tabulated in Table 10a. Initially a large improvement in the lift coefficient could be achieved to a value of $C_l = 0.7051$ using move limits of 0.5 for the first 10 iterations. After that move limits were reduced by half twice and a slightly improved value of $C_l = 0.7073$ was obtained in 6 more iterations. Finally very tight move limits of 0.02 were imposed which after 36 iterations resulted in $C_l = 0.7134$. In this strategy, more global iterations were required for the final refined design compared to 25 iterations on the strategy B. Here, also, the design achieved after 9 iterations, corresponding to the 0.5 move limits, was considered to be acceptable as a final design. The convergence history for these 9 global iterations is plotted in Fig. 17a. The other three initial-condition designs were run with 0.5 move limits and the results were tabulated in Tables 10b–10d. All cases did converge to approximately the same design result in 9–10 global iterations. The constraint relaxation, utilized to start design for the second initial-condition case in the TSD designs, was not necessary, here, in the Euler designs with this strategy. This strategy was as robust as the strategy B in the Euler designs even though it seemed to be less efficient for getting the final refined design. The convergence histories for those three design cases are plotted in Figs. 17b–17d. The pressure distributions and corresponding airfoil shapes of the initial and final designs are shown in Figs. 18a–18d. The wiggles in the pressure distribution behind the shock wave on the upper surface are due to surface wiggles in corresponding area, and maybe are unavoidable when an airfoil is defined by superposing several shape functions with

different slopes. These wiggles can be seen more clearly in Fig. 19 which has an enlarged $y$-scale for the airfoil geometry of the final design in Table 10a.

For the Euler designs, each exact airfoil analysis using the program FLOMG required approximately 40 CPU seconds on the CRAY 2S at NASA Langley, with 5 analyses needed per design cycle. The approximate optimization using the program NEWSUMT-A required 40–60 CPU seconds on the IBM 3090 at V. P. I. & S. U. The additional computer time associated with the approximate optimization of the Euler design is partly related to the more complicated wave drag calculation compared to that used for the TSD design. Thus for a typical case requiring say 8 design cycles, the total design process required approximately 1600 CPU seconds on the CRAY 2S for the exact airfoil analyses and 400 CPU seconds on the IBM 3090 in the optimization program.

# 7. CONCLUSIONS

Numerical optimization procedures have been considered for the design of transonic airfoils based on the transonic small-disturbance (TSD) and Euler equations. In order to achieve the goal of reliably producing design results utilizing few exact flow-field analyses, a sequential approximate optimization procedure has been implemented. A key ingredient in this procedure is to accurately approximate the wave drag.

A crucial ingredient of this approach was the utilization of Nixon's coordinate straining technique in obtaining accurate approximations to the wave drag. A modification of the surface boundary conditions was implemented in order to efficiently compute sensitivity derivatives without the need for remeshing with the Euler analysis.

The airfoil design problem considered here consisted of maximizing the lift with constraints on the wave drag and area. It was found that when the computed drag did not vary smoothly with the design parameters, the optimization process produced local extrema. A procedure interchanging the role of the objective function and constraint, initially minimizing drag with a constraint on the lift was found to be effective in producing converged designs. This procedure was also proven to be robust and efficient for cases where the drag varied smoothly, such as with the Euler solutions. The direct lift maximization with move limits which were fixed absolute values of the design variables, was also found to be a reliable and efficient procedure for designs based upon the Euler equations.

A recommendation for future work would involve the extention of these methods to the Navier-Stokes equations. Also the implementaion of these approaches within a complete three-dimensinal body design and eventually a multi-disciplinary design remains to be investigated.

## REFERENCES


1. Murman, E. M. and Cole, J. D., "Calculation of Plane Steady Transonic Flows," *AIAA Journal*, Vol. 9, No. 1, pp. 114–121, 1971.

2. Jameson, A., "Iterative Solution of Transonic Flow Over Airfoils and Wings, Including Flows at Mach 1," *Communications on Pure and Applied Mathematics*, Vol. 27, pp. 283–309, 1974.

3. Holst, T. L., Slooff, J. W., Yoshihara, H. and Ballhaus, W. F. Jr., "Computational Procedures in Transonic Aerodynamic Design," in *Applied Computational Transonic Aerodynamics*, AGARDograph No. 266, pp. 52–67, 1982.

4. Slooff, J. W., "Computational Methods For Subsonic and Transonic Aerodynamic Design," in *Special Course on Subsonic/Transonic Aerodynamic Interference for Aircraft*, AGARD Report No. 712, pp. 3.2–3.40, 1983.

5. Steger, J. L. and Klineberg, J. M., "A Finite Difference Method for Transonic Airfoil Design," *AIAA Journal*, Vol. 11, No. 5, pp. 628–635, May 1973.

6. Shankar, V., Malmuth, N. D. and Cole, J. D., "Computational Transonic Airfoil Design in Free Air and a Wind Tunnel," AIAA Paper No. 78–103, 1978.

7. Lock, R. C., "Research in the U.K. on Finite Difference Methods for Computing Steady Transonic Flow," in *Symposium Transonicum II*, Springer Verlag, 1976.

8. Shankar, V., Malmuth, N. D. and Cole, J. D., "Computational Transonic Design Procedure for Three-Dimensional Wings and Wing-Body Combinations," AIAA Paper No. 79–0344, 1979.

9. Shankar, V., "Computational Transonic Inverse Procedure for Wing Design with Automatic Trailing Edge Closure," AIAA Paper No. 80-1390, 1980.

10. Lighthill, M. J., "A New Method of Two-Dimensional Aerodynamic Design," ARC R&M 2112, 1945.

11. Tranen, T. L., "A Rapid Computer Aided Transonic Airfoil Design Method," AIAA Paper No. 74–501, 1974.

12. Bauer, F., Garabedian, P. and Korn, D., *Supercritical Wing Sections I*, Springer Verlag, 1972.

13. Volpe, G., "Two-Element Airfoil Systems Design: An Inverse Method," AIAA Paper No. 78–1226, 1978.

14. Arlinger, B. and Schmidt, W., "Design and Analysis of Slat Systems in Transonic Flow," ICAS Paper, 1978.

15. Carlson, L. A., "Transonic Airfoil Analysis and Design using Cartesian Coordinates," NASA CR-2578, April 1976.

16. Carlson, L. A., "TRANDES: A FORTRAN Program for Transonic Airfoil Analysis or Design," NASA CR-2821, June 1977.

17. Volpe, G. and Melnik, R. E., "The Role of Constraints in the Inverse Design Problem for Transonic Airfoils," AIAA Paper No. 81–1233, 1981.

18. Volpe, G. and Melnik, R. M., "The Design of Transonic Airfoils by a Well-posed Inverse Method," *Int'l. Journal for Numerical Methods in Engineering*, Vol. 22, pp. 341–361, 1986.

19. Henne, P. A., "An Inverse Transonic Wing Design Method," AIAA Paper No. 80–0330, 1980.

20. Jameson, A. and Caughey, D., "Numerical Calculation of the Transonic Flow Past a Swept Wing," ERDA R&D Rept. C00–3077–140, Courant Institute, New York University, 1977.

21. Shankar, V., "A Full Potential Inverse Method Based on a Density Linearization Scheme for Wing Design," AIAA Paper No. 81–1234, 1981.

22. Caughey, D. and Jameson, A., "Progress in Finite Volume Calculation for Wing-Body Combinations," *AIAA Journal*, Vol. 18, No. 11, pp. 1281–1288, 1980.

23. Weed, R. A., Anderson, W. K. and Carlson, L. A., "A Direct-Inverse Three-Dimensional Transonic Wing Design Method for Vector Computers," AIAA Paper No. 84–2156, Aug. 1984.

24. Gally, T. A. and Carlson, L. A., "Inviscid Transonic Wing Design Using Inverse Methods in Curvilinear Coordinates," AIAA Paper No. 87–2551, 1987.

25. Ratcliff, R. R. and Carlson, L. A., "A Direct-Inverse Transonic Wing-Design Method in Curvilinear Coordinates Including Viscous-Interaction," AIAA Paper No. 89–2204, 1989.

26. Davis, W. H. Jr., "Technique for Developing Design Tools from the Analysis Methods of Computational Aerodynamics," AIAA Paper No. 79–1529, 1979.

27. Fray, J. M. J., Slooff, J. W., Boerstoel, J. W. and Kassies, A., "Design of Transonic Airfoils with Given Pressure subject to Geometric Constraints," NLR TR 84064 U, July 1984.

28. Fray, J. M. J. and Slooff, J. W., "A Constrained Inverse Method for the Aerodynamic Design of Thick Wings with Given Pressure Distribution in Subsonic Flow," AGARD CP–285, Paper 16, 1980.

29. Boerstoel, J. W., "Numerical Modelling and Fast-Solver Calculation of Approximately Normal Shocks," NLR MP 82026 U, 1982.

30. McFadden, G., "An Artificial Viscosity Method for the Design of Supercritical Airfoils," DOE R&D Rept. C00–3077–158, Courant Institute of Mathematical Sciences, New York University, 1979.

31. Bauer, F., Garabedian, P. and Korn, D., *Supercritical Wing Sections II*, Springer Verlag, 1975.

32. Garabedian, P. and McFadden, G., "Design of Supercritical Swept Wings," proceedings of the 1980 *Army Numerical Analysis and Computers Conference*, ARO Rept. 80–3, 1980.

33. Garabedian, P. and McFadden, G., "Computational Fluid Dynamics of Airfoils and Wings," proceedings of Symposium on *Transonic, Shock and Multi-Dimensional Flows*, Univ. of Wisconsin, Madison, 1981, also Academic Press Inc., pp. 1–16, 1982.

34. Garabedian, P. and McFadden, G., "Design of Supercritical Swept Wings," AIAA Journal, Vol. 20, No. 3, pp. 289–291, Mar. 1982.

35. Malone, J. B., Vadyak, J. and Sankar, L. N., "A Technique for the Inverse Aerodynamic Design of Nacelles and Wing Configurations," AIAA Paper No. 85–4096, Oct. 1985.

36. Malone, J. B., Vadyak, J. and Sankar, L. N., "Inverse Aerodynamic Design Method for Aircraft Components," *Journal of Aircraft*, Vol. 24, No. 1, pp. 8–9, 1987.

37. Giles, M., Drela, M. and Tompkins, W. T. Jr., "Newton Solution of Direct and Inverse Transonic Euler Equations," AIAA Paper No. 85-1530, Jan. 1985.

38. Giles, M., Drela, M. and Tompkins, W. T. Jr., "Two-Dimensional Transonic Aerodynamic Design Methods," *AIAA Journal*, Vol. 25, No. 9, pp. 1199–1206, 1987.

39. Drela, M. and Giles, M., "A Two-Dimensional Viscous Aerodynamic Design and Analysis Code," AIAA Paper No. 87–0424, 1987.

40. Meauze, G., "On the Use of Inverse Modes of Calculation in Two-Dimensional Cascades and Ducts," *ICIDES-I*, Oct. 1984, also ONERA TP No. 1984–132.

41. Morice, Ph., Cambier, L. and Veuillot, J. P., "Numerical Computation of Transonic Flow Past an Axisymmetric Nacelle," 14th ICAS Congress, Toulouse, Sep. 1984, also ONERA TP No. 1984–91.

42. Couaillier, V. and Veuillot, J. P., "Inverse Method for the Euler Equations," proceedings of Second *International Conference on Inverse Design Concepts and Optimization in Engineering Sciences, ICIDES-II*, pp. 131–146, Oct. 1987.

43. Malone, J. B., Narramore, J. C. and Sankar, L. N., "An Efficient Airfoil Design Method Using the Navier-Stokes Equations," in *Computational Methods for Aerodynamic Design(Inverse) and Optimization*, AGARD CP–463, pp. 5.1–5.18, 1990.

44. Birckelbaw, L., "Inverse Airfoil Design Using the Navier-Stokes Equations," AIAA Paper No. 89–2202, 1989.

45. Nieuwland, G. Y., "The Computation by Lighthill's Method of Transonic Potential Flow around a Family of Quasi-Elliptical Aerofoils," NLR TR T.83, 1964.

46. Boerstoel, J. W., "Review of the Application of Hodograph Theory to Transonic Aerofoil Design and Theoretical and Experimental Analysis of Shock-Free Aerofoils," proceedings of *IUTAM Symposium Transonicum II*, edited by K. Oswatitsch and D. Rues, pp. 109–133, Springer-Verlag, 1976, also NLR MP 75033 U, 1975.

47. Nieuwland, G. Y., "Transonic Potential Flow around a Family of Quasi-Elliptical Aerofoil Sections," NLR TR T.172, 1967.

48. Takanaski, S., "A Method of Obtaining Transonic Shock-Free Flow Around Lifting Aerofoils," *Transactions. Japan Society for Aeronautical and Space Science*, Vol. 16, No. 34, pp. 246–263, 1973.

49. Boerstoel, J. W. and Huizing, G. H., "Transonic Shock-Free Airfoil Design by an Analytic Hodograph Method," AIAA Paper No. 74–539, 1974, also NLR MP 74025 U, 1974.

50. Sobieczky, H., "Entwurf überkritischer Profile mit Hilfe der Rheo-Elektrische Analogie," DLR–FB 75–43, 1975.

51. Eberle, A., "Eine exakte Hodographenmethode zum Entwurf überkritische Profile," MBB UFE 1168–75 ö, 1975, also *IUTAM Symp. Trans. II.*

52. Bauer, F., Garabedian, P. and Korn, D., *Supercritical Wing Sections III*, Springer Verlag, 1977.

53. Oyibo, G. A., "Exact Closed-Form Solutions for Nonlinear Unsteady Transonic Aerodynamics," *AIAA Journal*, Vol. 27, No. 11, pp. 1572–1578, 1989.

54. Oyibo, G. A., "Formulation of Three-Dimensional Hodograph Method and Separable Solutions for Nonlinear Transonic Flows," *AIAA Journal*, Vol. 28, No. 10, pp.1745–1750, 1990.

55. Sobieczky, H., "Related Analytical, Analog and Numerical Methods in Transonic Airfoil Design," AIAA Paper No. 79–1556, 1979.

56. Sobieczky, H., Fung, K. Y. and Seebass, A. R., "A New Method for Designing Shock-Free Transonic Configurations," AIAA Paper No. 78–1114, 1978.

57. Yu, N. J., "Efficient Transonic Shock-Free Wing Redesign Procedure Using a Fictitious Gas Method," AIAA Paper No. 79–0075, 1979.

58. Eberle, A., "Transonic Flow Computations by Finite Elements: Airfoil Optimization and Analysis," in *Recent Developments in Theoretical and Experimental Fluid Mechnics*, Springer Verlag, 1979.

59. Fung, K. Y., Sobieczky, H. and Seebass, A. R., "Shock-Free Wing Design," AIAA Paper No. 79–1557, 1979.

60. Rai, P., Miranda, L. R. and Seebass, A. R., "A Cost-Effective Method for Shock-Free Supercritical Wing Design," AIAA Paper No. 81–0383, 1981.

61. Fung, K. Y., Seebass, A. R., Dickson, L. J. and Pearson, C. F., "An Effective Algorithm for Shock-Free Wing Design," AIAA Paper No. 81–1236, 1981.

62. Nebeck, H. E., Seebass, A. R. and Sobieczky, H., "Inviscid-Viscous Interaction in the Nearly Direct Design of Shock-Free Supercritical Airfoils," AGARD CP–291, 1980.

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

64. Vanderplaats, G. N. and Hicks, R. M. and Murman, E. M., "Application of Numerical Optimization Techniques to Airfoil Design," NASA SP–347, part II, 1975.

65. Hicks, R. M., Vanderplaats, G. N. and Murman, E. M. and King, R. T., "Airfoil Section Drag Reduction at Transonic Speeds by Numerical Optimization," SAE Paper No. 760477, 1976, also NASA TM X–73097, 1976.

66. Hicks, R. M. and Vanderplaats, G. N., "Application of Numerical Optimization to the Design of Supercritical Airfoils without Drag-Creep," SAE Paper No. 770440, Business Aircraft Meeting, 1977.

67. Vanderplaats, G. N. and Hicks, R. M., "Numerical Airfoil Optimization Using a Reduced Number of Design Coordinates," NASA TM X–73151, July 1976.

68. Vanderplaats, G. N., "Approximation Concepts for Numerical Airfoil Optimization," NASA TP 1370, 1979.

69. Hicks, R. M. and Henne, P. A., "Wing Design by Numerical Optimization," AIAA Paper No. 77–1247, Aug. 1977.

70. Lores, M. E., Smith, P. R. and Hicks, R. M., "Supercritical Wing Design Using Numerical Optimization and Comparisons with Experiment," AIAA Paper No. 79–0065, Jan. 1979.

71. Aidala, P. V., Davis, W. H. Jr. and Mason, W. H., "Smart Aerodynamic Optimization," AIAA Paper 83–1863, July 1983.

72. Davis, W. H. Jr., "TRO–2D: A code for Rational Transonic Aero Optimization," AIAA Paper No. 85–0425, Jan. 1985.

73. Davis, W. H. Jr., Aidala, P. V. and Mason, W. H., "A Study to Develop Improved Methods for the Design of Transonic Fighter Wings by the Use of Numerical Optimization," NASA CR–3995, 1986.

74. Haney, H. P., Johnson, R. R. and Hicks, R. M., "Computatinal Optimization and Wind Tunnel Test of Transonic Wing Designs," AIAA Paper No. 79–0080, 1979.

75. Hicks, R. M., "Transonic Wing Design Using Potential-Flow Codes – Success & Failure," SAE Tech. Paper No. 810565, April 1981.

76. Lores, M. E., Smith, P. E. and Large, R. A., "Numerical Optimization: An Assessment of its Role in Transport Aircraft Aerodynamic Design through a Case Study," ICAS–80–1.2, 1980.

77. Lores, M. E. and Hinson, B. L., "Transonic Design Using Computatinal Aerodynamics," in *Transonic Aerodynamics*, Progress in Astronautics and Aeronautics, edited by D. Nixon, Vol. 81, pp. 377–402, 1981.

78. Lores, M. E., Burdges, K. P. and Shrewsbury, G. D., "Analysis of a Theoretically Optimized Transonic Airfoil," NASA CR–3065, Nov. 1978.

79. Haney, H. P. and Johnson, R. R., "Application of Numerical Optimization to the Design of Wings with Specified Pressure Distributions," NASA CR–3238, Feb. 1980.

80. Cosentino, G. B. and Holst, T. L., "Numerical Optimization Design of Advanced Transonic Wing Configurations," AIAA Paper No. 85–0424, Jan. 1985, also NASA TM–85950, 1984.

81. Holst, T. and Thomas, S., "Numerical Solution of Transonic Wing Flow Fields," AIAA Paper No. 82–0105, Jan. 1982.

82. Vanderplaats, G. N., "CONMIN – A Fortran Program for Constrained Function Minimization," NASA TM X–62282, 1973.

83. Gill, P. E. and Murray, W., "Quasi-Newton Methods for Unconstrained Optimization," *Journal Int'l. Mathematics Applications*, Vol. 9, pp. 91–108, 1972.

84. Kennelly, R. A. Jr., "Improved Method for Transonic Airfoil Design-by-Optimization," AIAA Paper No. 83–1864, July 1983.

85. Joh, C.-Y., Grossman, B. and Haftka, R. T., "Efficient Optimization Procedures for Transonic Airfoil Design," proceedings of ASME Winter Annual Meeting, AD-Vol. 16, *Computational Structural Mechanics and Multidisciplinary*

*Optimization*, edited by Grandhi, Stroud and Venkayya, Book No. H00534, pp. 67–76, Dec. 1989.

86. Miura, H. and Schmit, L. A., Jr., "NEWSUMT – A Fortran Program for Inequality Constrained Function Minimization – Users Guide," NASA CR–159070, June 1979.

87. Grandhi, R. V., Thareja, R. and Haftka, R. T., "NEWSUMT-A: A General Purpose Program for Constrained Optimization using Constraint Approximations," *ASME Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 107, pp. 94–99, 1985.

88. Gregg, R. D. and Misegades, K. P., "Transonic Wing Optimization Using Evolution Theory," AIAA Paper No. 87–0520, Jan. 1987.

89. Haftka, R. T., Grossman, B., Eppard, W. M., Kao, P.-J., and Polen, D., "Efficient Optimization of Integrated Aerodynamic-Structural Design," *International Journal for Numerical Methods in Engineering*, Vol. 28, pp. 593–607, 1989.

90. Grossman, B., Haftka, R. T. , Kao, P.-J., Polen, D. M., Rais-Rohani, M. and Sobieszczanski-Sobieski, J., "Integrated Aerodynamic-Structural Design of a Transport Wing," AIAA Paper No. 89–2129, July 1989, also *Journal of Aircraft*, Vol. 27, No. 12, pp. 1050–1056, Dec. 1990.

91. Mason, W. H., Ballhaus, W. F., MacKenzie, D., Frick, J. and Stern, M., "An Automated Procedure for Computing the Three-Dimensional Transonic Flow over Wing-Body Combinations, Including Viscous Effects; Description of Analysis Methods and Application," AFFDL-TR-77-122, Vol. 1, pp.177–197, Feb. 1977.

92. Rubbert, P. and Goldhammer, M., "CFD in Design: An Airframe Perspective," AIAA Paper No. 89–0092, Jan. 1989.

93. Ashley, H., "On Making Things the Best - Aeronautical Uses of Optimization," *Journal of Aircraft*, Vol. 19, pp. 5–28, 1982.

94. Waggoner, C. G., Haney, H. P. and Ballhaus, W. F. Jr., "Computational Wing Optimization and Comparisons with Experiment for a Semi–Span Wing Model," NASA TM–78480, 1978.

95. Aidala, P., "Numerical Aircraft Design Using 3-D Transonic Analysis with Optimization," *Fighter Design*, AFWAL TR–81–3091, Vol. II, Part 2, Aug. 1981.

96. Levinsky, E. S. and Palco, R. L., "Supercritical Tests of a Self-Optimizing Variables Camber Wind-Tunnel Model," in *Advanced Technology Airfoil Research*, NASA CP–2045, Vol. I, Part 1, pp.297–313, 1979.

97. Sobieszczanski-Sobieski, J., "Multidisciplinary Optimization for Engineering systems: Achievements and Potential," NASA TM 101566, presented at the DFVLR Seminar on Optimization, Bonn, West Germany, June 7–8, 1989.

98. Khot, N. S., Grandhi, R. V. and Venkayya, V. B., "Structural and Control Optimization of Space Structures," AIAA Paper No. 87–0939, 1987.

99. Schmit, L. A. and Farshi, B., "Some Approximation Concepts for Structural Synthesis," *AIAA Journal*, Vol. 12, No. 7, 1974, pp. 692–699.

100. Spreiter, J. R., "Transonic Aerodynamics–History and Statement of the Problem," Chapter 1 in *Transonic Aerodynamics*, Vol. 81, Progress in Astronautics and Aeronautics, edited by D. Nixon, AIAA, New York, pp. 3–66, 1982.

101. Murman, E. M. and Cole, J. D., "Inviscid Drag at Transonic Speeds," AIAA Paper No. 74–540, June 1974.

102. Murman, E. M., Bailey, F. R., and Johnson, M. H., "TSFOIL - A Computer Code for 2-D Transonic Calculations, Including Wind-Tunnel Wall Effects and Wave-Drag Evaluation", NASA SP-347, Part II, pp. 769–788, 1975.

103. Stahara, S. S., "Operational Manual for Two-Dimensional Transonic Code TSFOIL", NASA CR–3064, Dec. 1978.

104. Ballhaus, W. F. and Steger, J. L., "Implicit Approximate Factorization Schemes for the Low Frequency Transonic Equation," NASA TM X–73082, 1975.

105. Ballhaus, W. F., Jameson, A. and Albert, J., "Implicit Approximate Factorization Schemes for Steady Transonic Flow Problems," *AIAA Journal*, Vol. 16, No. 6, pp. 573–579, June 1978.

106. Fedorenko, R. P., "The Speed of Convergence of One Iterative Process," *USSR. Comp. Math. and Math. Phys.*, Vol. 4, pp. 227–235, 1964.

107. South, J. C. and Brandt, A., "Application of a Multi-Level Grid Method to Transonic Flow Calculations," *Transonic Flow Problems in Turbomachinery*, edited by T. C. Adamson and M. F. Platzer, Hemisphere, Washington, D. C., pp. 180–207, 1977.

108. Thomas, J. L. and Salas, M. D., "Far-Field Boundary conditions for Transonic Lifting Solutions to the Euler Equations," *AIAA Journal*, Vol. 24, No. 7, pp. 1074–1080, July 1986.

109. Rizzi, A., "Numerical Implimentation of Solid Body Boundary Conditions for the Euler Equations," *ZAMM*, Vol. 26, pp. 181–196, 1978.

110. Swanson, R. C. and Turkel, E., "A Multistage Time-Stepping Scheme for the Navier-Stokes Equations," AIAA Paper No. 85-0035, Jan. 1985.

111. Swanson, R. C. and Turkel, E., "Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations," AIAA Paper No. 87–1107, June 1987.

112. Jameson, A., Schmidt, W. and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes," AIAA Paper No. 81-1259, June 1981.

113. Jameson, A., "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method," *Applied Mathematics and Computation*, Vol. 13, pp. 327–355, 1983.

114. Turkel, E., "Acceleration to a Steady State for the Euler Equations," NASA CR–172398, also ICASE Report No. 84–32, July 1984.

115. Nixon, D., "Perturbation of a Discontinuous Transonic Flow," *AIAA Journal*, Vol. 16, No. 1, pp. 47–52, 1978.

116. Nixon, D., "Perturbations in Two- and Three-Dimensional Transonic Flows," *AIAA Journal*, Vol. 16, No. 7, pp. 699–709, 1978.

117. Stahara, S. S., Elliott, J. P. and Spreiter, J. R., "A Rapid Approximation Procedure for Nonlinear Solutions: Application to Aerodynamic Flows," *AIAA Journal*, Vol. 20, No. 8, pp. 1104–1112, 1982.

118. Haftka, R. T., Gürdal, Z. and Kamat, M. P., *Elements of Structural Optimization*, 2nd Ed., Kluwer Academic Publishers, 1990.

119. Lighthill, M. J., "A Technique for Rendering Approximate Solutions to Physical Problems Uniformly Valid," *Philosopy Magazine*, Vol. 40, pp. 1197–1201, 1949.

120. Oswatitsch, K., *Gas Dynamics*, pp. 203–210, Academic Press, Inc., 1956.

121. Steger, J. L. and Baldwin, B. S., "Shock Waves and Drag in the Numerical Calculation of Isentropic Transonic Flow," NASA TN D–6997, 1972.

122. Murman, E. M. and Cole, J. D., "Inviscid Drag at Transonic Speeds," AIAA Paper No. 74–540, June 1974.

# TABLES

**Table 1a.** Lift Linear Approximation – TSD.

| change | Design Parameters | | | | $C_l$ | |
|---|---|---|---|---|---|---|
| % | $X_1$ | $X_2$ | $X_3$ | $X_4$ | Linear | Exact |
| 0.0 | 0.5 | 0.5 | -0.5 | 0.5 | | 0.5341 |
| 0.5 | 0.5025 | 0.5025 | -0.4975 | 0.5025 | 0.5472 | 0.5471 |
| -0.5 | 0.4975 | 0.4975 | -0.5025 | 0.4975 | 0.5210 | 0.5216 |
| 1.0 | 0.5050 | 0.5050 | -0.4950 | 0.5050 | 0.5602 | 0.5614 |
| -1.0 | 0.4950 | 0.4950 | -0.5050 | 0.4950 | 0.5080 | 0.5098 |
| 1.5 | 0.5075 | 0.5075 | -0.4925 | 0.5075 | 0.5733 | 0.5762 |
| -1.5 | 0.4925 | 0.4925 | -0.5075 | 0.4925 | 0.4949 | 0.4987 |
| 2.0 | 0.5100 | 0.5100 | -0.4900 | 0.5100 | 0.5864 | 0.5918 |
| -2.0 | 0.4900 | 0.4900 | -0.5100 | 0.4900 | 0.4818 | 0.4881 |

**Table 1b.** Drag Approximations – TSD.

| change* | Drag Coefficient | | | | |
|---|---|---|---|---|---|
| % | $C_d$-1 | $C_d$-2 | $C_d$-3 | $C_d$-4 | Exact |
| 0.0 | | | | | 0.00511 |
| 0.5 | 0.00584 | 0.00585 | 0.00584 | 0.00579 | 0.00585 |
| -0.5 | 0.00439 | 0.00336 | 0.00343 | 0.00447 | 0.00436 |
| 1.0 | 0.00656 | 0.00555 | 0.00563 | 0.00654 | 0.00657 |
| -1.0 | 0.00367 | 0.00059 | 0.00081 | 0.00396 | 0.00400 |
| 1.5 | 0.00728 | 0.00422 | 0.00446 | 0.00736 | 0.00755 |
| -1.5 | 0.00295 | -.00323 | -.00277 | 0.00346 | 0.00342 |
| 2.0 | 0.00800 | 0.00186 | 0.00234 | 0.00827 | 0.00845 |
| -2.0 | 0.00223 | -.00808 | -.00730 | 0.00302 | 0.00295 |

* Design variables the same as in Table 1a.

*$C_d$-1: linear approximation; $C_d$-2: quadratic polynomial in $C_l$; $C_d$-3: quadratic polynomial in $C_l$ and linear in shock position; $C_d$-4: strained coordinate approximation*

**Table 2a.** Lift Linear Approximation – Euler.

| change | Design Parameters | | | | $C_l$ | |
|---|---|---|---|---|---|---|
| % | $X_1$ | $X_2$ | $X_3$ | $X_4$ | Linear | Exact |
| 0.0 | 0.5 | 0.5 | -0.5 | 0.5 | | 0.4878 |
| 0.5 | 0.5025 | 0.5025 | -0.4975 | 0.5025 | 0.4942 | 0.4948 |
| -0.5 | 0.4975 | 0.4975 | -0.5025 | 0.4975 | 0.4813 | 0.4809 |
| 1.0 | 0.5050 | 0.5050 | -0.4950 | 0.5050 | 0.5006 | 0.5016 |
| -1.0 | 0.4950 | 0.4950 | -0.5050 | 0.4950 | 0.4749 | 0.4737 |
| 1.5 | 0.5075 | 0.5075 | -0.4925 | 0.5075 | 0.5070 | 0.5086 |
| -1.5 | 0.4925 | 0.4925 | -0.5075 | 0.4925 | 0.4685 | 0.4667 |
| 2.0 | 0.5100 | 0.5100 | -0.4900 | 0.5100 | 0.5134 | 0.5153 |
| -2.0 | 0.4900 | 0.4900 | -0.5100 | 0.4900 | 0.4621 | 0.4598 |

**Table 2b.** Drag Approximations – Euler.

| change* | Drag Coefficient | | | | |
|---|---|---|---|---|---|
| % | $C_d$-1 | $C_d$-2 | $C_d$-3 | $C_d$-4 | Exact |
| 0.0 | | | | | 0.00787 |
| 0.5 | 0.00866 | 0.00869 | 0.00868 | 0.00870 | 0.00854 |
| -0.5 | 0.00708 | 0.00567 | 0.00618 | 0.00718 | 0.00724 |
| 1.0 | 0.00946 | 0.00815 | 0.00861 | 0.00957 | 0.00923 |
| -1.0 | 0.00628 | 0.00213 | 0.00360 | 0.00652 | 0.00664 |
| 1.5 | 0.01029 | 0.00625 | 0.00766 | 0.01049 | 0.00996 |
| -1.5 | 0.00549 | -.00277 | 0.00014 | 0.00590 | 0.00607 |
| 2.0 | 0.01104 | 0.00300 | 0.00583 | 0.01149 | 0.01072 |
| -2.0 | 0.00470 | -.00901 | -.00420 | 0.00533 | 0.00553 |

* Design variables the same as in Table 2a.

*$C_d$-1: linear approximation; $C_d$-2: quadratic polynomial in $C_l$; $C_d$-3: quadratic polynomial in $C_l$ and linear in shock position; $C_d$-4: strained coordinate approximation*

**Table 3a.** TSD Design Strategy A : Approximate Optimization with Tight Move Limits – initial condition 1, initial interpolations.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.5 | 0.5 | -0.5 | 0.5 | 0.5296 | 0.0059 | 0.0805 |
| 1 | 0.510 | 0.498 | -0.524 | 0.490 | 0.4954 | 0.0040 | 0.0779 |
| – | — | — | — | — | — | — | — |
| 27 | 0.400 | 0.750 | -0.649 | 0.481 | 0.5385 | 0.0040 | 0.0750 |
| – | — | — | — | — | — | — | — |
| 60 | 0.301 | 0.891 | -0.763 | 0.565 | 0.5542 | 0.0040 | 0.0750 |
| – | — | — | — | — | — | — | — |
| 71 | 0.280 | 0.934 | -0.805 | 0.591 | 0.5592 | 0.0040 | 0.0750 |

**Table 3b.** TSD Design Strategy A : Approximate Optimization with Tight Move Limits – initial condition 2, initial interpolation.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.8 | -0.4 | 0.7 | -0.3 | 0.4418 | 0.0061 | 0.0739 |
| 1 | 0.837 | -0.426 | 0.659 | -0.281 | 0.4081 | 0.0038 | 0.0729 |
| – | — | — | — | — | — | — | — |
| 9 | 1.093 | -0.483 | 0.504 | -0.273 | 0.4061 | 0.0041 | 0.0750 |
| – | — | — | — | — | — | — | — |
| 11 | 1.114 | -0.487 | 0.491 | -0.275 | 0.4049 | 0.0040 | 0.0750 |
| – | — | — | — | — | — | — | — |
| 15 | 1.122 | -0.487 | 0.487 | -0.277 | 0.4055 | 0.0040 | 0.0750 |

**Table 4a.** TSD Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 1, initial interpolations.

| Design cycle | Design Parameters | | | | $C_l$ | $C_d$ | A |
|---|---|---|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | | | |
| 0 | 0.5 | 0.5 | -0.5 | 0.5 | 0.5296 | 0.0059 | 0.0805 |
| 1 | -0.021 | 1.041 | -0.705 | 0.658 | 0.5793 | 0.0046 | 0.0750 |
| 2 | 0.052 | 0.965 | -0.862 | 0.811 | 0.5027 | 0.0025 | 0.0750 |
| 3 | 0.073 | 1.048 | -1.030 | 0.895 | 0.5011 | 0.0021 | 0.0750 |
| 4 | 0.056 | 1.087 | -1.088 | 0.934 | 0.5009 | 0.0020 | 0.0750 |
| *5 | 0.058 | 1.142 | -1.058 | 0.888 | 0.5601 | 0.0039 | 0.0775 |
| 6 | 0.058 | 1.199 | -1.067 | 0.845 | 0.5815 | 0.0042 | 0.0768 |
| 7 | 0.059 | 1.259 | -1.086 | 0.803 | 0.5956 | 0.0041 | 0.0754 |

* begin lift maximization.

**Table 4b.** TSD Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 2, initial interpolations.

| Design cycle | Design Parameters | | | | $C_l$ | $C_d$ | A |
|---|---|---|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | | | |
| 0 | 0.8 | -0.4 | 0.7 | -0.3 | 0.4418 | 0.0061 | 0.0739 |
| 1 | 1.433 | 0.260 | -0.495 | -0.144 | 0.6033 | 0.0108 | 0.0750 |
| 2 | 0.099 | 0.431 | 0.003 | 0.332 | 0.5164 | 0.0066 | 0.0750 |
| 3 | 0.248 | 0.348 | -0.013 | 0.290 | 0.4945 | 0.0052 | 0.0750 |
| 4 | 0.370 | 0.355 | -0.081 | 0.251 | 0.5027 | 0.0049 | 0.0750 |
| 5 | 0.560 | 0.539 | -0.484 | 0.348 | 0.5161 | 0.0040 | 0.0750 |
| 6 | 0.454 | 0.742 | -0.725 | 0.517 | 0.5236 | 0.0036 | 0.0750 |
| 7 | 0.335 | 0.864 | -0.948 | 0.740 | 0.4926 | 0.0024 | 0.0750 |
| 8 | 0.167 | 0.922 | -0.877 | 0.763 | 0.5029 | 0.0025 | 0.0750 |
| 9 | 0.084 | 1.019 | -1.002 | 0.881 | 0.4966 | 0.0021 | 0.0750 |
| *10 | 0.086 | 1.069 | -0.973 | 0.837 | 0.5553 | 0.0040 | 0.0776 |
| 11 | 0.087 | 1.123 | -0.989 | 0.795 | 0.5654 | 0.0038 | 0.0761 |
| 12 | 0.091 | 1.179 | -1.010 | 0.759 | 0.5803 | 0.0039 | 0.0750 |
| 13 | 0.090 | 1.222 | -1.045 | 0.761 | 0.5924 | 0.0041 | 0.0750 |

* begin lift maximization.

**Table 4c.** TSD Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 3, initial interpolations.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.5874 | 0.0127 | 0.0822 |
| 1 | 1.003 | 0.100 | -0.068 | -0.086 | 0.5245 | 0.0063 | 0.0750 |
| 2 | 0.879 | 0.196 | -0.194 | 0.065 | 0.5022 | 0.0050 | 0.0750 |
| 3 | 0.643 | 0.429 | -0.410 | 0.293 | 0.4997 | 0.0038 | 0.0750 |
| 4 | 0.389 | 0.663 | -0.610 | 0.519 | 0.5000 | 0.0030 | 0.0750 |
| 5 | 0.295 | 0.923 | -0.969 | 0.748 | 0.5067 | 0.0026 | 0.0750 |
| 6 | 0.063 | 1.047 | -1.016 | 0.890 | 0.5022 | 0.0022 | 0.0750 |
| *7 | 0.064 | 1.099 | -0.999 | 0.863 | 0.5629 | 0.0040 | 0.0781 |
| 8 | 0.065 | 1.154 | -1.017 | 0.828 | 0.5782 | 0.0040 | 0.0771 |
| 9 | 0.066 | 1.212 | -1.043 | 0.797 | 0.5920 | 0.0040 | 0.0761 |
| 10 | 0.067 | 1.272 | -1.078 | 0.776 | 0.6062 | 0.0040 | 0.0752 |

* begin lift maximization.

**Table 4d.** TSD Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 4, initial interpolations.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0554 | 0.0468 | 0.0772 |
| 1 | 0.052 | 0.914 | -0.100 | 0.100 | 0.8493 | 0.0260 | 0.0750 |
| 2 | 0.132 | 0.705 | -0.272 | 0.364 | 0.5840 | 0.0072 | 0.0750 |
| 3 | 0.791 | 0.768 | -1.126 | 0.614 | 0.5077 | 0.0044 | 0.0750 |
| 4 | 0.396 | 0.857 | -0.950 | 0.697 | 0.5026 | 0.0028 | 0.0750 |
| 5 | 0.198 | 0.986 | -1.022 | 0.832 | 0.5007 | 0.0023 | 0.0750 |
| *6 | 0.200 | 1.035 | -1.006 | 0.801 | 0.5532 | 0.0039 | 0.0775 |
| 7 | 0.200 | 1.087 | -1.019 | 0.764 | 0.5682 | 0.0040 | 0.0765 |
| 8 | 0.197 | 1.141 | -1.033 | 0.727 | 0.5814 | 0.0040 | 0.0753 |
| 9 | 0.193 | 1.152 | -1.033 | 0.718 | 0.5841 | 0.0040 | 0.0750 |

* begin lift maximization.

**Table 4e.** TSD Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 5, initial interpolations.

| Design cycle | Design Parameters | | | | $C_l$ | $C_d$ | A |
|---|---|---|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | | | |
| 0 | 0.0 | 0.0 | 0.8 | 0.0 | 0.6811 | 0.0342 | 0.0787 |
| 1 | 0.087 | 0.037 | 0.625 | 0.038 | 0.5448 | 0.0155 | 0.0753 |
| 2 | 0.520 | 0.221 | 0.145 | 0.009 | 0.5501 | 0.0079 | 0.0750 |
| 3 | 1.076 | 0.279 | -0.366 | 0.009 | 0.5483 | 0.0071 | 0.0750 |
| 4 | 0.935 | 0.160 | -0.184 | 0.037 | 0.5000 | 0.0051 | 0.0750 |
| 5 | 0.712 | 0.354 | -0.340 | 0.224 | 0.4991 | 0.0041 | 0.0750 |
| 6 | 0.491 | 0.582 | -0.544 | 0.431 | 0.5027 | 0.0033 | 0.0750 |
| 7 | 0.395 | 0.708 | -0.683 | 0.550 | 0.5031 | 0.0031 | 0.0750 |
| 8 | 0.289 | 0.822 | -0.806 | 0.671 | 0.5004 | 0.0027 | 0.0750 |
| 9 | 0.222 | 0.908 | -0.909 | 0.760 | 0.4993 | 0.0024 | 0.0750 |
| 10 | 0.181 | 1.016 | -1.067 | 0.866 | 0.4998 | 0.0022 | 0.0750 |
| *11 | 0.180 | 1.220 | -1.125 | 0.770 | 0.5913 | 0.0040 | 0.0753 |

**Table 4f.** TSD Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 6, initial interpolations.

| Design cycle | Design Parameters | | | | $C_l$ | $C_d$ | A |
|---|---|---|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | | | |
| 0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.5860 | 0.0279 | 0.0996 |
| 1 | 0.087 | 0.097 | -0.026 | 0.807 | 0.5291 | 0.0172 | 0.0926 |
| 2 | 0.280 | 0.472 | -0.153 | 0.306 | 0.5232 | 0.0053 | 0.0750 |
| 3 | 0.865 | 0.705 | -0.921 | 0.399 | 0.5530 | 0.0057 | 0.0750 |
| 4 | 0.433 | 0.701 | -0.693 | 0.534 | 0.5060 | 0.0032 | 0.0750 |
| 5 | 0.311 | 0.786 | -0.761 | 0.636 | 0.5005 | 0.0028 | 0.0750 |
| 6 | 0.169 | 1.004 | -1.016 | 0.836 | 0.5048 | 0.0023 | 0.0750 |
| *7 | 0.170 | 1.054 | -0.999 | 0.809 | 0.5627 | 0.0042 | 0.0779 |
| 8 | 0.170 | 1.107 | -1.019 | 0.771 | 0.5693 | 0.0039 | 0.0762 |
| 9 | 0.169 | 1.162 | -1.038 | 0.738 | 0.5837 | 0.0040 | 0.0752 |
| 10 | 0.166 | 1.174 | -1.039 | 0.729 | 0.5871 | 0.0040 | 0.0750 |

* begin lift maximization.

**Table 5a.** TSD Design Strategy A : Approximate Optimization with Tight Move Limits – initial condition 1.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.5 | 0.5 | -0.5 | 0.5 | 0.5341 | 0.0051 | 0.0805 |
| 1 | 0.501 | 0.525 | -0.520 | 0.476 | 0.5185 | 0.0040 | 0.0782 |
| – | — | — | — | — | — | — | — |
| 10 | 0.483 | 0.748 | -0.667 | 0.431 | 0.5660 | 0.0040 | 0.0750 |
| – | — | — | — | — | — | — | — |
| 20 | 0.355 | 0.977 | -0.891 | 0.580 | 0.5944 | 0.0040 | 0.0750 |
| – | — | — | — | — | — | — | — |
| 30 | 0.245 | 1.172 | -1.081 | 0.706 | 0.6194 | 0.0040 | 0.0750 |
| – | — | — | — | — | — | — | — |
| 40 | 0.163 | 1.359 | -1.277 | 0.823 | 0.6453 | 0.0040 | 0.0750 |
| – | — | — | — | — | — | — | — |

**Table 5b.** TSD Design Strategy A : Approximate Optimization with Tight Move Limits – initial condition 2.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.8 | -0.4 | 0.7 | -0.3 | 0.4190 | 0.0044 | 0.0739 |
| 1 | 0.841 | -0.421 | 0.670 | -0.283 | 0.4112 | 0.0041 | 0.0744 |
| – | — | — | — | — | — | — | — |
| 10 | 1.104 | -0.372 | 0.438 | -0.303 | 0.4301 | 0.0040 | 0.0750 |
| – | — | — | — | — | — | — | — |
| 20 | 1.173 | -0.313 | 0.299 | -0.268 | 0.4359 | 0.0040 | 0.0750 |
| – | — | — | — | — | — | — | — |
| 30 | 1.159 | -0.248 | 0.215 | -0.224 | 0.4434 | 0.0040 | 0.0750 |
| – | — | — | — | — | — | — | — |
| 40 | 1.135 | -0.200 | 0.166 | -0.193 | 0.4490 | 0.0040 | 0.0750 |
| – | — | — | — | — | — | — | — |

**Table 6a.** TSD Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 1.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.5 | 0.5 | -0.5 | 0.5 | 0.5341 | 0.0051 | 0.0805 |
| 1 | 0.024 | 1.500 | -1.394 | 0.943 | 0.6548 | 0.0037 | 0.0750 |
| 2 | -0.100 | 1.543 | -1.796 | 1.408 | 0.5456 | 0.0013 | 0.0750 |
| 3 | -0.080 | 1.484 | -1.912 | 1.550 | 0.4970 | 0.0006 | 0.0750 |
| *4 | -0.066 | 1.781 | -1.827 | 1.240 | 0.6771 | 0.0037 | 0.0763 |
| 5 | -0.079 | 1.958 | -1.963 | 1.234 | 0.7257 | 0.0041 | 0.0750 |

* begin lift maximization.

**Table 6b.** TSD Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 2.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.8 | -0.4 | 0.7 | -0.3 | 0.4190 | 0.0044 | 0.0739 |
| 1 | 1.236 | -0.137 | 0.192 | -0.351 | 0.5250 | 0.0068 | 0.0750 |
| 2 | 1.139 | -0.039 | 0.024 | -0.181 | 0.4995 | 0.0053 | 0.0750 |
| 3 | 0.897 | 0.119 | -0.052 | -0.029 | 0.5042 | 0.0046 | 0.0750 |
| 4 | 0.878 | 0.197 | -0.183 | 0.055 | 0.5025 | 0.0041 | 0.0750 |
| 5 | 0.724 | 0.349 | -0.337 | 0.216 | 0.5017 | 0.0035 | 0.0750 |
| 6 | -0.080 | 1.386 | -1.573 | 1.293 | 0.5245 | 0.0012 | 0.0750 |
| 7 | -0.177 | 1.525 | -1.870 | 1.558 | 0.5044 | 0.0007 | 0.0755 |
| *8 | -0.148 | 1.830 | -1.807 | 1.246 | 0.6868 | 0.0037 | 0.0760 |
| 9 | -0.173 | 2.047 | -1.991 | 1.270 | 0.7428 | 0.0042 | 0.0750 |

* begin lift maximization.

**Table 6c.** TSD Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 3.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.5656 | 0.0103 | 0.0822 |
| 1 | 1.024 | 0.100 | -0.075 | -0.096 | 0.5195 | 0.0054 | 0.0750 |
| 2 | 0.773 | 0.280 | -0.244 | 0.137 | 0.5041 | 0.0038 | 0.0750 |
| 3 | 0.631 | 0.417 | -0.393 | 0.295 | 0.4999 | 0.0032 | 0.0750 |
| 4 | 0.504 | 0.516 | -0.476 | 0.404 | 0.4989 | 0.0028 | 0.0750 |
| 5 | 0.418 | 0.650 | -0.648 | 0.542 | 0.5005 | 0.0025 | 0.0750 |
| 6 | 0.071 | 1.038 | -1.072 | 0.945 | 0.5080 | 0.0017 | 0.0750 |
| 7 | 0.080 | 1.308 | -1.614 | 1.261 | 0.5124 | 0.0011 | 0.0750 |
| 8 | 0.001 | 1.475 | -1.938 | 1.516 | 0.5049 | 0.0006 | 0.0750 |
| *9 | 0.001 | 1.770 | -1.863 | 1.215 | 0.6696 | 0.0034 | 0.0750 |
| 10 | 0.001 | 1.876 | -1.940 | 1.209 | 0.7078 | 0.0041 | 0.0750 |

* begin lift maximization.

**Table 6d.** TSD Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 4.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0676 | 0.0459 | 0.0771 |
| 1 | 0.085 | 0.880 | -0.100 | 0.100 | 0.8285 | 0.0228 | 0.0750 |
| 2 | 0.499 | 0.819 | -0.598 | 0.296 | 0.6385 | 0.0061 | 0.0751 |
| 3 | 0.503 | 0.751 | -0.836 | 0.578 | 0.5220 | 0.0028 | 0.0750 |
| 4 | -0.037 | 0.989 | -0.903 | 0.905 | 0.5054 | 0.0020 | 0.0750 |
| 5 | 0.035 | 1.205 | -1.373 | 1.141 | 0.5085 | 0.0013 | 0.0750 |
| 6 | -0.025 | 1.469 | -1.887 | 1.493 | 0.5069 | 0.0007 | 0.0750 |
| *7 | -0.022 | 1.762 | -1.812 | 1.194 | 0.6760 | 0.0036 | 0.0755 |
| 8 | -0.025 | 1.878 | -1.889 | 1.179 | 0.7149 | 0.0042 | 0.0750 |

* begin lift maximization.

**Table 6e.** TSD Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 5.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.0 | 0.0 | 0.8 | 0.0 | 0.6491 | 0.0297 | 0.0786 |
| 1 | 0.099 | 0.088 | 0.596 | 0.014 | 0.5509 | 0.0145 | 0.0753 |
| 2 | 0.594 | 0.199 | 0.131 | -0.021 | 0.5421 | 0.0066 | 0.0750 |
| 3 | 1.036 | 0.286 | -0.376 | 0.046 | 0.5351 | 0.0052 | 0.0750 |
| 4 | 0.681 | 0.382 | -0.375 | 0.263 | 0.4978 | 0.0033 | 0.0750 |
| 5 | 0.501 | 0.502 | -0.446 | 0.389 | 0.4996 | 0.0029 | 0.0750 |
| 6 | 0.428 | 0.747 | -0.817 | 0.625 | 0.5070 | 0.0024 | 0.0750 |
| 7 | 0.198 | 0.878 | -0.898 | 0.793 | 0.5007 | 0.0019 | 0.0750 |
| 8 | 0.173 | 0.976 | -1.060 | 0.896 | 0.5026 | 0.0017 | 0.0750 |
| 9 | 0.064 | 1.173 | -1.257 | 1.133 | 0.4997 | 0.0014 | 0.0750 |
| 10 | 0.058 | 1.502 | -1.991 | 1.502 | 0.5192 | 0.0007 | 0.0750 |
| *11 | 0.066 | 1.802 | -1.977 | 1.250 | 0.6762 | 0.0036 | 0.0750 |
| 12 | 0.053 | 1.842 | -1.962 | 1.213 | 0.6979 | 0.0041 | 0.0750 |
| 13 | 0.043 | 1.904 | -2.068 | 1.279 | 0.7034 | 0.0041 | 0.0750 |

* begin lift maximization.

**Table 6f.** TSD Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 6.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.5534 | 0.0242 | 0.0996 |
| 1 | 0.098 | 0.100 | -0.028 | 0.796 | 0.5111 | 0.0146 | 0.0924 |
| 2 | 0.504 | 0.422 | -0.167 | 0.174 | 0.5464 | 0.0051 | 0.0750 |
| 3 | 0.686 | 0.484 | -0.500 | 0.303 | 0.5188 | 0.0037 | 0.0750 |
| 4 | 0.001 | 0.981 | -0.880 | 0.858 | 0.5155 | 0.0022 | 0.0750 |
| 5 | 0.002 | 1.062 | -1.082 | 0.992 | 0.5032 | 0.0016 | 0.0750 |
| 6 | 0.005 | 1.198 | -1.352 | 1.150 | 0.5034 | 0.0012 | 0.0750 |
| 7 | 0.006 | 1.327 | -1.627 | 1.320 | 0.5024 | 0.0009 | 0.0750 |
| 8 | -0.005 | 1.458 | -1.902 | 1.500 | 0.5031 | 0.0007 | 0.0750 |
| *9 | -0.005 | 1.896 | -2.027 | 1.284 | 0.6962 | 0.0038 | 0.0750 |
| 10 | -0.006 | 1.923 | -2.033 | 1.270 | 0.7091 | 0.0040 | 0.0750 |

* begin lift maximization.

**Table 7a.** TSD Design Strategy C : Approximate Optimization with Absolute Move Limits – initial condition 3.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.5656 | 0.0103 | 0.0822 |
| 1 | 1.123 | 0.347 | -0.491 | 0.041 | 0.5594 | 0.0061 | 0.0750 |
| 2 | 0.640 | 0.524 | -0.452 | 0.263 | 0.5450 | 0.0043 | 0.0750 |
| 3 | 0.518 | 0.881 | -0.844 | 0.473 | 0.5966 | 0.0045 | 0.0750 |
| 4 | 0.032 | 1.257 | -1.000 | 0.737 | 0.6407 | 0.0044 | 0.0750 |
| 5 | 0.155 | 1.381 | -1.249 | 0.785 | 0.6649 | 0.0046 | 0.0750 |
| 6 | -0.006 | 1.524 | -1.374 | 0.929 | 0.6672 | 0.0041 | 0.0750 |
| 7 | -0.099 | 1.667 | -1.504 | 1.024 | 0.6853 | 0.0041 | 0.0750 |
| 8 | -0.224 | 1.916 | -1.752 | 1.179 | 0.7218 | 0.0041 | 0.0750 |

**Table 7b.** TSD Design Strategy C : Approximate Optimization with Absolute Move Limits – initial condition 4.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0676 | 0.0459 | 0.0771 |
| 1 | 1.262 | 0.537 | -0.500 | 0.500 | 0.3825 | 0.0001 | 0.0641 |
| 2 | 0.504 | 0.923 | -0.942 | 0.549 | 0.5868 | 0.0042 | 0.0750 |
| 3 | 0.034 | 1.299 | -1.045 | 0.747 | 0.6512 | 0.0045 | 0.0750 |
| 4 | 0.111 | 1.441 | -1.294 | 0.818 | 0.6746 | 0.0047 | 0.0750 |
| 5 | -0.040 | 1.570 | -1.426 | 0.974 | 0.6685 | 0.0039 | 0.0750 |
| 6 | -0.191 | 1.790 | -1.592 | 1.092 | 0.7059 | 0.0042 | 0.0750 |
| 7 | -0.265 | 1.917 | -1.734 | 1.194 | 0.7176 | 0.0040 | 0.0750 |
| 8 | -0.332 | 2.035 | -1.841 | 1.264 | 0.7360 | 0.0040 | 0.0750 |

**Table 7c.** TSD Design Strategy C : Approximate Optimization with Absolute Move Limits – initial condition 5.

| Design cycle | Design Parameters | | | | $C_l$ | $C_d$ | A |
|---|---|---|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | | | |
| 0 | 0.0 | 0.0 | 0.8 | 0.0 | 0.6491 | 0.0297 | 0.0786 |
| 1 | 0.419 | 0.018 | 0.301 | 0.096 | 0.4497 | 0.0045 | 0.0750 |
| 2 | 0.912 | 0.339 | -0.199 | -0.067 | 0.5953 | 0.0072 | 0.0750 |
| 3 | 0.945 | 0.551 | -0.696 | 0.231 | 0.5659 | 0.0056 | 0.0750 |
| 4 | 0.447 | 0.749 | -0.639 | 0.433 | 0.5661 | 0.0041 | 0.0750 |
| 5 | 0.311 | 1.150 | -1.060 | 0.649 | 0.6303 | 0.0045 | 0.0750 |
| 6 | -0.058 | 1.558 | -1.371 | 0.942 | 0.6744 | 0.0042 | 0.0750 |
| 7 | -0.238 | 1.859 | -1.657 | 1.141 | 0.7139 | 0.0042 | 0.0750 |
| 8 | -0.539 | 2.340 | -2.069 | 1.423 | 0.7922 | 0.0041 | 0.0750 |

**Table 7d.** TSD Design Strategy C : Approximate Optimization with Absolute Move Limits – initial condition 6.

| Design cycle | Design Parameters | | | | $C_l$ | $C_d$ | A |
|---|---|---|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | | | |
| 0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.5534 | 0.0242 | 0.0996 |
| 1 | 0.489 | 0.500 | -0.497 | 0.503 | 0.5284 | 0.0048 | 0.0802 |
| 2 | 0.623 | 0.961 | -0.994 | 0.473 | 0.6252 | 0.0053 | 0.0750 |
| 3 | 0.124 | 1.242 | -1.003 | 0.676 | 0.6546 | 0.0048 | 0.0750 |
| 4 | 0.066 | 1.621 | -1.500 | 0.921 | 0.7035 | 0.0047 | 0.0750 |
| 5 | -0.434 | 2.077 | -1.749 | 1.224 | 0.7638 | 0.0042 | 0.0750 |

**Table 8.** Comparison of Lift and Drag Predicted by the TSD and Euler Analysis.

| Design | Design Parameters | | | | $C_l$ | | $C_d$ | |
|---|---|---|---|---|---|---|---|---|
| No. * | $X_1$ | $X_2$ | $X_3$ | $X_4$ | TSD | Euler | TSD | Euler |
| 0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.5656 | 0.4891 | 0.01027 | 0.01177 |
| 4 | 0.504 | 0.516 | -0.476 | 0.404 | 0.4989 | 0.4725 | 0.00284 | 0.00537 |
| 8 | 0.001 | 1.475 | -1.938 | 1.516 | 0.5049 | 0.4683 | 0.00063 | 0.00197 |
| 10 | 0.001 | 1.876 | -1.940 | 1.209 | 0.7078 | 0.6161 | 0.00414 | 0.00739 |

* selected from the design results in Table 6c.

**Table 9a.** Euler Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 3.

| Move Limits | Design cycle | Design Parameters | | | | | | | ** |
|---|---|---|---|---|---|---|---|---|---|
| | | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A | $C_{l_{eqv.}}$ |
| 500% | 0 | 1.0 | 0.0 | 0.0 | 0.0 | .48912 | .011768 | 0.0810 | |
| | 1 | 0.827 | 0.100 | 0.094 | -0.088 | .49602 | .009069 | 0.0750 | |
| | 2 | 0.555 | 0.331 | -0.108 | 0.157 | .49838 | .007961 | 0.0750 | |
| | 3 | 0.325 | 0.558 | -0.346 | 0.405 | .49742 | .006879 | 0.0750 | |
| | 4 | -0.313 | 1.150 | -0.923 | 1.040 | .49438 | .005745 | 0.0750 | |
| 50% | *5 | -0.270 | 1.645 | -1.231 | 0.927 | .65665 | .009951 | 0.0750 | |
| | 6 | -0.375 | 1.924 | -1.549 | 1.110 | .68561 | .009783 | 0.0750 | |
| | 7 | -0.480 | 2.067 | -1.665 | 1.200 | .70255 | .010024 | 0.0750 | |
| | $^{\Delta}$8 | -0.632 | 2.218 | -1.777 | 1.319 | .71358 | .010223 | 0.0750 | .70617 |
| | 9 | -0.721 | 2.277 | -1.822 | 1.390 | .71068 | .010204 | 0.0750 | .70358 |
| 25% | 10 | -0.679 | 2.252 | -1.822 | 1.375 | .70759 | .010026 | 0.0750 | .70669 |
| | $^{\Delta}$11 | -0.729 | 2.303 | -1.870 | 1.425 | .70838 | .010026 | 0.0750 | .70746 |
| | 12 | -0.792 | 2.350 | -1.905 | 1.474 | .70806 | .010084 | 0.0750 | .70484 |
| 12.5% | 13 | -0.729 | 2.295 | -1.860 | 1.421 | .70698 | .010019 | 0.0750 | .70660 |
| 2% | 14 | -0.721 | 2.302 | -1.877 | 1.426 | .70759 | .009973 | 0.0750 | .70864 |
| | 15 | -0.717 | 2.307 | -1.885 | 1.426 | .70899 | .009979 | 0.0750 | .70980 |
| | 16 | -0.712 | 2.309 | -1.888 | 1.424 | .70995 | .009985 | 0.0750 | .71054 |
| | 17 | -0.711 | 2.317 | -1.900 | 1.429 | .71090 | .009979 | 0.0750 | .71173 |
| | 18 | -0.713 | 2.327 | -1.913 | 1.435 | .71208 | .009985 | 0.0750 | .71267 |
| | 19 | -0.711 | 2.328 | -1.914 | 1.434 | .71275 | .009994 | 0.0750 | .71297 |
| | 20 | -0.711 | 2.331 | -1.918 | 1.435 | .71307 | .009994 | 0.0750 | .71333 |
| | 21 | -0.710 | 2.331 | -1.918 | 1.434 | .71335 | .009996 | 0.0750 | .71349 |
| | 22 | -0.719 | 2.344 | -1.931 | 1.446 | .71384 | .009999 | 0.0750 | .71388 |
| | 23 | -0.725 | 2.351 | -1.939 | 1.452 | .71404 | .010001 | 0.0750 | .71402 |
| | 24 | -0.736 | 2.365 | -1.954 | 1.465 | .71427 | .009998 | 0.0750 | .71435 |
| | 25 | -0.736 | 2.365 | -1.953 | 1.465 | .71439 | .010002 | 0.0750 | .71432 |

* begin lift maximization.

$^{\Delta}$ starting point for reduced move limits.

** $C_{l_{eqv.}}$ : Equivalent Value of $C_l$.

**Table 9b.** Euler Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 4.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.6977 | 0.0266 | 0.0759 |
| 1 | 0.100 | 0.624 | 0.100 | 0.100 | 0.5827 | 0.0165 | 0.0750 |
| 2 | 0.307 | 0.481 | -0.244 | 0.378 | 0.4846 | 0.0068 | 0.0750 |
| 3 | 0.152 | 0.777 | -0.600 | 0.627 | 0.4992 | 0.0061 | 0.0750 |
| 4 | -0.008 | 0.936 | -0.762 | 0.796 | 0.4984 | 0.0058 | 0.0750 |
| 5 | -0.093 | 1.031 | -0.861 | 1.890 | 0.4996 | 0.0056 | 0.0750 |
| 6 | -0.260 | 1.164 | -0.973 | 1.036 | 0.4982 | 0.0056 | 0.0750 |
| *7 | -0.254 | 1.657 | -1.268 | 0.939 | 0.6565 | 0.0097 | 0.0750 |
| 8 | -0.331 | 1.822 | -1.425 | 1.031 | 0.6783 | 0.0099 | 0.0750 |
| 9 | -0.401 | 1.956 | -1.563 | 1.121 | 0.6921 | 0.0100 | 0.0750 |
| 10 | -0.552 | 2.148 | -1.728 | 1.259 | 0.7098 | 0.0101 | 0.0750 |
| 11 | -0.640 | 2.233 | -1.801 | 1.337 | 0.7127 | 0.0101 | 0.0750 |

* begin lift maximization.

**Table 9c.** Euler Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 5.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.5050 | 0.0467 | 0.0967 |
| 1 | 0.100 | -0.056 | 0.634 | 0.100 | 0.4146 | 0.0112 | 0.0750 |
| 2 | 0.567 | 0.189 | 0.172 | -0.019 | 0.5044 | 0.0100 | 0.0750 |
| 3 | 0.637 | 0.260 | -0.040 | 0.077 | 0.4989 | 0.0084 | 0.0750 |
| 4 | 0.445 | 0.427 | -0.200 | 0.263 | 0.4978 | 0.0075 | 0.0750 |
| 5 | 0.061 | 0.739 | -0.472 | 0.656 | 0.4962 | 0.0066 | 0.0750 |
| 6 | -0.111 | 0.949 | -0.701 | 0.810 | 0.4999 | 0.0062 | 0.0750 |
| 7 | -0.214 | 1.127 | -0.938 | 0.992 | 0.4997 | 0.0056 | 0.0750 |
| 8 | -0.272 | 1.221 | -1.057 | 1.084 | 0.4999 | 0.0055 | 0.0750 |
| *9 | -0.278 | 1.729 | -1.329 | 0.964 | 0.6704 | 0.0100 | 0.0750 |
| 10 | -0.412 | 1.979 | -1.592 | 1.140 | 0.6933 | 0.0099 | 0.0750 |
| 11 | -0.558 | 2.154 | -1.730 | 1.261 | 0.7111 | 0.0102 | 0.0750 |
| 12 | -0.649 | 2.237 | -1.803 | 1.344 | 0.7121 | 0.0101 | 0.0750 |

* begin lift maximization.

**Table 9d.** Euler Design Strategy B : Drag Minimization followed by Lift Maximization – initial condition 6.

| Design | Design Parameters | | | | | | |
|---|---|---|---|---|---|---|---|
| cycle | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A |
| 0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.3558 | 0.0165 | 0.0984 |
| 1 | 0.200 | 0.200 | 0.200 | 0.481 | 0.5045 | 0.0303 | 0.0980 |
| 2 | 0.424 | 0.336 | -0.524 | 0.669 | 0.3727 | 0.0027 | 0.0750 |
| 3 | 0.169 | 0.817 | -0.640 | 0.622 | 0.5102 | 0.0064 | 0.0750 |
| 4 | -0.180 | 1.070 | -0.882 | 0.953 | 0.4946 | 0.0055 | 0.0750 |
| *5 | -0.178 | 1.560 | -1.179 | 0.865 | 0.6490 | 0.0098 | 0.0750 |
| 6 | -0.222 | 1.677 | -1.299 | 0.928 | 0.6649 | 0.0099 | 0.0750 |
| 7 | -0.280 | 1.777 | -1.391 | 0.990 | 0.6764 | 0.0100 | 0.0750 |
| 8 | -0.409 | 1.982 | -1.591 | 1.134 | 0.6959 | 0.0100 | 0.0750 |
| 9 | -0.558 | 2.148 | -1.721 | 1.257 | 0.7105 | 0.0102 | 0.0750 |
| 10 | -0.630 | 2.206 | -1.772 | 1.322 | 0.7090 | 0.0101 | 0.0750 |

* begin lift maximization.

**Table 10a.** Euler Design Strategy C : Approximate Optimization with Absolute Move Limits – initial condition 3.

| Move Limits | Design cycle | Design Parameters | | | | | | | ** |
|---|---|---|---|---|---|---|---|---|---|
| | | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A | $C_{l_{equv.}}$ |
| | 0 | 1.0 | 0.0 | 0.0 | 0.0 | .48912 | .011768 | 0.0810 | |
| | 1 | 0.652 | 0.500 | -0.220 | 0.057 | .56529 | .010919 | 0.0750 | |
| | 2 | 0.239 | 1.000 | -0.672 | 0.455 | .60127 | .010031 | 0.0750 | |
| | 3 | 0.071 | 1.267 | -0.923 | 0.634 | .62789 | .009978 | 0.0750 | |
| 0.5 | 4 | -0.078 | 1.470 | -1.098 | 0.772 | .64813 | .010042 | 0.0750 | |
| | 5 | -0.227 | 1.635 | -1.227 | 0.894 | .66244 | .010129 | 0.0750 | |
| | 6 | -0.341 | 1.792 | -1.380 | 1.018 | .67371 | .009992 | 0.0750 | |
| | 7 | -0.434 | 1.956 | -1.553 | 1.137 | .68770 | .009898 | 0.0750 | |
| | 8 | -0.514 | 2.082 | -1.675 | 1.226 | .69950 | .009951 | 0.0750 | |
| | $^{\Delta}9$ | -0.605 | 2.159 | -1.722 | 1.288 | .70507 | .010117 | 0.0750 | .70143 |
| | 10 | -0.677 | 2.199 | -1.749 | 1.342 | .70144 | .010089 | 0.0750 | .69855 |
| | 11 | -0.665 | 2.232 | -1.805 | 1.362 | .70542 | .009989 | 0.0750 | .70581 |
| 0.25 | 12 | -0.721 | 2.292 | -1.858 | 1.415 | .70795 | .010031 | 0.0750 | .70684 |
| | $^{\Delta}13$ | -0.768 | 2.340 | -1.907 | 1.465 | .70768 | .010012 | 0.0750 | .70723 |
| | 14 | -0.811 | 2.367 | -1.924 | 1.495 | .70710 | .010064 | 0.0750 | .70464 |
| 0.125 | $^{\Delta}15$ | -0.784 | 2.360 | -1.931 | 1.486 | .70734 | .009989 | 0.0750 | .70776 |
| | 16 | -0.810 | 2.379 | -1.942 | 1.504 | .70790 | .010040 | 0.0750 | .70632 |
| | 17 | -0.778 | 2.365 | -1.942 | 1.488 | .70809 | .009970 | 0.0750 | .70932 |
| | 18 | -0.777 | 2.372 | -1.950 | 1.490 | .70953 | .009984 | 0.0750 | .71022 |
| | 19 | -0.777 | 2.377 | -1.957 | 1.492 | .71030 | .009988 | 0.0750 | .71077 |
| | 20 | -0.777 | 2.381 | -1.962 | 1.494 | .71092 | .009992 | 0.0750 | .71127 |
| | 21 | -0.778 | 2.385 | -1.967 | 1.497 | .71133 | .009993 | 0.0750 | .71157 |
| | 22 | -0.779 | 2.389 | -1.973 | 1.500 | .71170 | .009994 | 0.0750 | .71197 |
| 0.02 | 23 | -0.780 | 2.393 | -1.978 | 1.503 | .71202 | .009994 | 0.0750 | .71226 |
| | 24 | -0.781 | 2.397 | -1.982 | 1.505 | .71226 | .009994 | 0.0750 | .71250 |
| | 25 | -0.783 | 2.401 | -1.987 | 1.509 | .71250 | .009995 | 0.0750 | .71273 |
| | 26 | -0.784 | 2.405 | -1.992 | 1.512 | .71272 | .009996 | 0.0750 | .71290 |
| | 27 | -0.786 | 2.408 | -1.997 | 1.515 | .71287 | .009997 | 0.0750 | .71304 |

continued in the next page

| Move Limits | Design cycle | Design Parameters | | | | | | | ** |
|---|---|---|---|---|---|---|---|---|---|
| | | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $C_l$ | $C_d$ | A | $C_{l_{equv.}}$ |
| | 28 | -0.788 | 2.412 | -2.001 | 1.518 | .71297 | .009997 | 0.0750 | .71312 |
| | 29 | -0.790 | 2.415 | -2.005 | 1.521 | .71310 | .009997 | 0.0750 | .71323 |
| | 30 | -0.792 | 2.419 | -2.009 | 1.524 | .71322 | .009998 | 0.0750 | .71331 |
| | 31 | -0.795 | 2.422 | -2.013 | 1.527 | .71324 | .009998 | 0.0750 | .71335 |
| | 32 | -0.797 | 2.425 | -2.017 | 1.530 | .71331 | .009998 | 0.0750 | .71341 |
| | 33 | -0.799 | 2.428 | -2.020 | 1.533 | .71335 | .009998 | 0.0750 | .71342 |
| | 34 | -0.801 | 2.432 | -2.024 | 1.537 | .71335 | .009998 | 0.0750 | .71345 |
| | 35 | -0.804 | 2.435 | -2.028 | 1.540 | .71336 | .009998 | 0.0750 | .71345 |
| | 36 | -0.807 | 2.438 | -2.032 | 1.543 | .71343 | .010000 | 0.0750 | .71339 |

$\Delta$ starting point for reduced move limits.

** $C_{l_{equv.}}$ : Equivalent Value of $C_l$.

**Table 10b.** Euler Design Strategy C: Approximate Optimization with Absolute Move Limits – initial condition 4.

| Design cycle | Design Parameters | | | | $C_l$ | $C_d$ | A |
|---|---|---|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | | | |
| 0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.6977 | 0.0266 | 0.0759 |
| 1 | 0.289 | 0.686 | -0.460 | 0.448 | 0.5197 | 0.0074 | 0.0750 |
| 2 | 0.177 | 1.080 | -0.746 | 0.517 | 0.6067 | 0.0099 | 0.0750 |
| 3 | 0.043 | 1.314 | -0.970 | 0.666 | 0.6324 | 0.0100 | 0.0750 |
| 4 | -0.140 | 1.593 | -1.227 | 0.855 | 0.6606 | 0.0100 | 0.0750 |
| 5 | -0.306 | 1.796 | -1.402 | 1.007 | 0.6774 | 0.0100 | 0.0750 |
| 6 | -0.445 | 1.993 | -1.588 | 1.153 | 0.6939 | 0.0100 | 0.0750 |
| 7 | -0.542 | 2.107 | -1.687 | 1.242 | 0.7023 | 0.0100 | 0.0750 |
| 8 | -0.636 | 2.194 | -1.759 | 1.323 | 0.7055 | 0.0101 | 0.0750 |

**Table 10c.** Euler Design Strategy C: Approximate Optimization with Absolute Move Limits – initial condition 5.

| Design cycle | Design Parameters | | | | $C_l$ | $C_d$ | A |
|---|---|---|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | | | |
| 0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.5050 | 0.0467 | 0.0967 |
| 1 | 0.486 | -0.285 | 0.500 | 0.091 | 0.3613 | 0.0050 | 0.0750 |
| 2 | 0.639 | 0.212 | 0.096 | -0.021 | 0.5105 | 0.0097 | 0.0750 |
| 3 | 0.541 | 0.478 | -0.180 | 0.125 | 0.5440 | 0.0099 | 0.0750 |
| 4 | 0.424 | 0.744 | -0.460 | 0.292 | 0.5728 | 0.0098 | 0.0750 |
| 5 | 0.212 | 1.094 | -0.780 | 0.511 | 0.6120 | 0.0100 | 0.0750 |
| 6 | -0.091 | 1.459 | -1.067 | 0.760 | 0.6486 | 0.0102 | 0.0750 |
| 7 | -0.244 | 1.665 | -1.264 | 0.922 | 0.6633 | 0.0100 | 0.0750 |
| 8 | -0.488 | 2.026 | -1.615 | 1.189 | 0.6936 | 0.0099 | 0.0750 |
| 9 | -0.578 | 2.135 | -1.708 | 1.271 | 0.7024 | 0.0100 | 0.0750 |
| 10 | -0.650 | 2.208 | -1.774 | 1.339 | 0.7049 | 0.0100 | 0.0750 |

**Table 10d.** Euler Design Strategy C: Approximate Optimization with Absolute Move Limits – initial condition 6.

| Design cycle | Design Parameters | | | | $C_l$ | $C_d$ | A |
|---|---|---|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | | | |
| 0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.3558 | 0.0165 | 0.0984 |
| 1 | 0.474 | 0.500 | -0.449 | 0.500 | 0.5110 | 0.0103 | 0.0822 |
| 2 | 0.324 | 1.000 | -0.679 | 0.392 | 0.6164 | 0.0108 | 0.0750 |
| 3 | 0.043 | 1.372 | -1.041 | 0.692 | 0.6417 | 0.0100 | 0.0750 |
| 4 | -0.150 | 1.574 | -1.186 | 0.837 | 0.6603 | 0.0102 | 0.0750 |
| 5 | -0.310 | 1.798 | -1.403 | 1.010 | 0.6770 | 0.0100 | 0.0750 |
| 6 | -0.417 | 1.974 | -1.584 | 1.140 | 0.6920 | 0.0099 | 0.0750 |
| 7 | -0.510 | 2.111 | -1.714 | 1.239 | 0.7041 | 0.0100 | 0.0750 |
| 8 | -0.591 | 2.179 | -1.754 | 1.292 | 0.7097 | 0.0101 | 0.0750 |
| 9 | -0.664 | 2.255 | -1.833 | 1.372 | 0.7090 | 0.0100 | 0.0750 |

# FIGURES

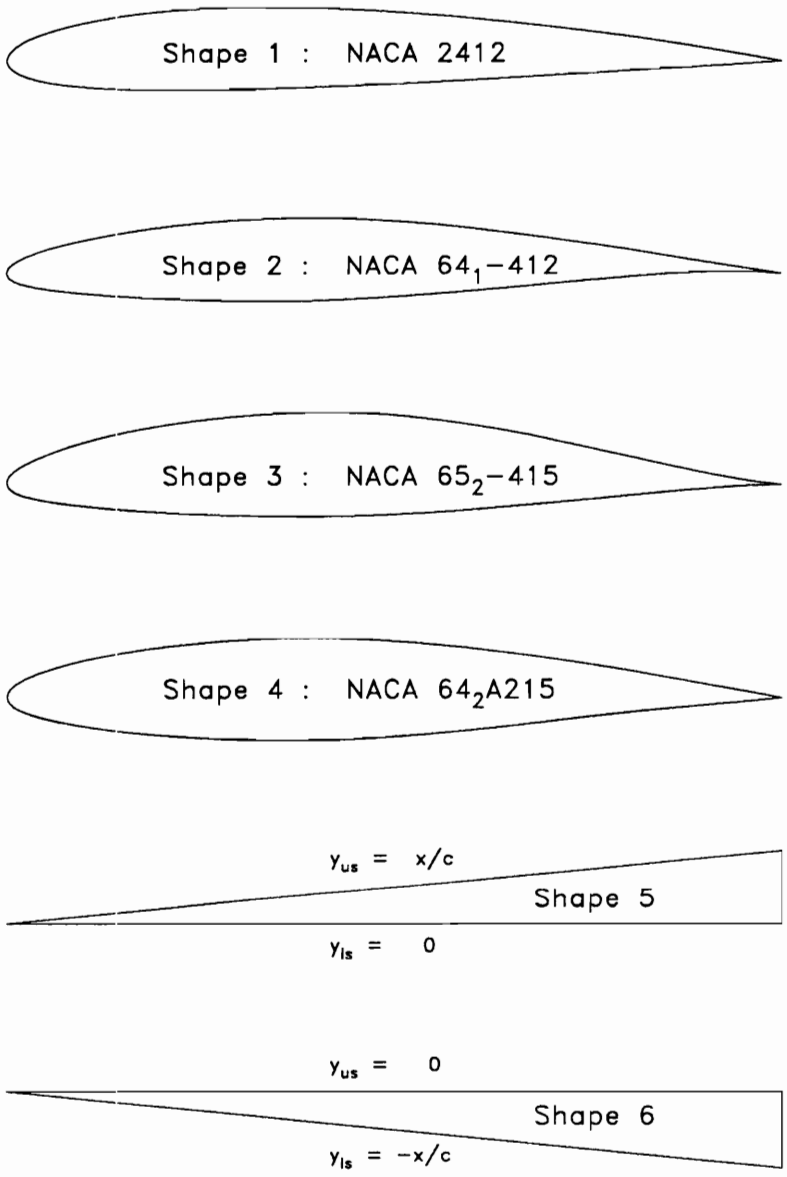



**Figure 1.** Six Shape Functions

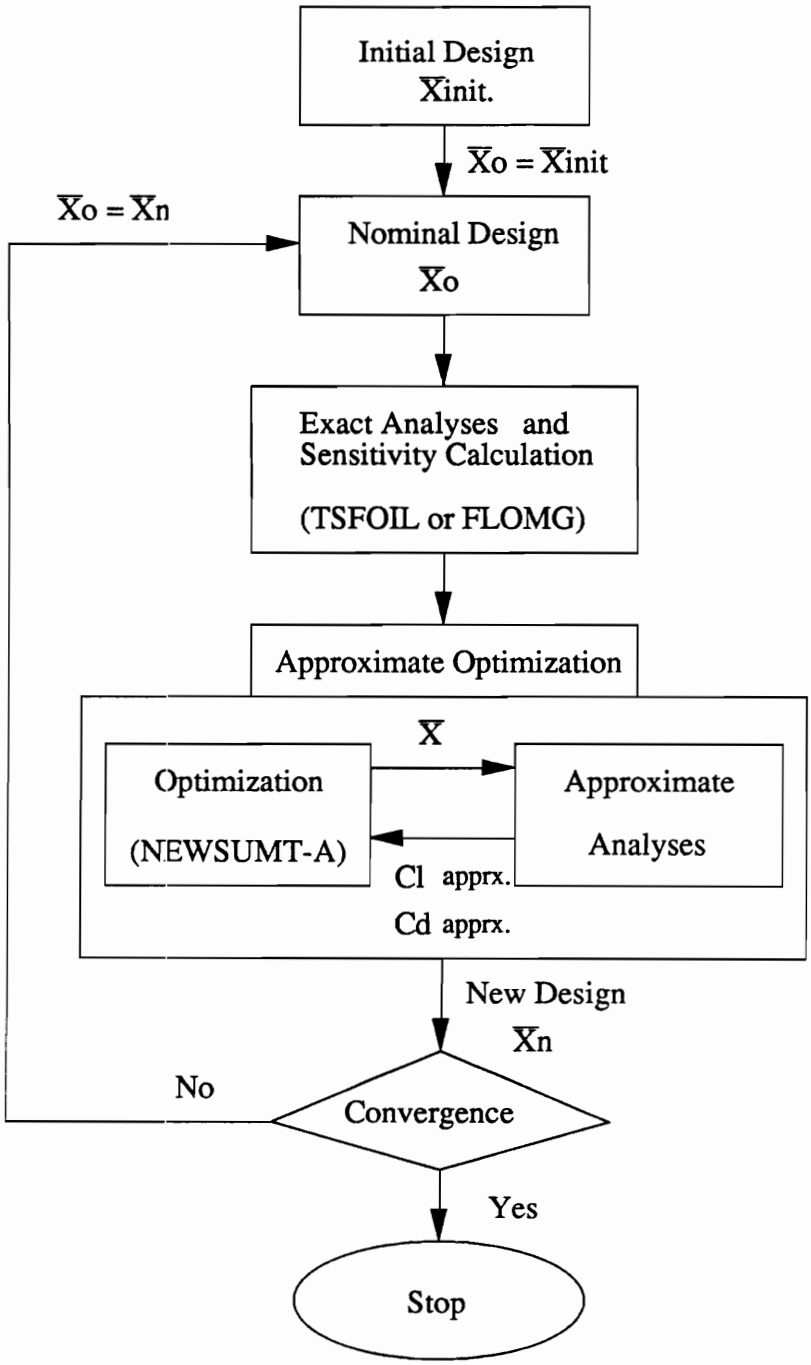



**Figure 2.** Flowchart of Approximate Optimization Design Procedure.

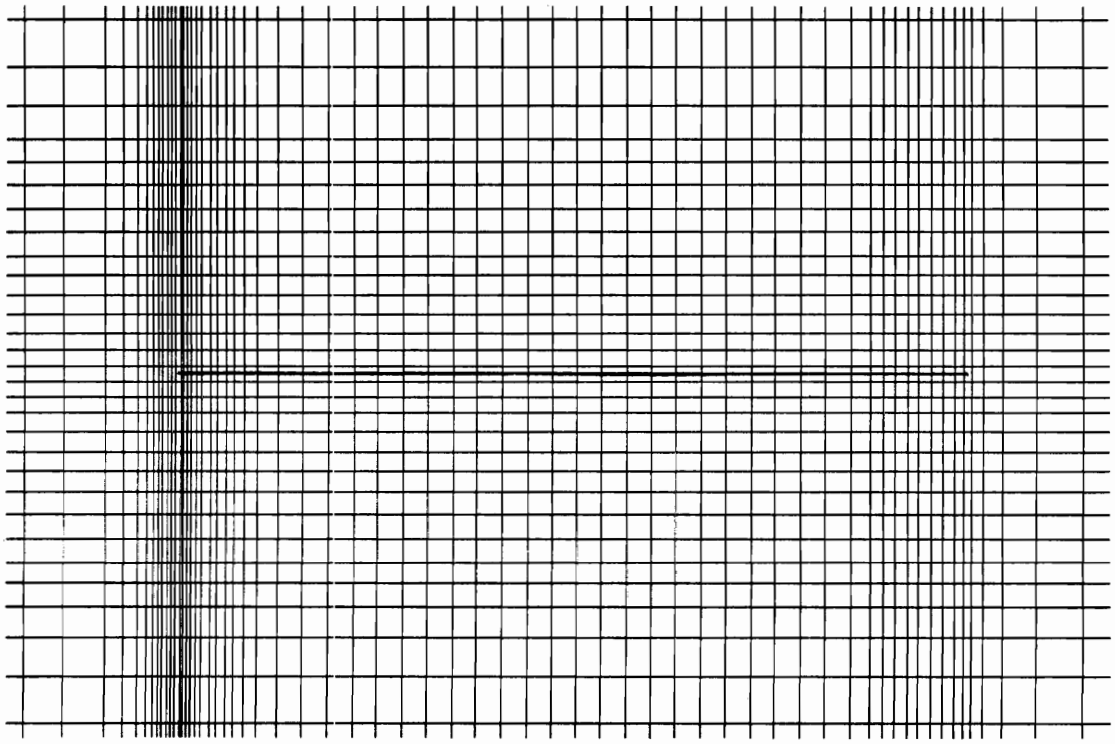


**Figure 3.** Close-Up View of TSFOIL Cartesian Mesh $77 \times 57$, NACA2412.

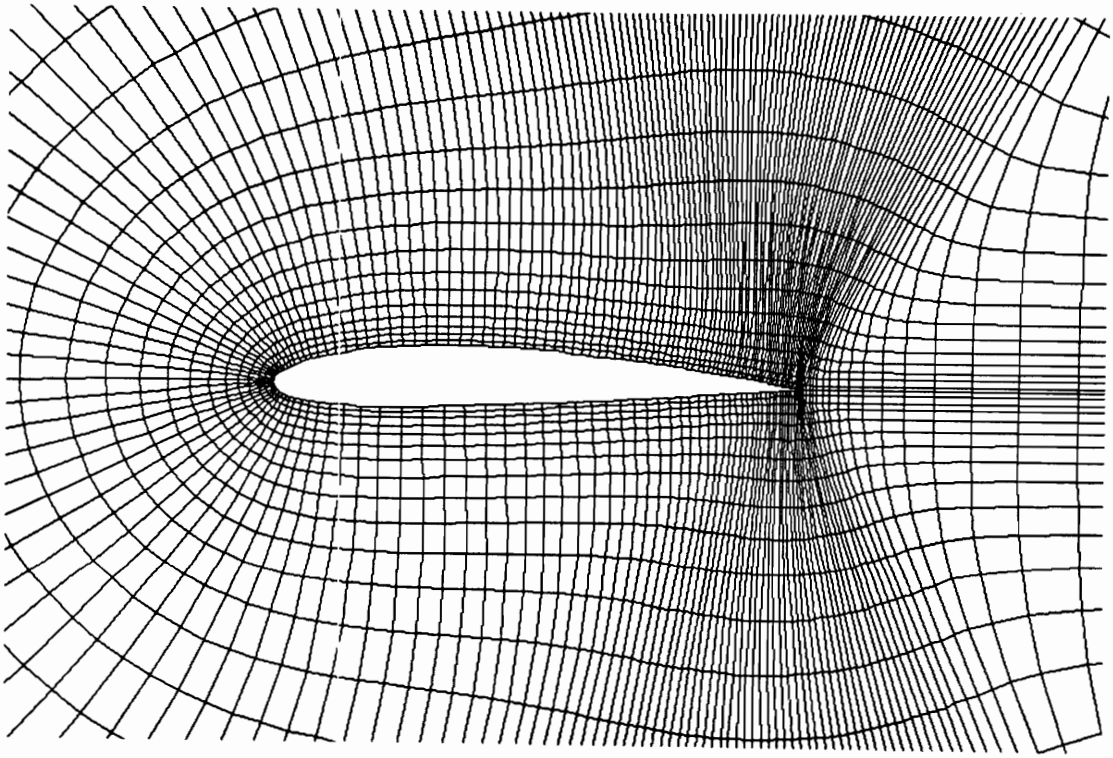


**Figure 4.** Close-Up View of FLOMG Hyperbolic Type C-Grid $224 \times 32$ cells, NACA2412.

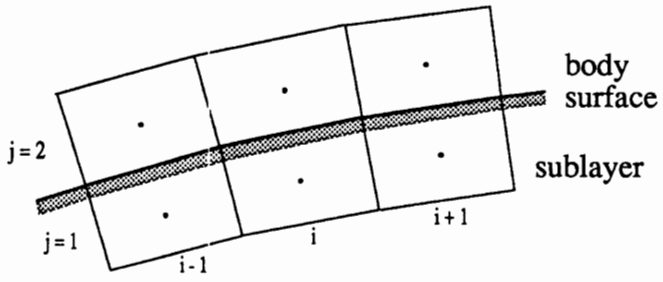



**Figure 5a.** Sublayer

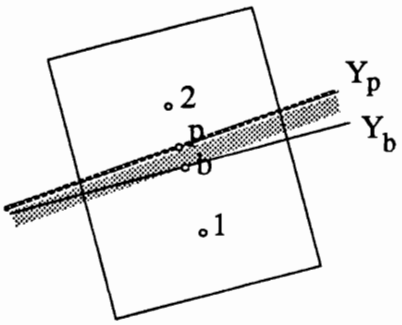



**Figure 5b.** Notations for Modified Euler Boundary Conditions

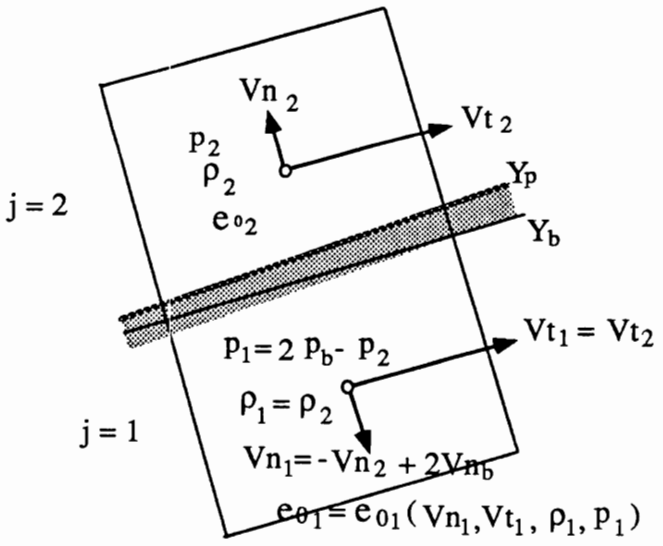



**Figure 5c.** Reflection procedure for Implementing Modified Euler Boundary Conditions.

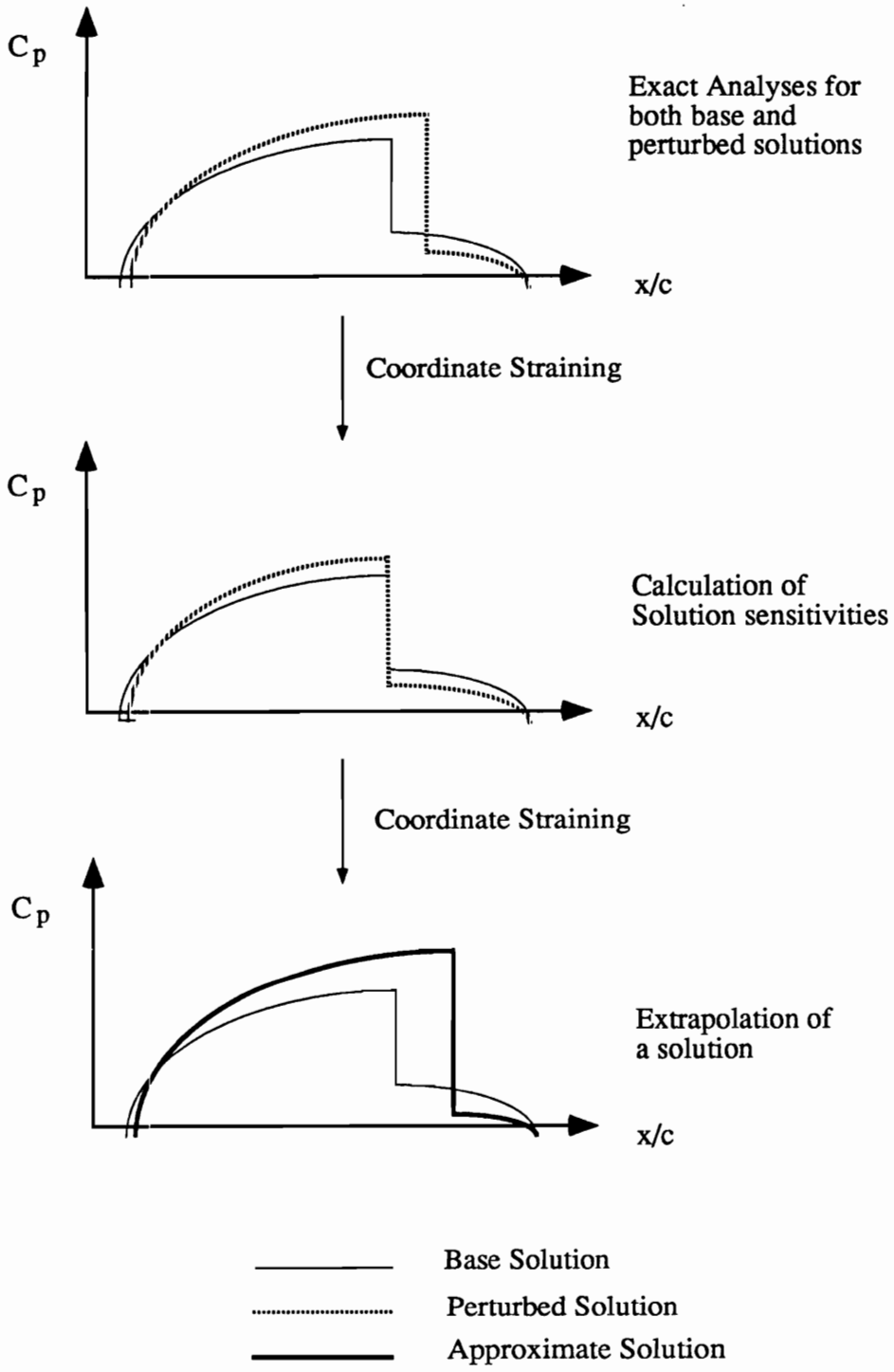
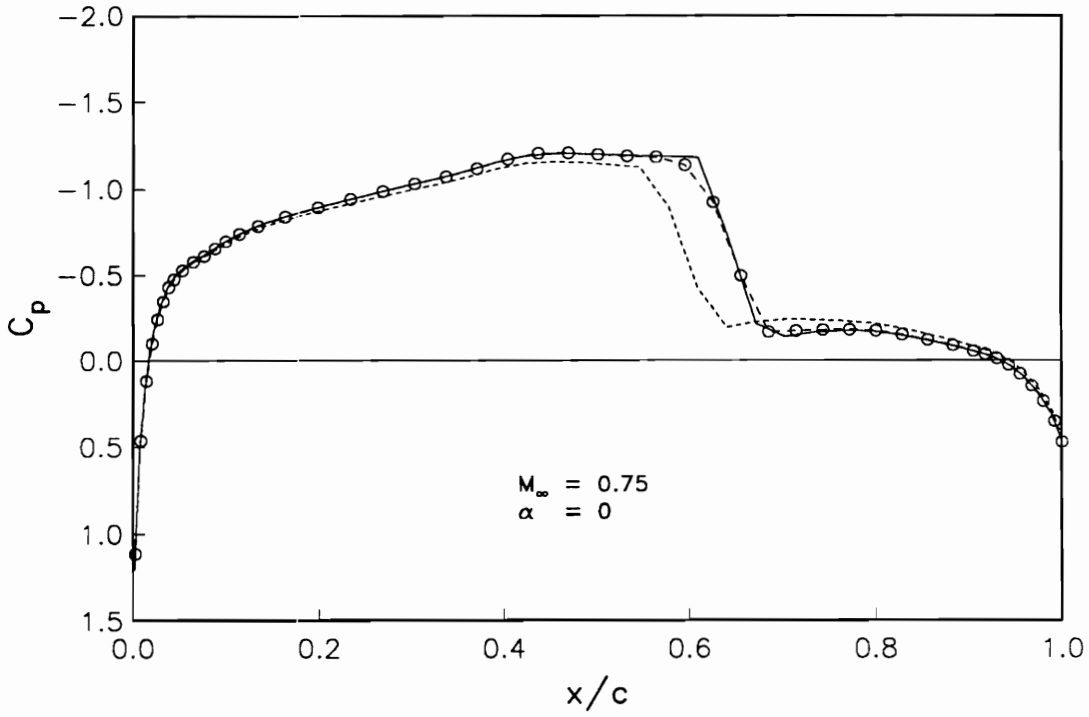



**Figure 6.** Method of Strained Coordinates

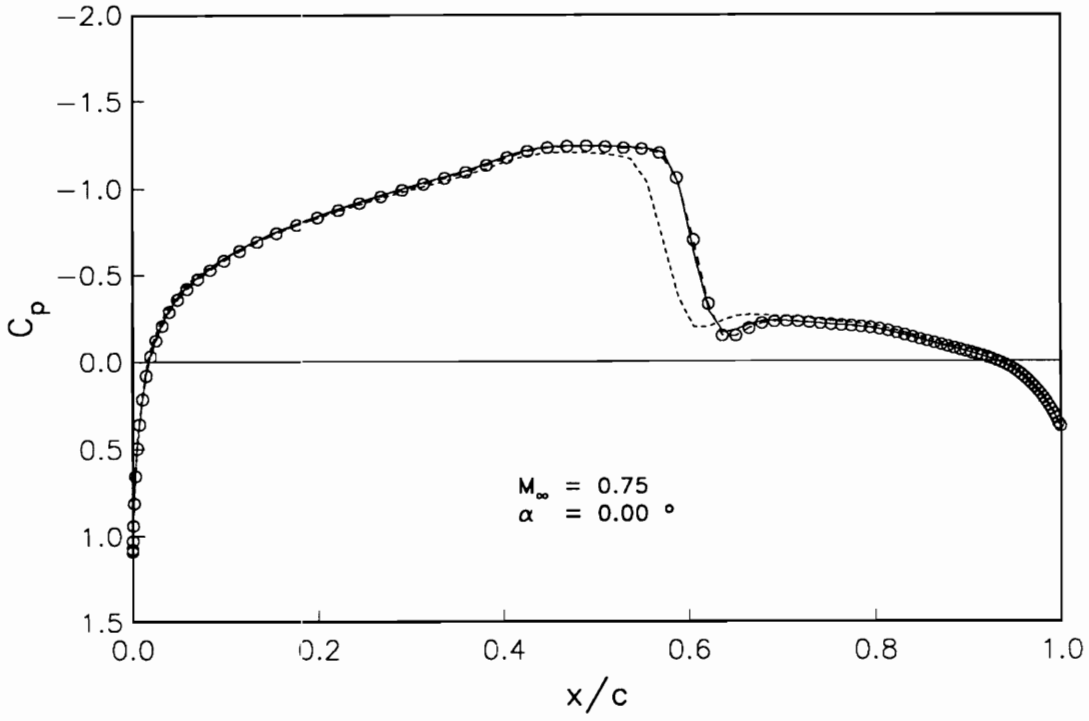



..... Base solution, X = (0.50, 0.50,−0.50, 0.50)

o−o Approximation, X = (0.51, 0.51,−0.49, 0.51)

—— Exact Analysis, X = (0.51, 0.51,−0.49, 0.51)

**Figure 7a.** Approximation of Pressure Distribution Based on the Method of Strained Coordinates – TSD.




..... Base solution, X = (0.50, 0.50,−0.50, 0.50)

o−o Approximation, X = (0.51, 0.51,−0.49, 0.51)

—— Exact Analysis, X = (0.51, 0.51,−0.49, 0.51)

**Figure 7b.** Approximation of Pressure Distribution Based on the Method of Strained Coordinates – Euler.

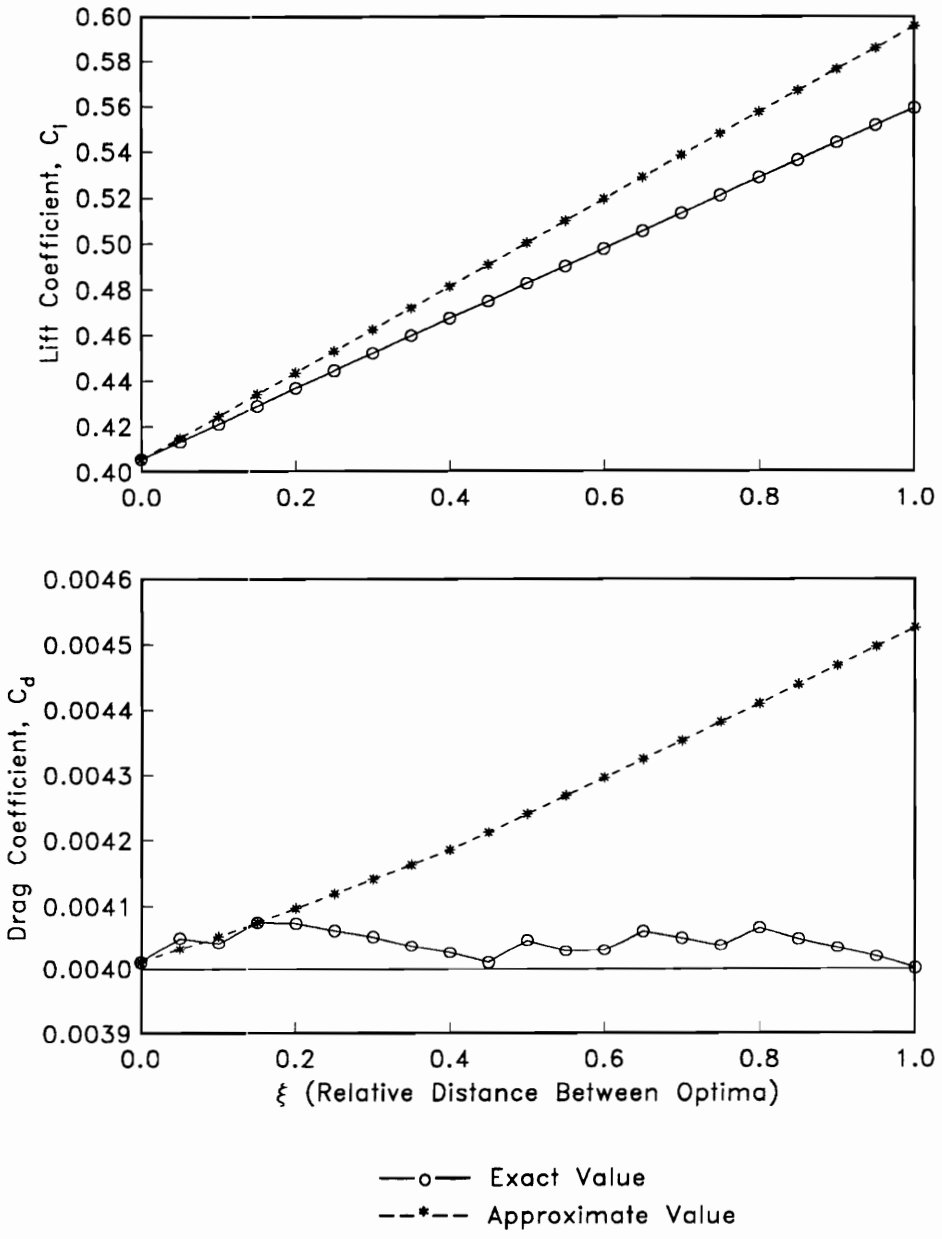



**Figure 8.** Objective Function and Constraint Behavior for Two TSD Local Optima – Initial Interpolations.

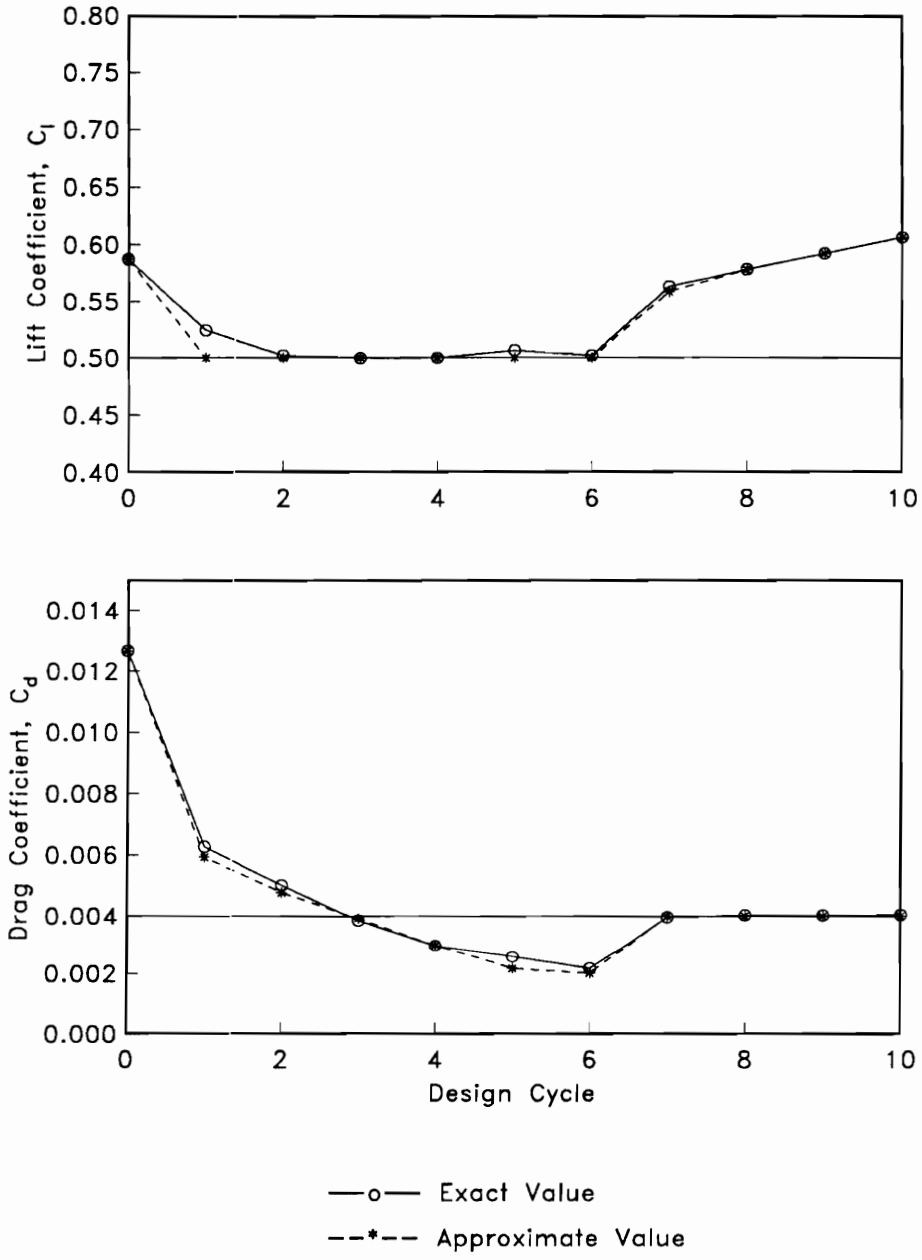



**Figure 9.** TSD Convergence History(Table 4c) : Initial Interpolations, Drag Minimization followed by Lift Maximization.

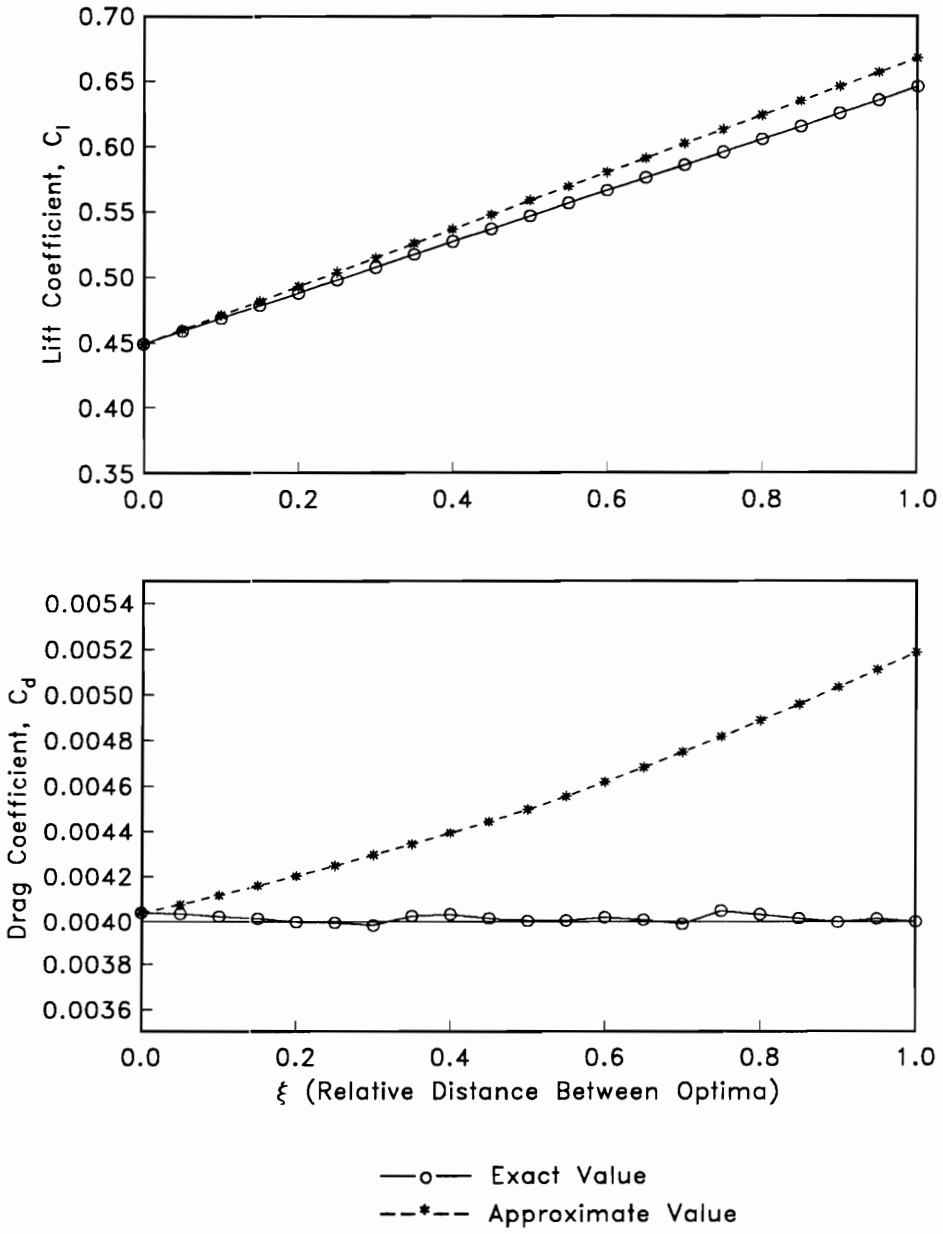



**Figure 10.** Objective Function and Constraint Behavior for Two TSD Designs – Smooth Leading Edge Geometries.

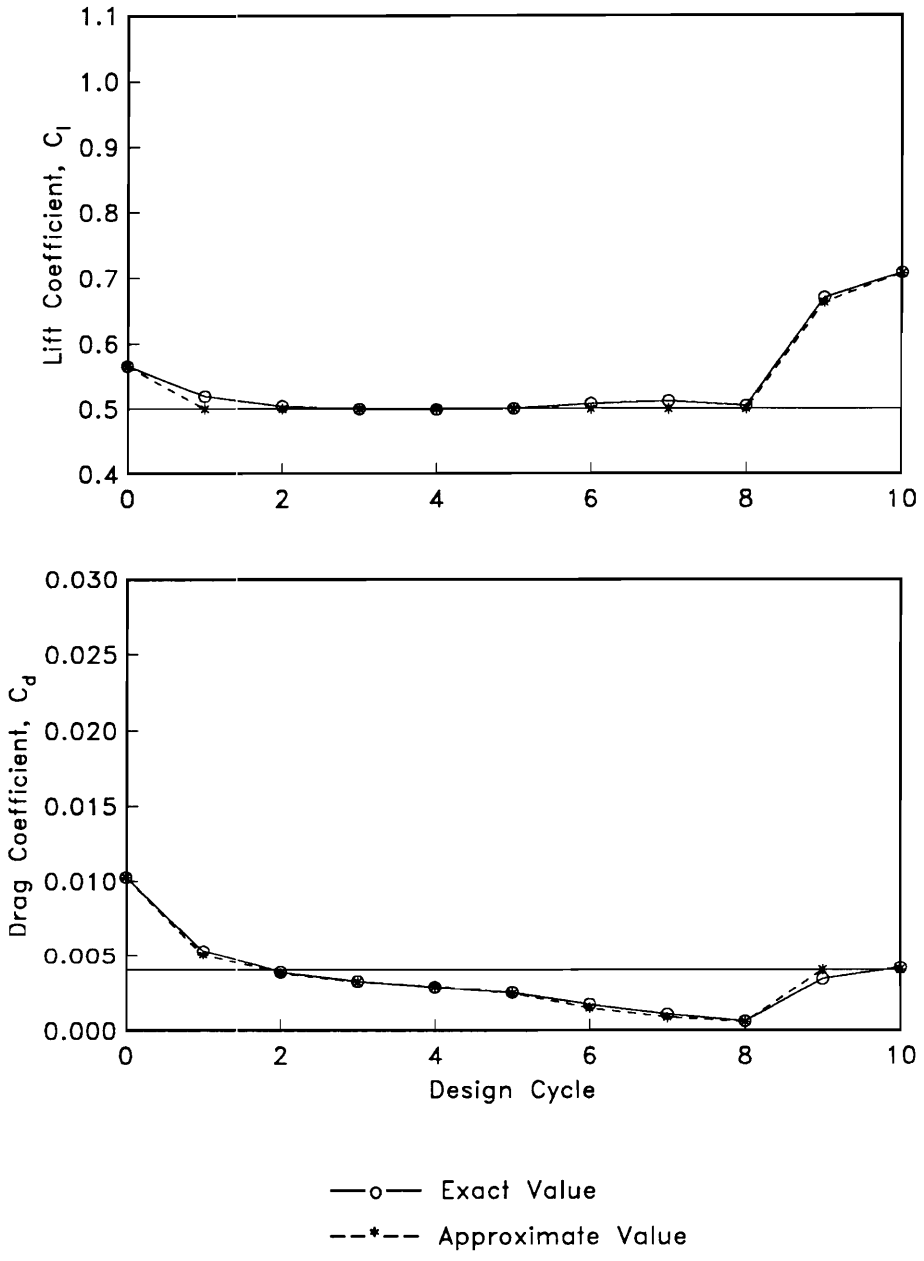



**Figure 11a.** TSD Convergence History(Table 6c) : Strategy B, Drag Minimization followed by Lift Maximization.

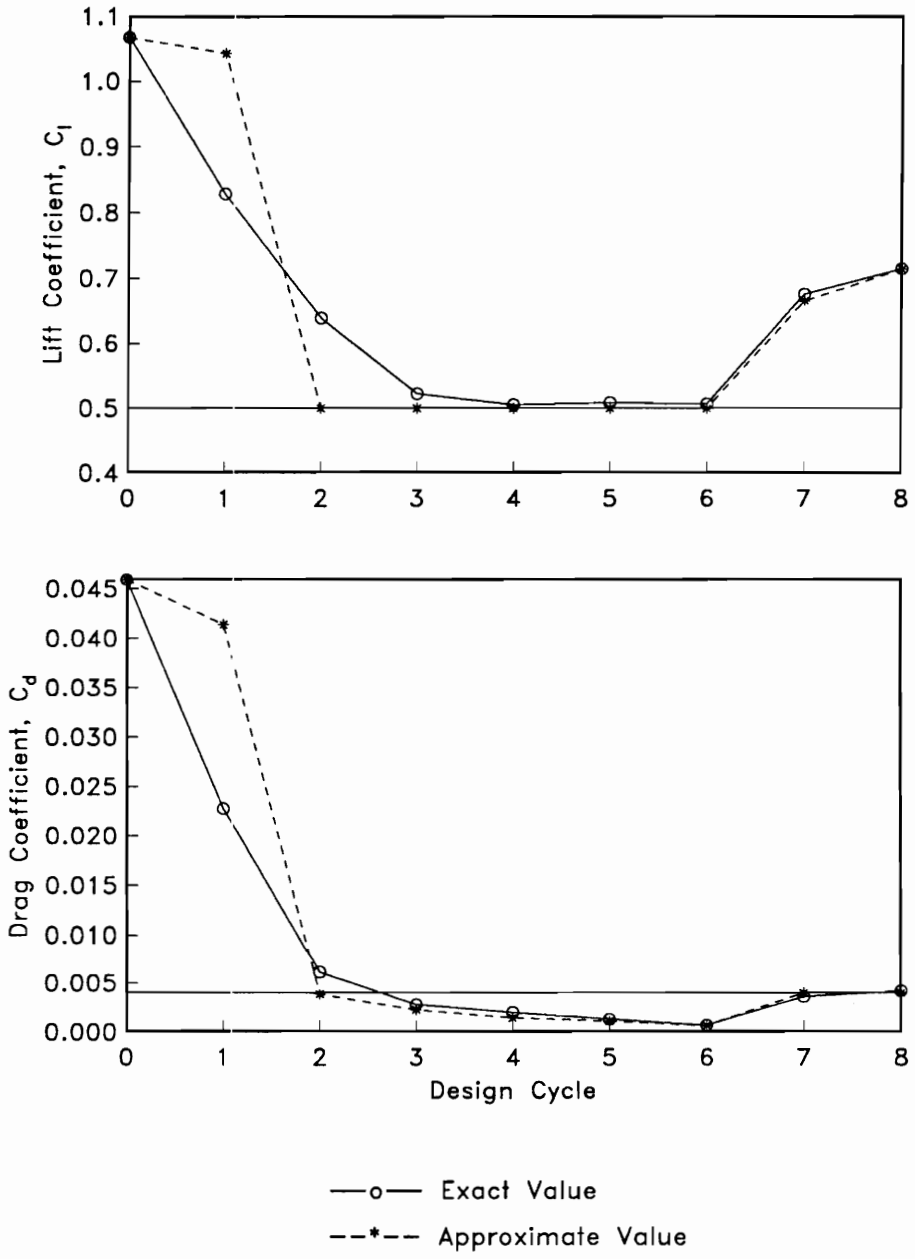



**Figure 11b.** TSD Convergence History(Table 6d) : Strategy B, Drag Minimization followed by Lift Maximization.

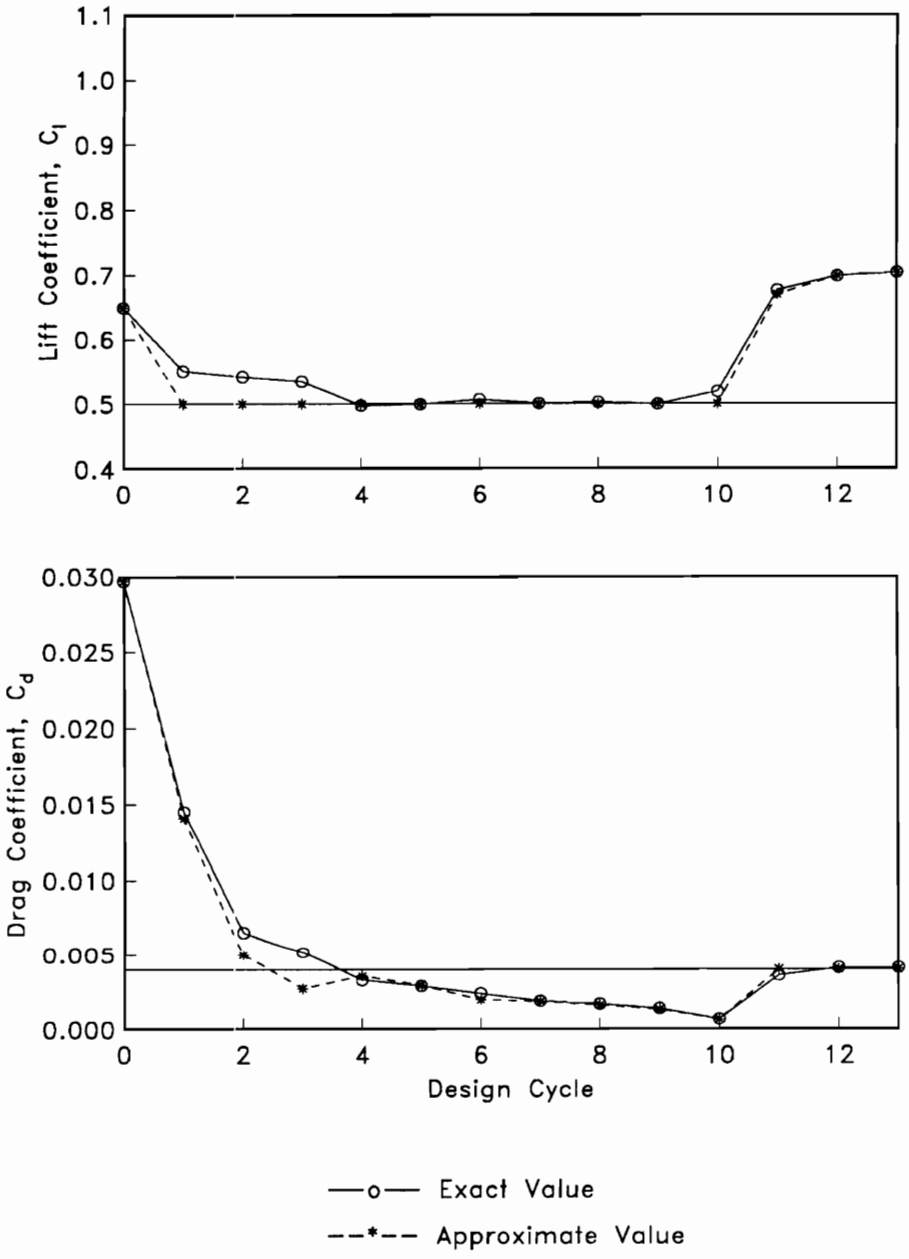



**Figure 11c.** TSD Convergence History(Table 6e) : Strategy B, Drag Minimization followed by Lift Maximization.

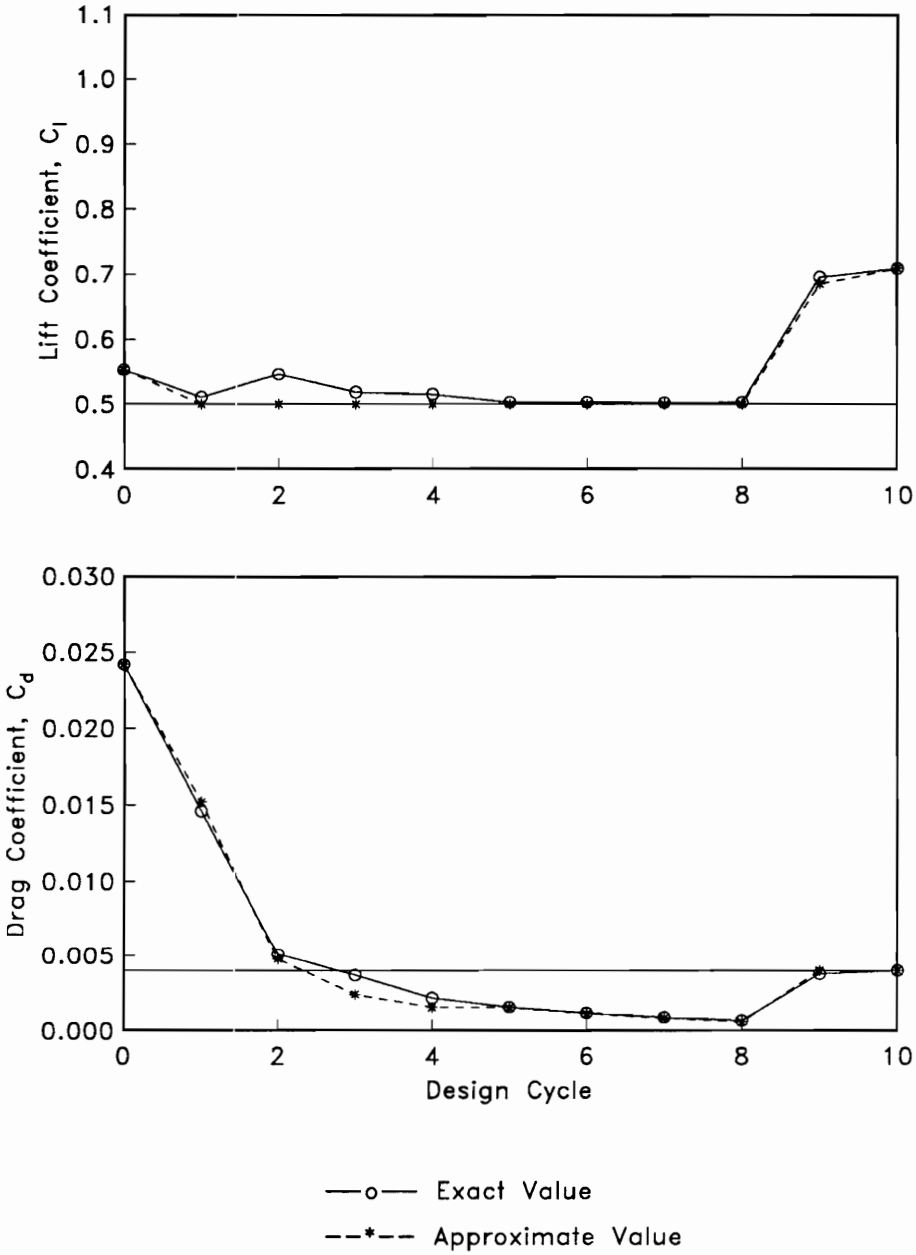



**Figure 11d.** TSD Convergence History(Table 6f) : Strategy B, Drag Minimization followed by Lift Maximization.

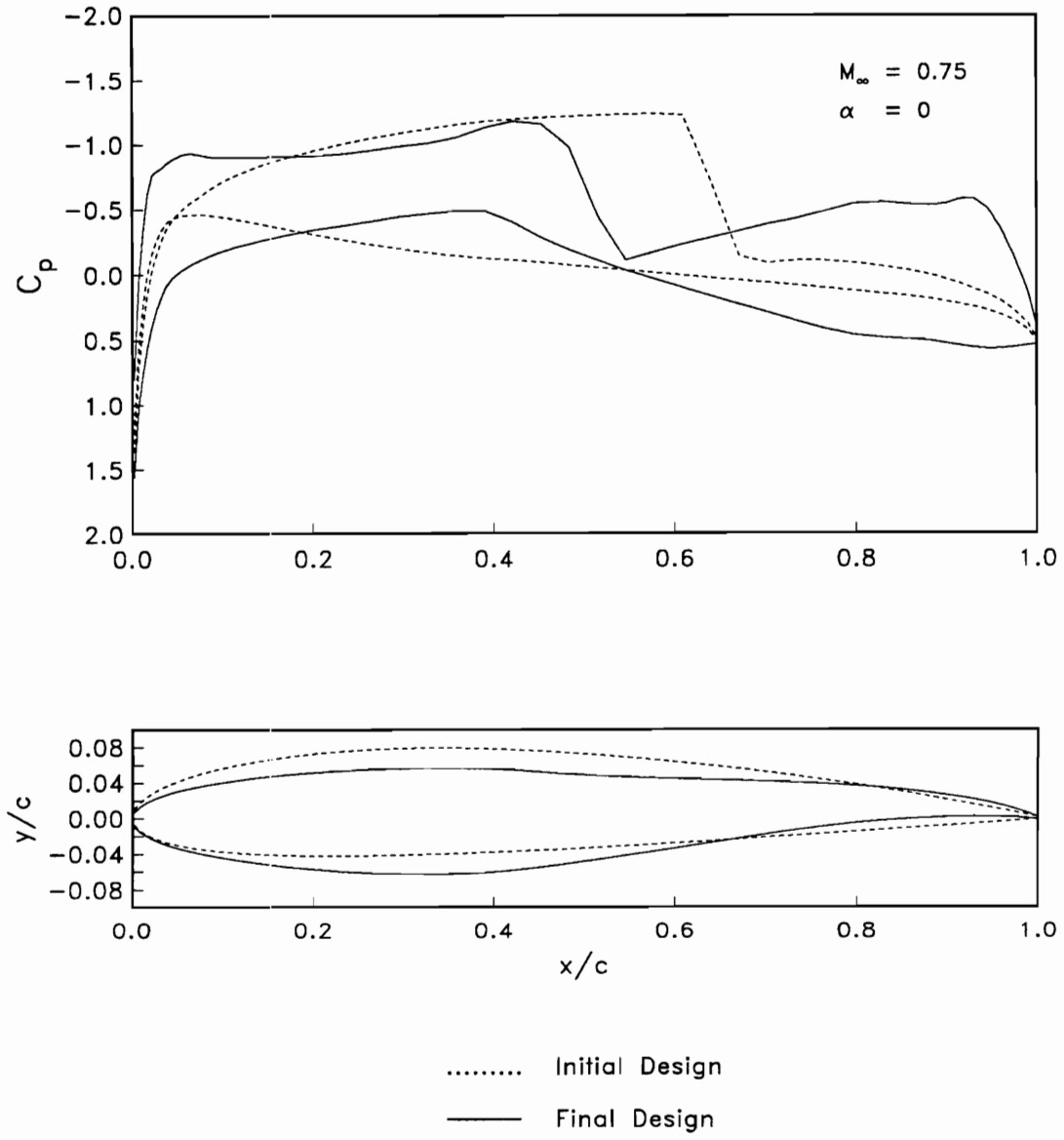



**Figure 12a.** TSD Design Results (Table 6c) : Strategy B, Drag Minimization followed by Lift maximization.

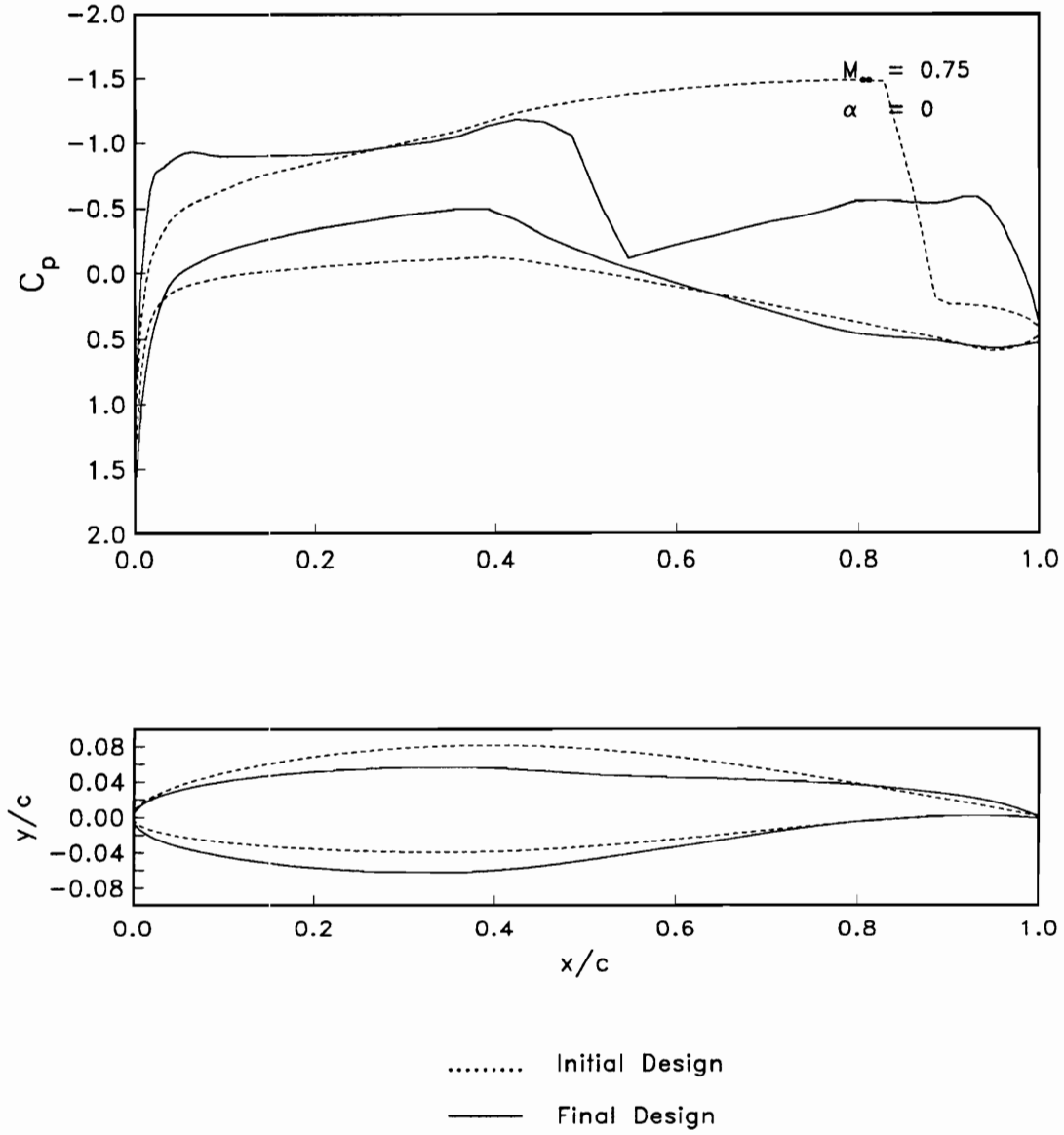



**Figure 12b.** TSD Design Results (Table 6d) : Strategy B, Drag Minimization followed by Lift maximization.

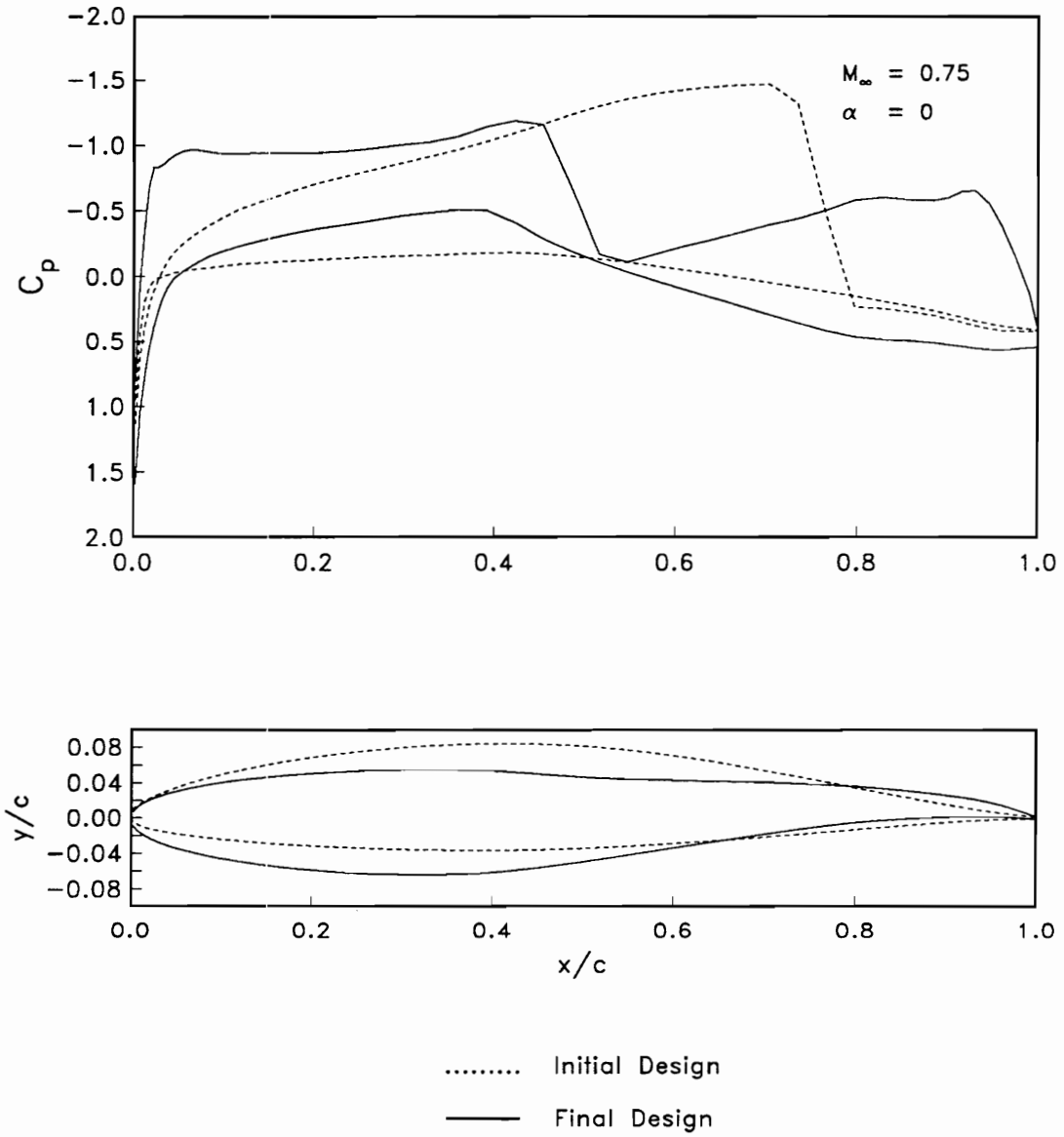



**Figure 12c.** TSD Design Results (Table 6e) : Strategy B, Drag Minimization followed by Lift maximization.

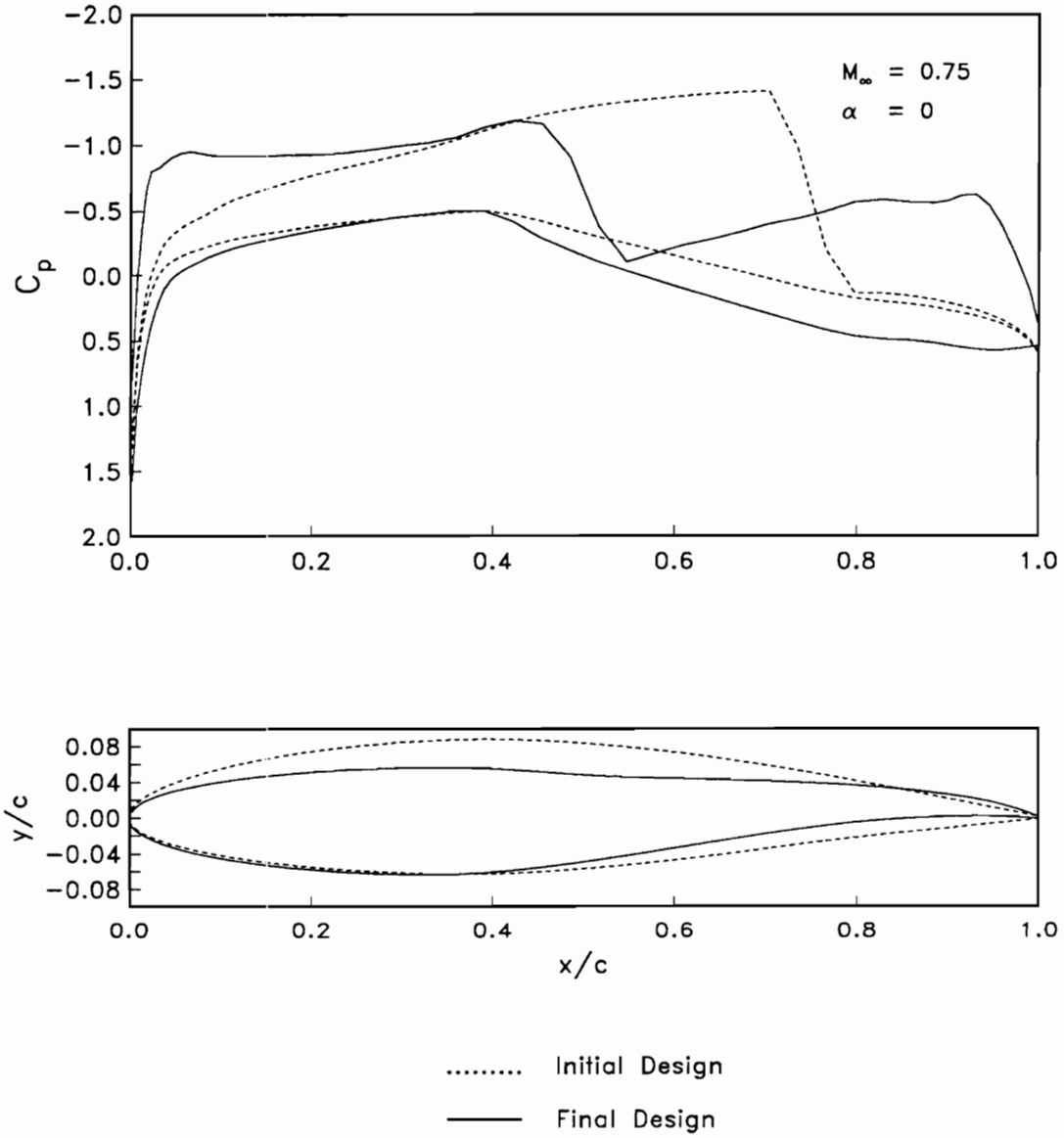



**Figure 12d.** TSD Design Results (Table 6f) : Strategy B, Drag Minimization followed by Lift maximization.

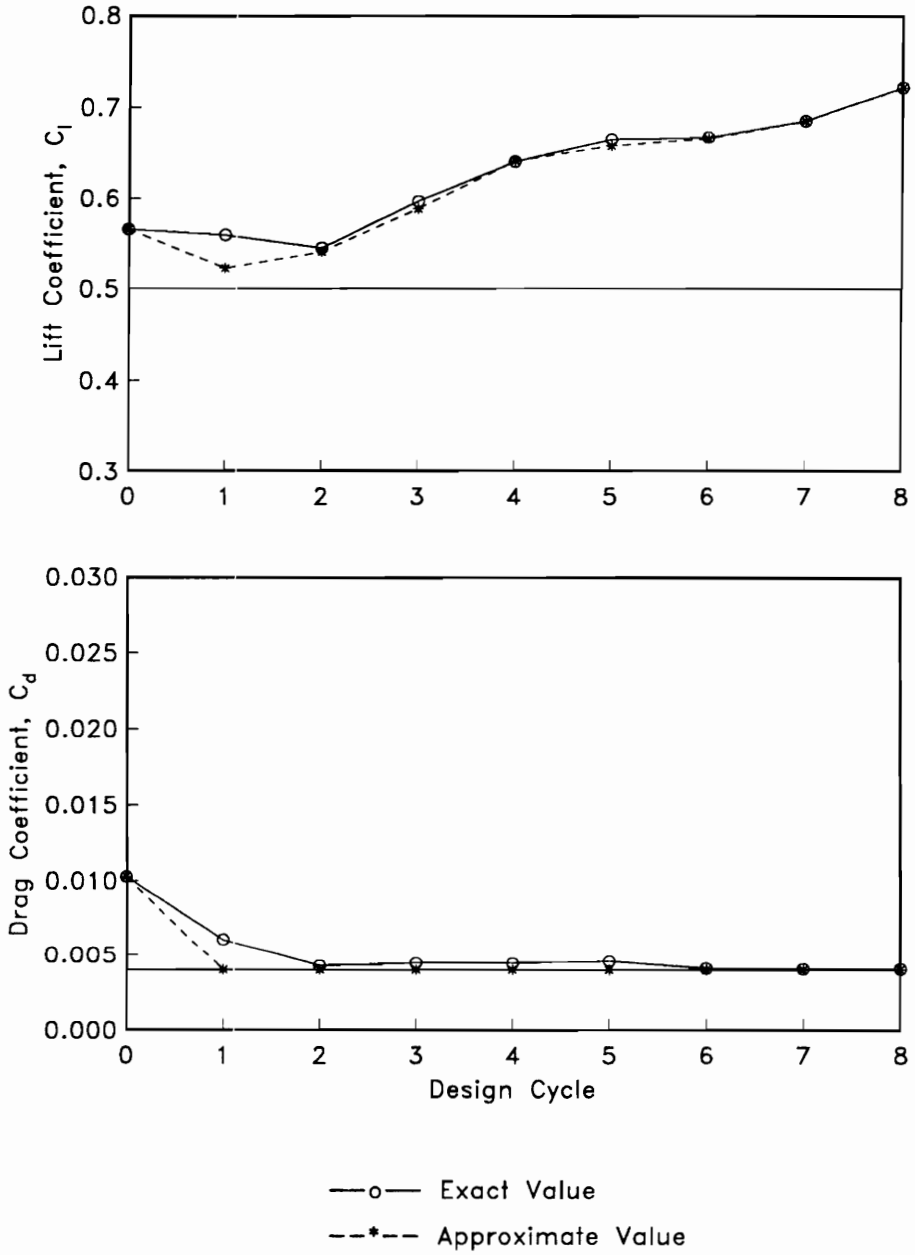



**Figure 13a.** TSD Convergence History (Table 7a) : Strategy C, Approximate Optimization with Absolute Move Limits.

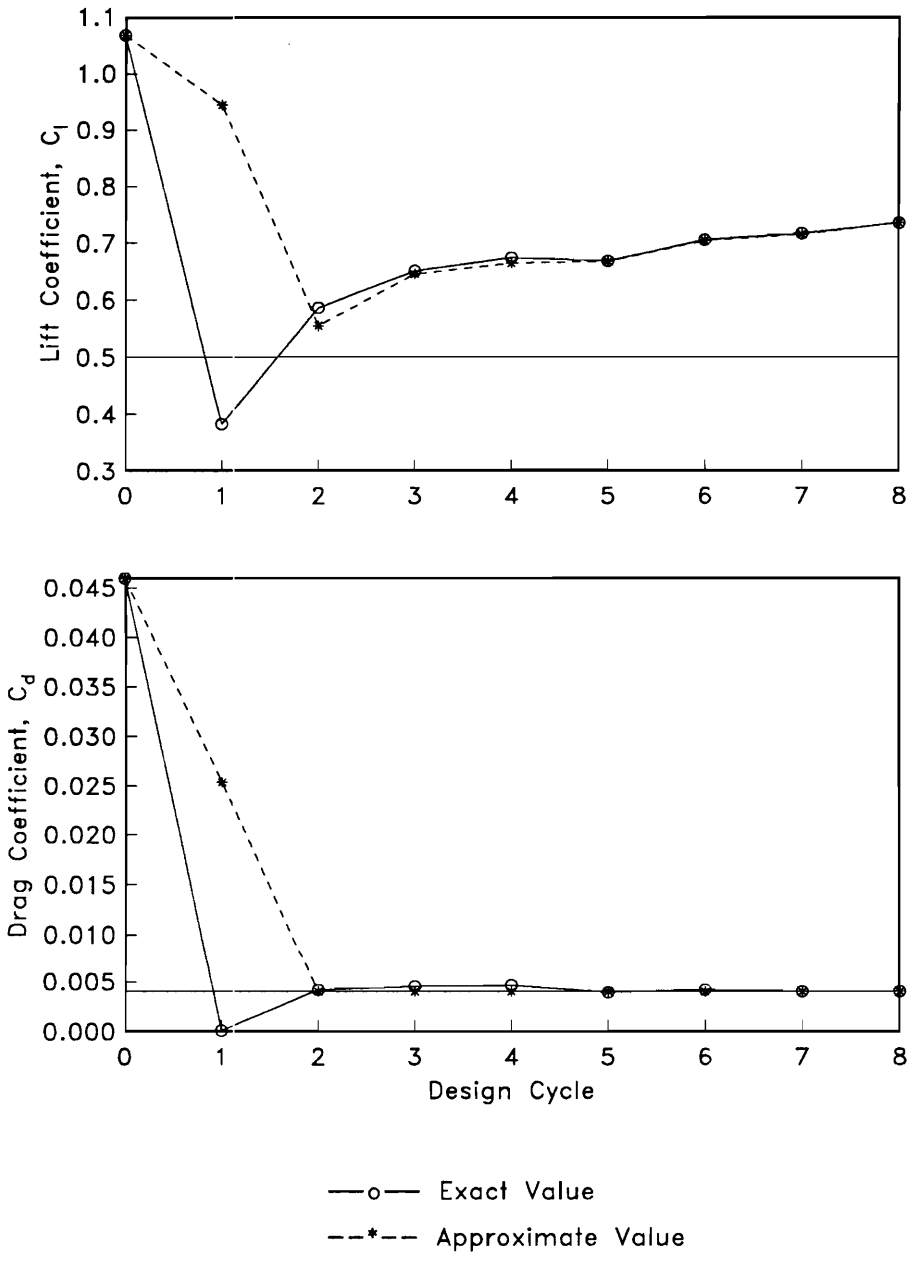



**Figure 13b.** TSD Convergence History (Table 7b) : Strategy C, Approximate Optimization with Absolute Move Limits.

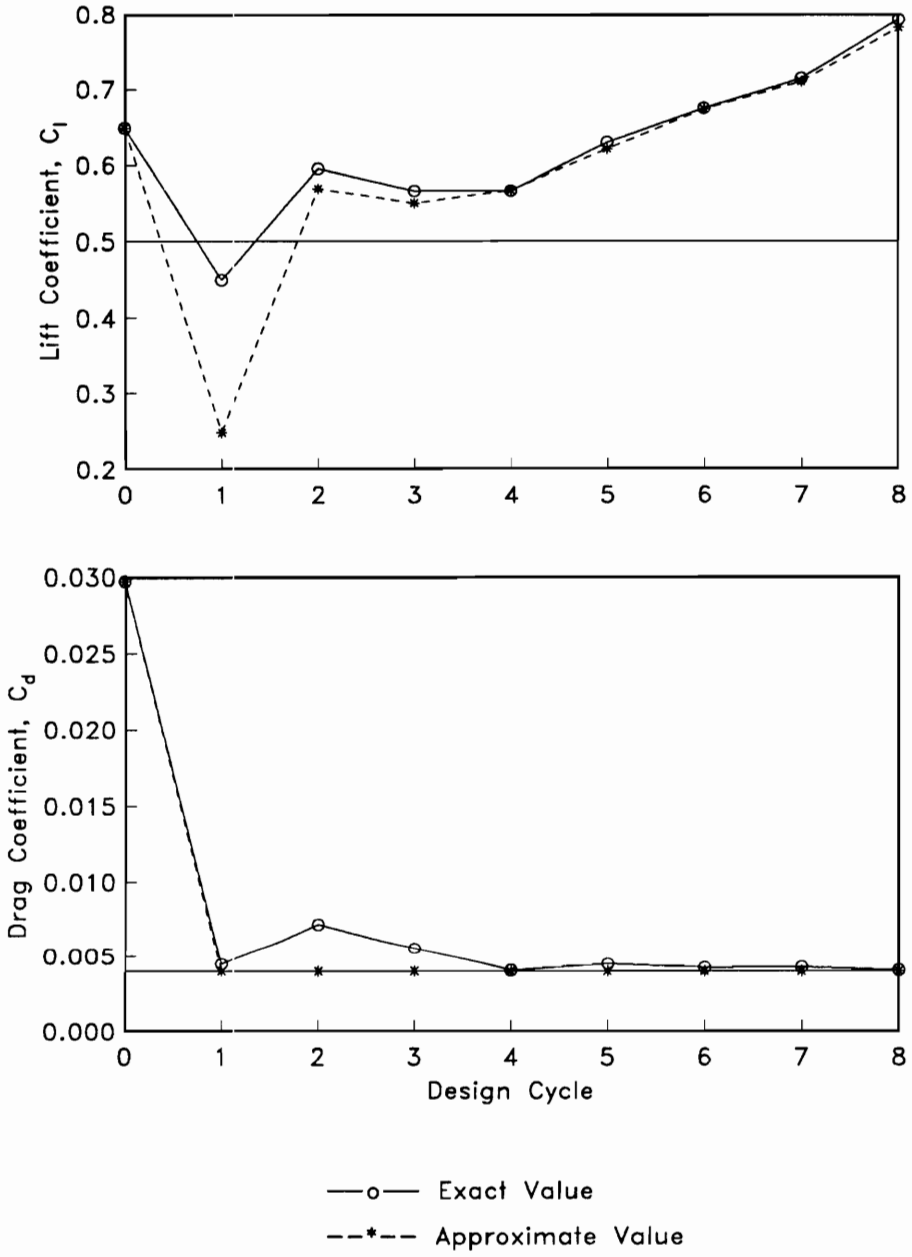



**Figure 13c.** TSD Convergence History (Table 7c) : Strategy C, Approximate Optimization with Absolute Move Limits.

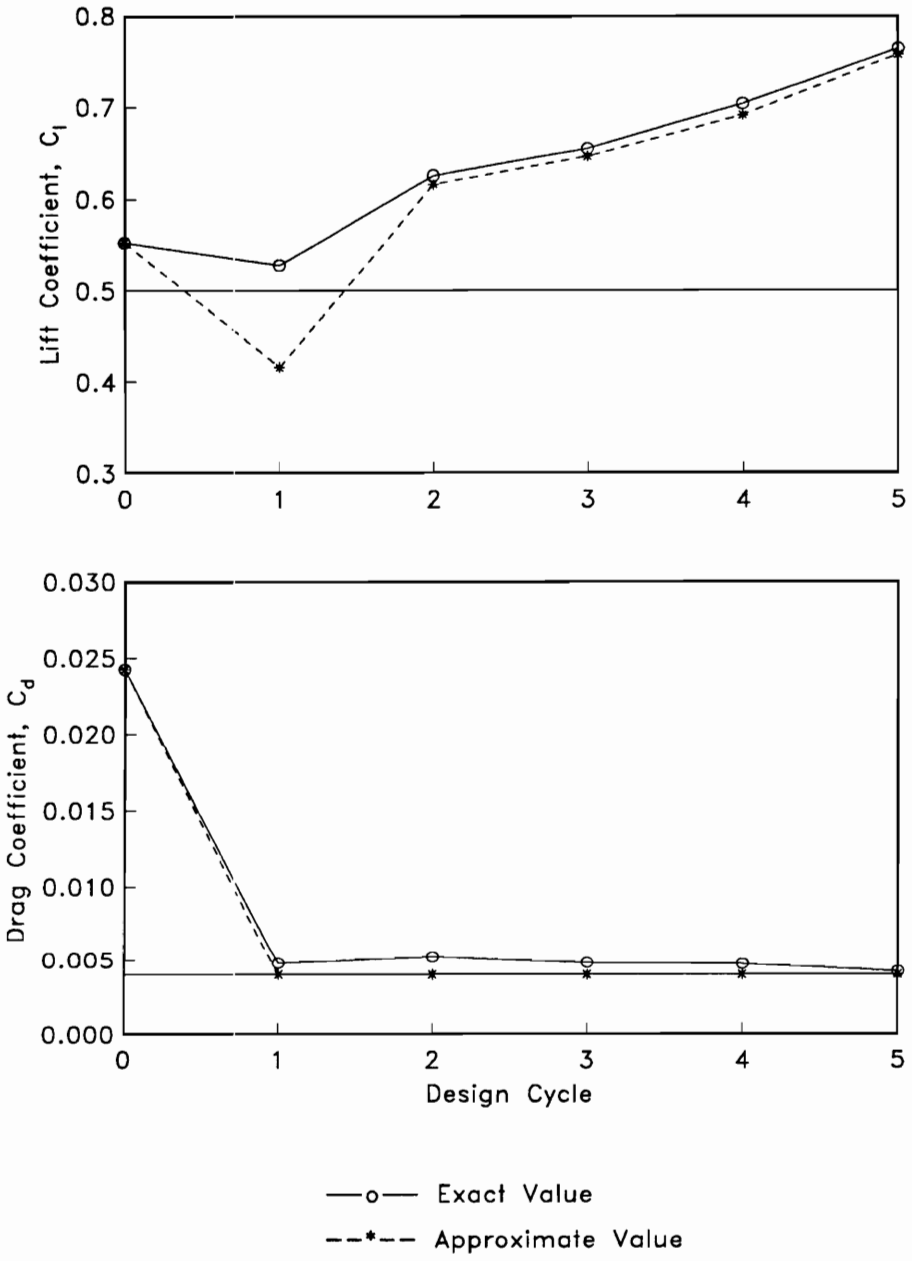



**Figure 13d.** TSD Convergence History (Table 7d) : Strategy C, Approximate Optimization with Absolute Move Limits.

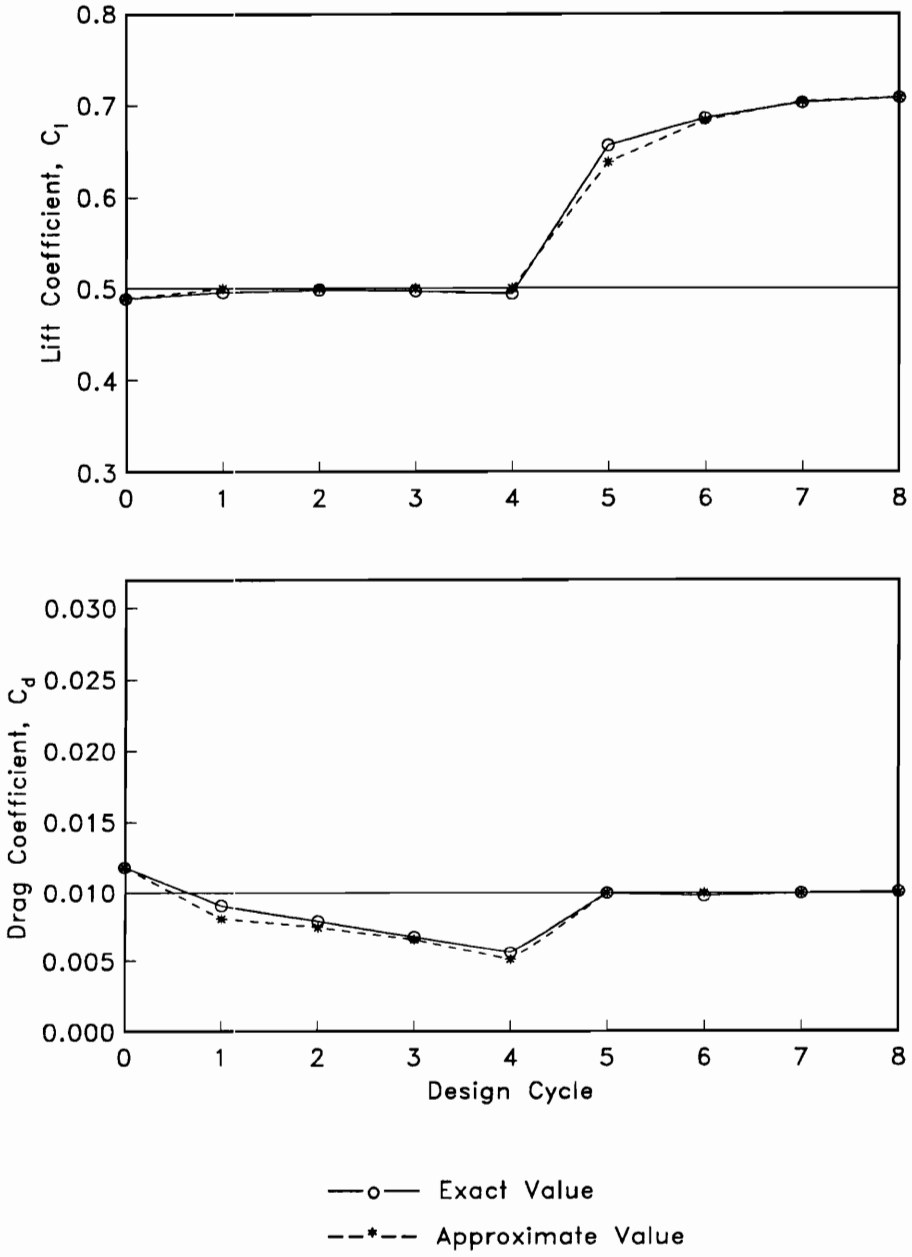



**Figure 14a.** Euler Convergence History(Table 9a) : Strategy B, Drag Minimization followed by Lift Maximization.

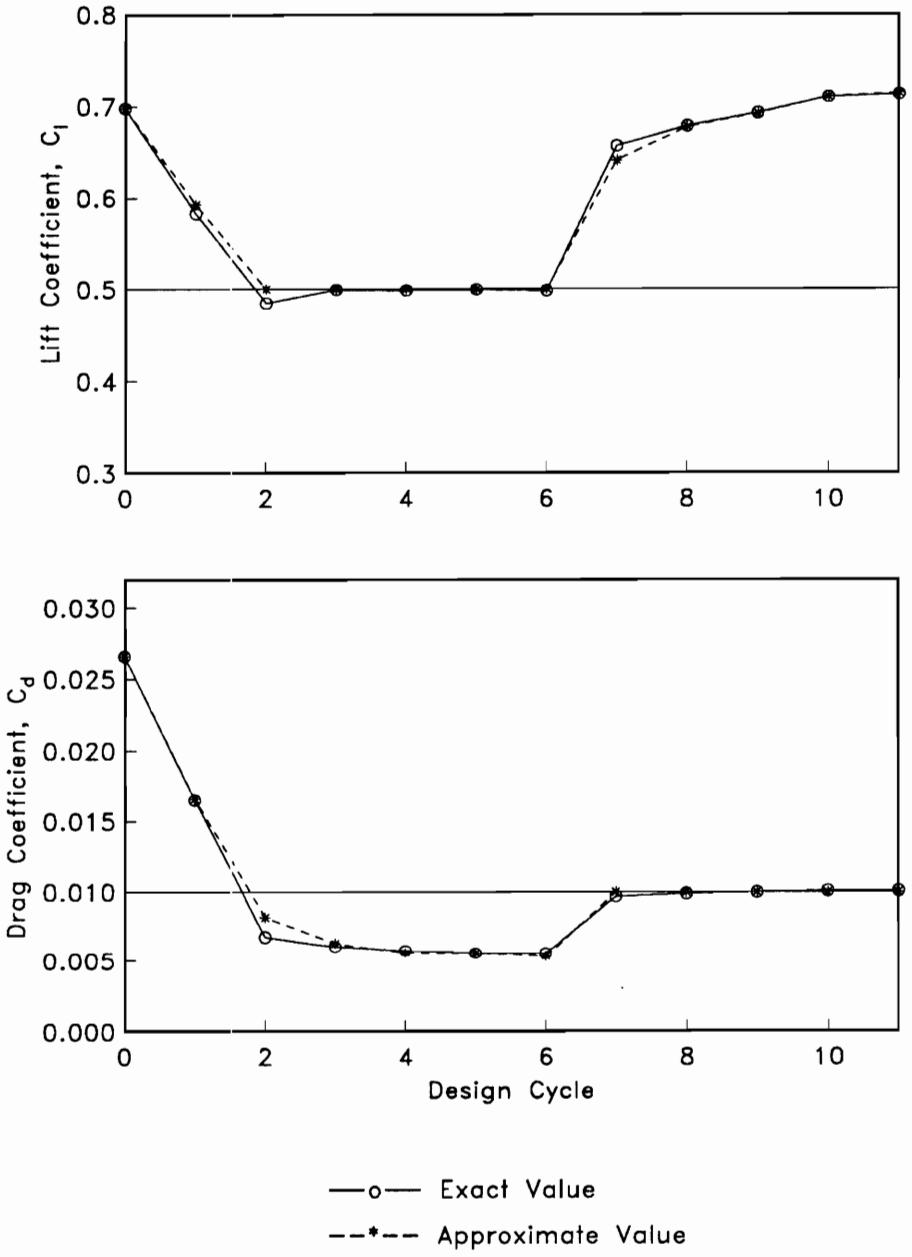



**Figure 14b.** Euler Convergence History(Table 9b) : Strategy B, Drag Minimization followed by Lift Maximization.

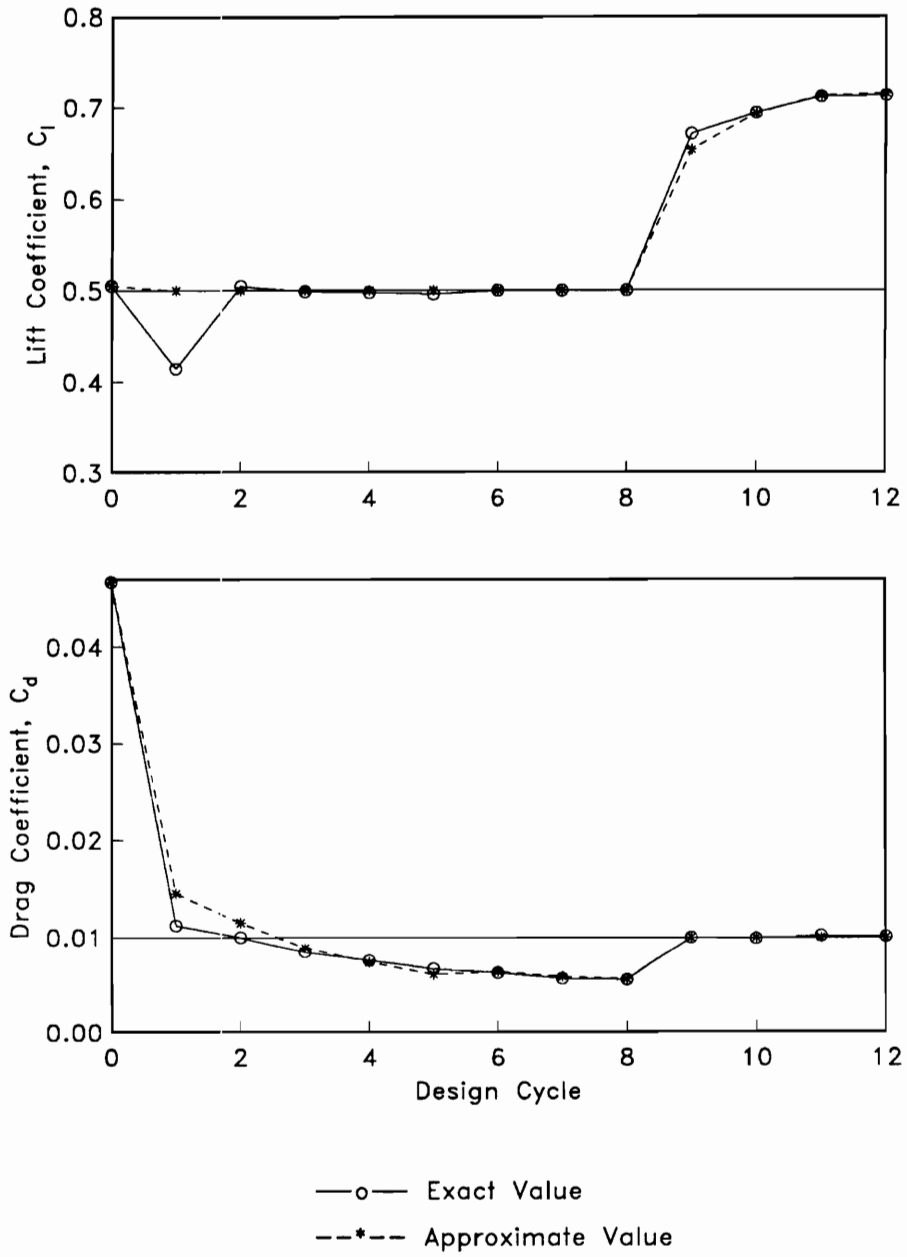



**Figure 14c.** Euler Convergence History(Table 9c) : Strategy B, Drag Minimization followed by Lift Maximization.

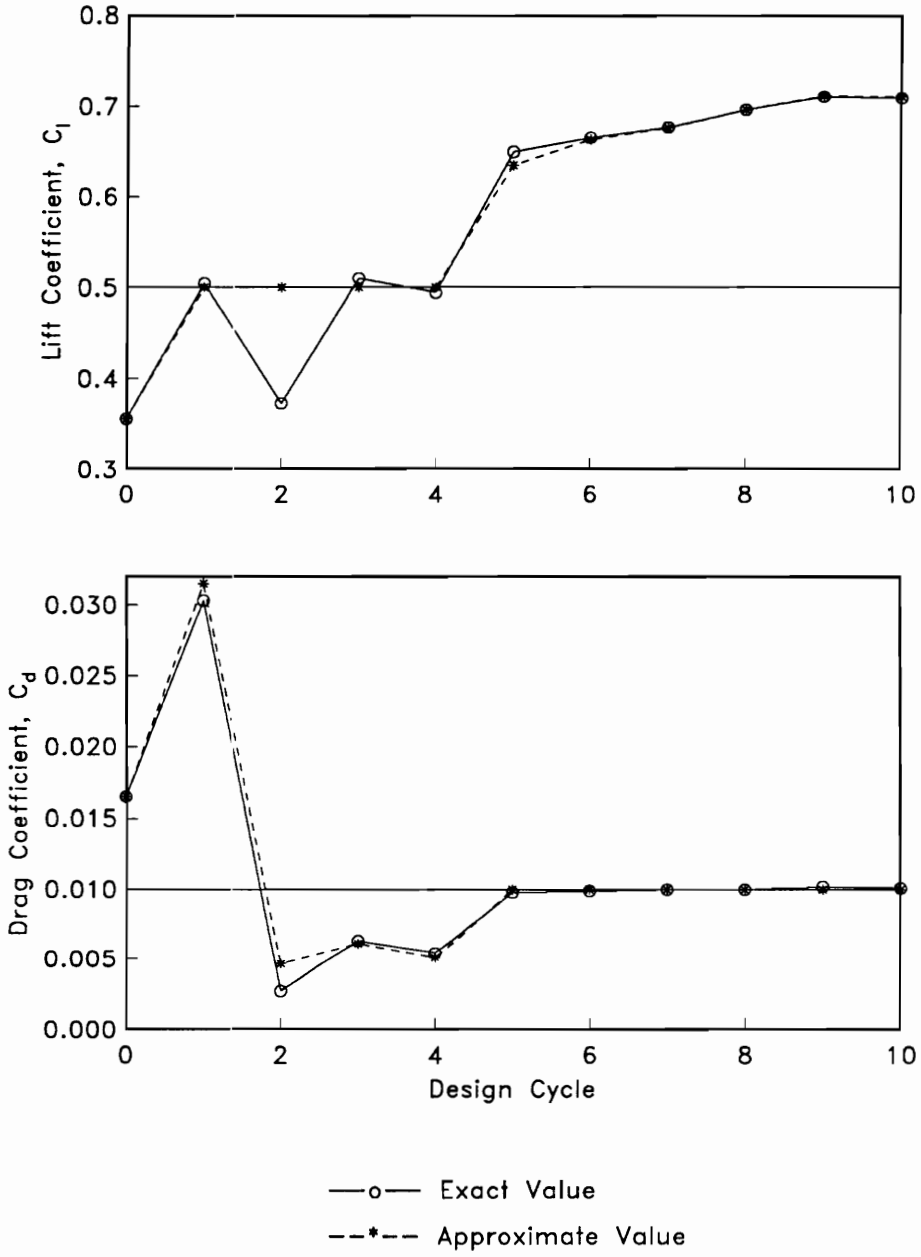



**Figure 14d.** Euler Convergence History(Table 9d) : Strategy B, Drag Minimization followed by Lift Maximization.

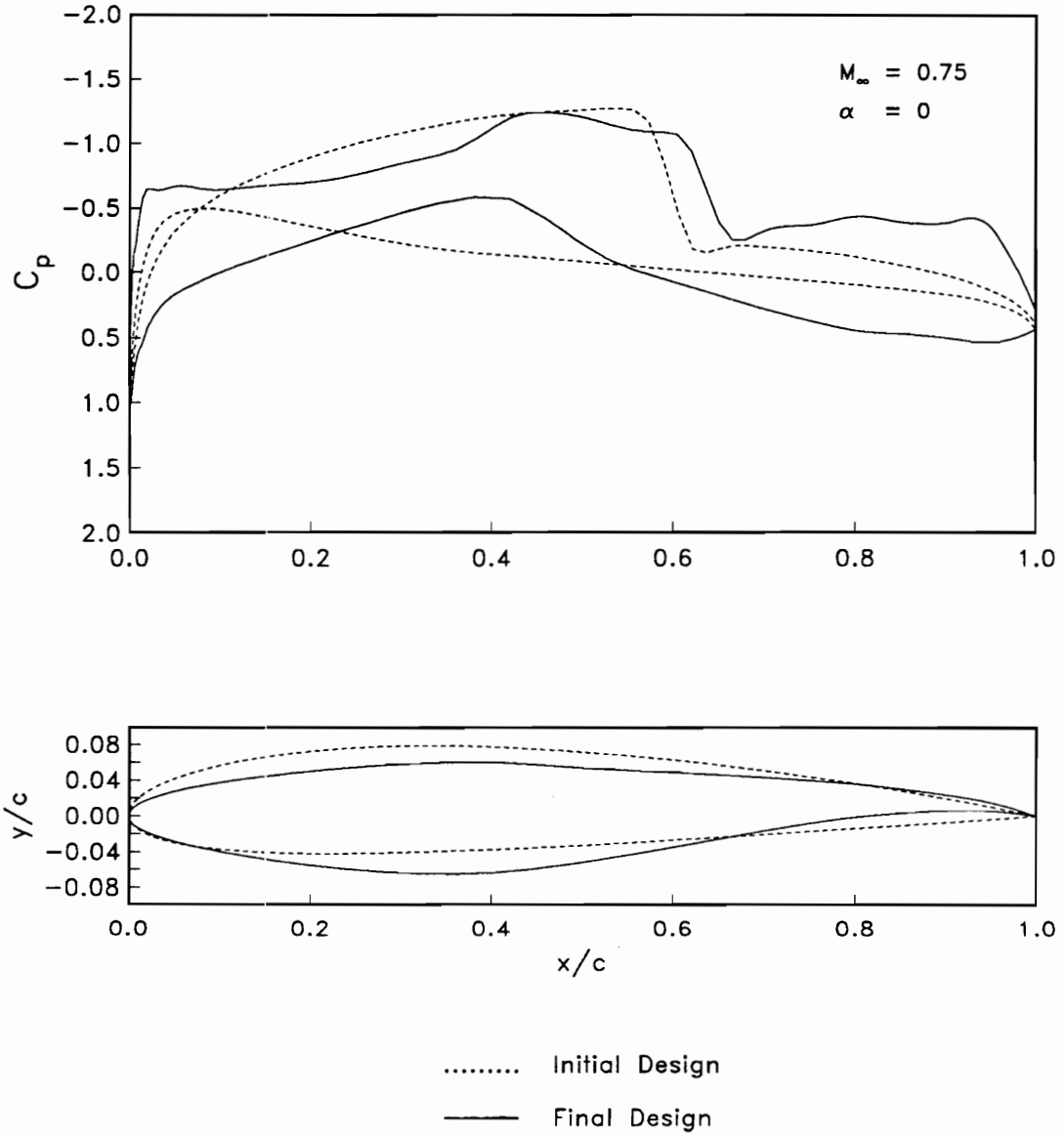



**Figure 15a.** Euler Design Results (Table 9a) : Strategy B, Drag Minimization followed by Lift maximization.

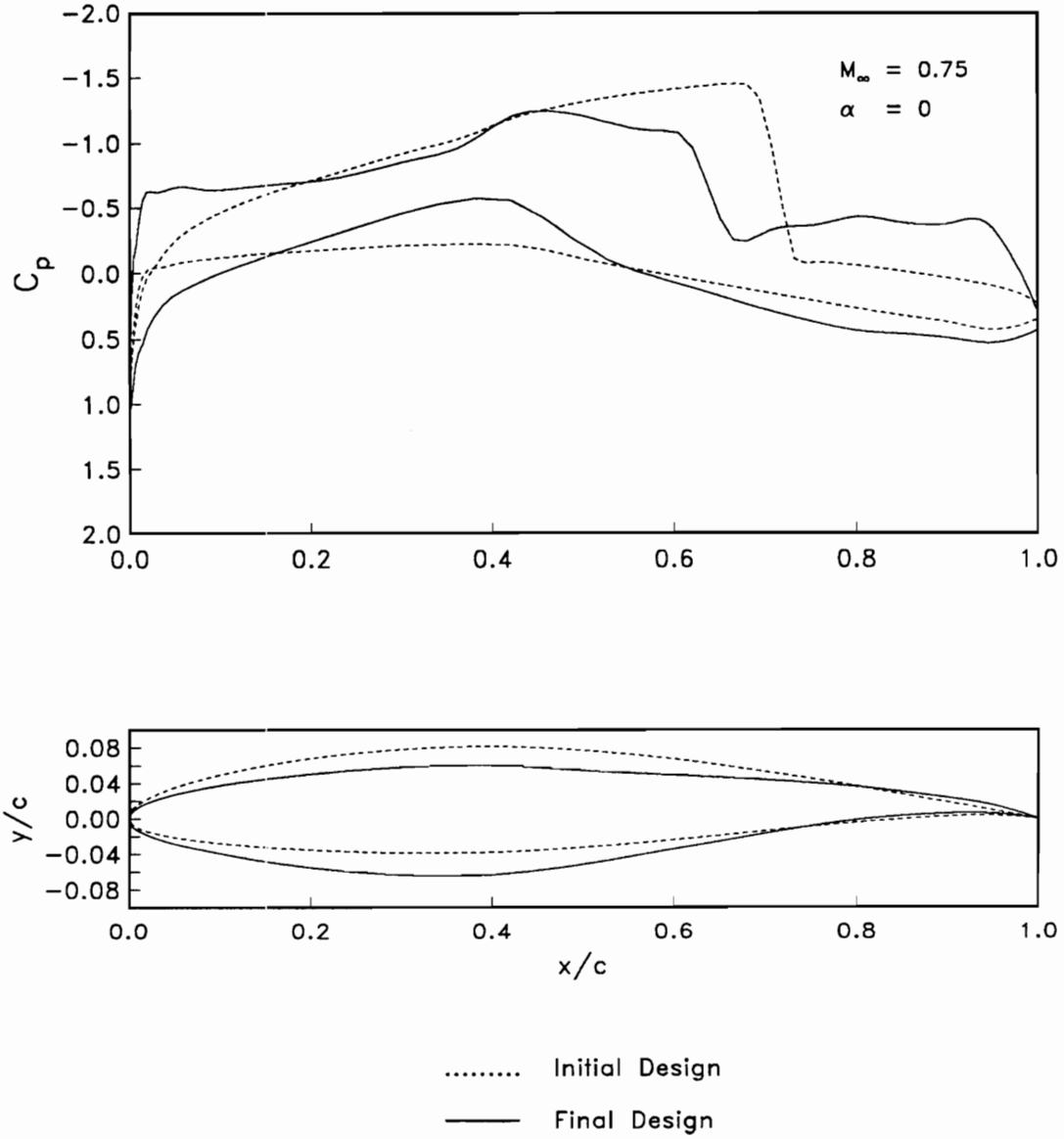



**Figure 15b.** Euler Design Results (Table 9b) : Strategy B, Drag Minimization followed by Lift maximization.

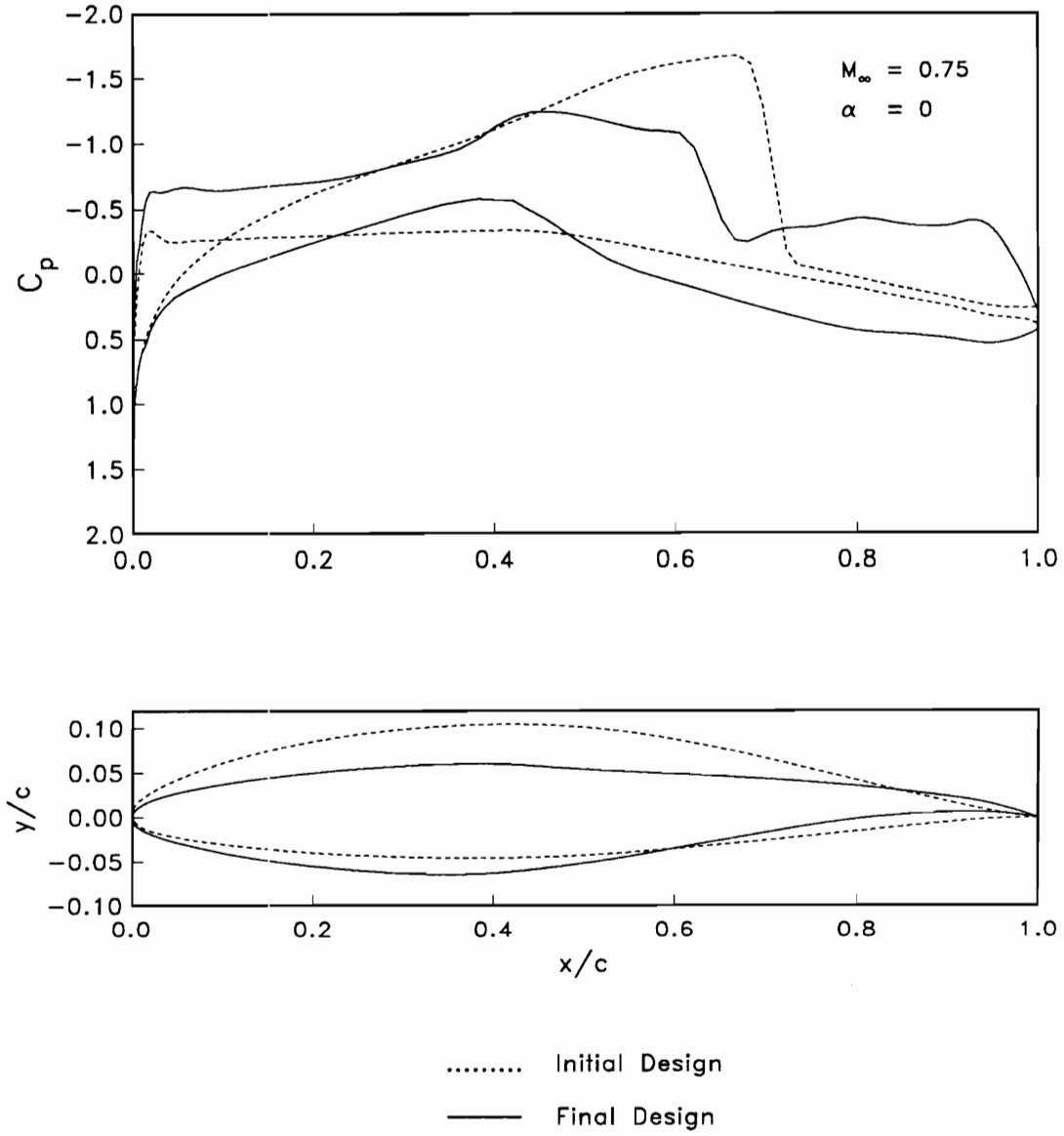



**Figure 15c.** Euler Design Results (Table 9c) : Strategy B, Drag Minimization followed by Lift maximization.

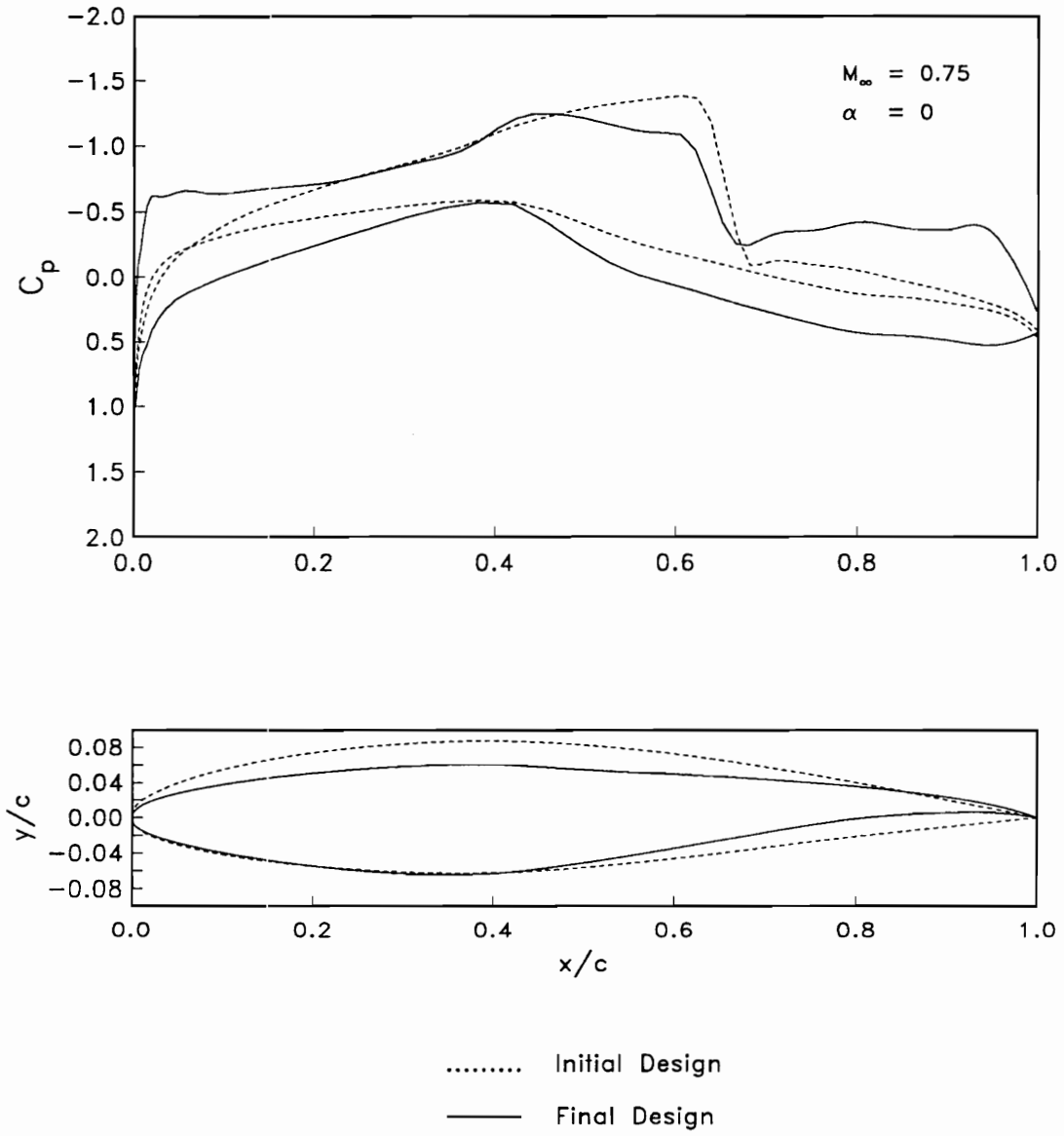



**Figure 15d.** Euler Design Results (Table 9d) : Strategy B, Drag Minimization followed by Lift maximization.

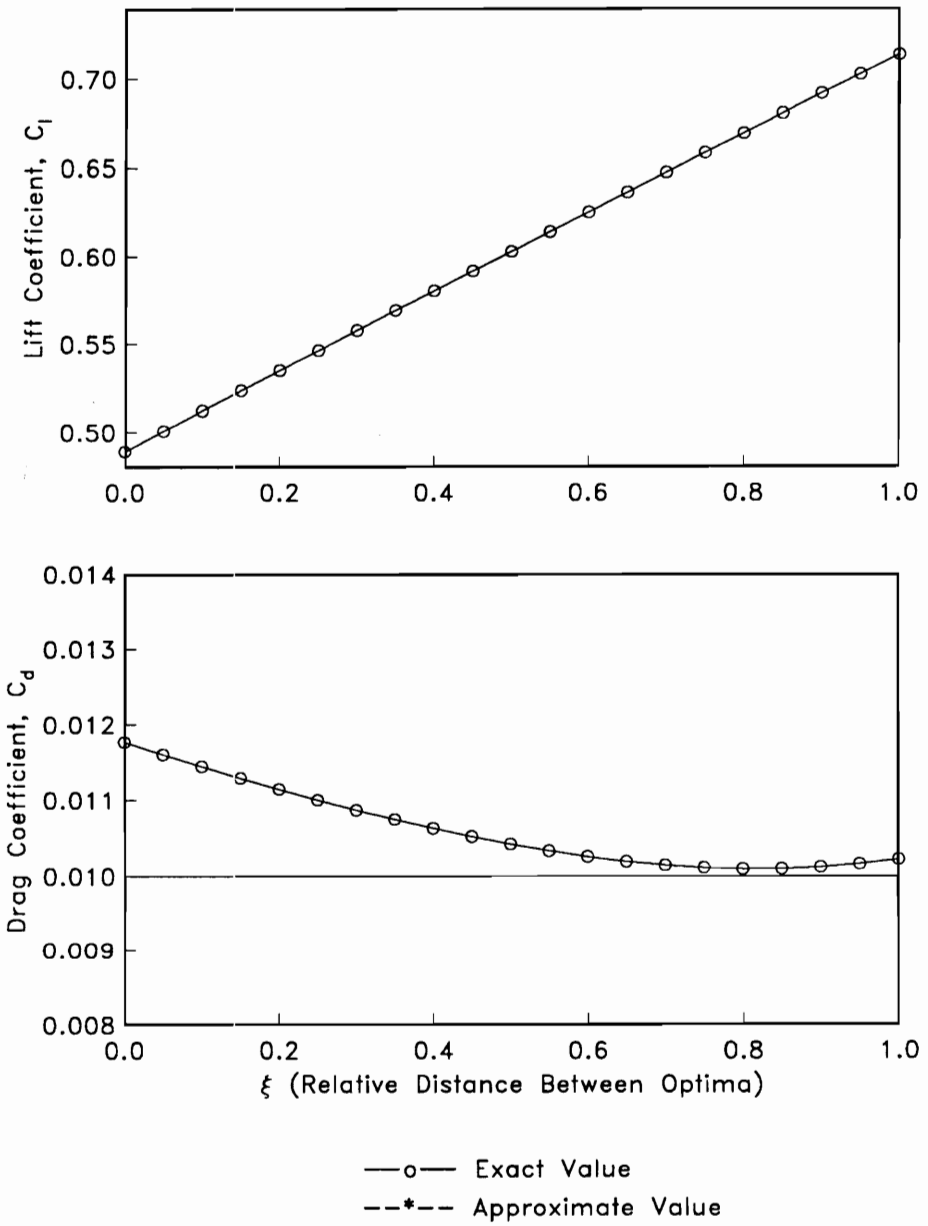



**Figure 16.** Objective Function and Constraint Behavior for Two Euler Designs(Initial and 9th Designs in Table 9a).

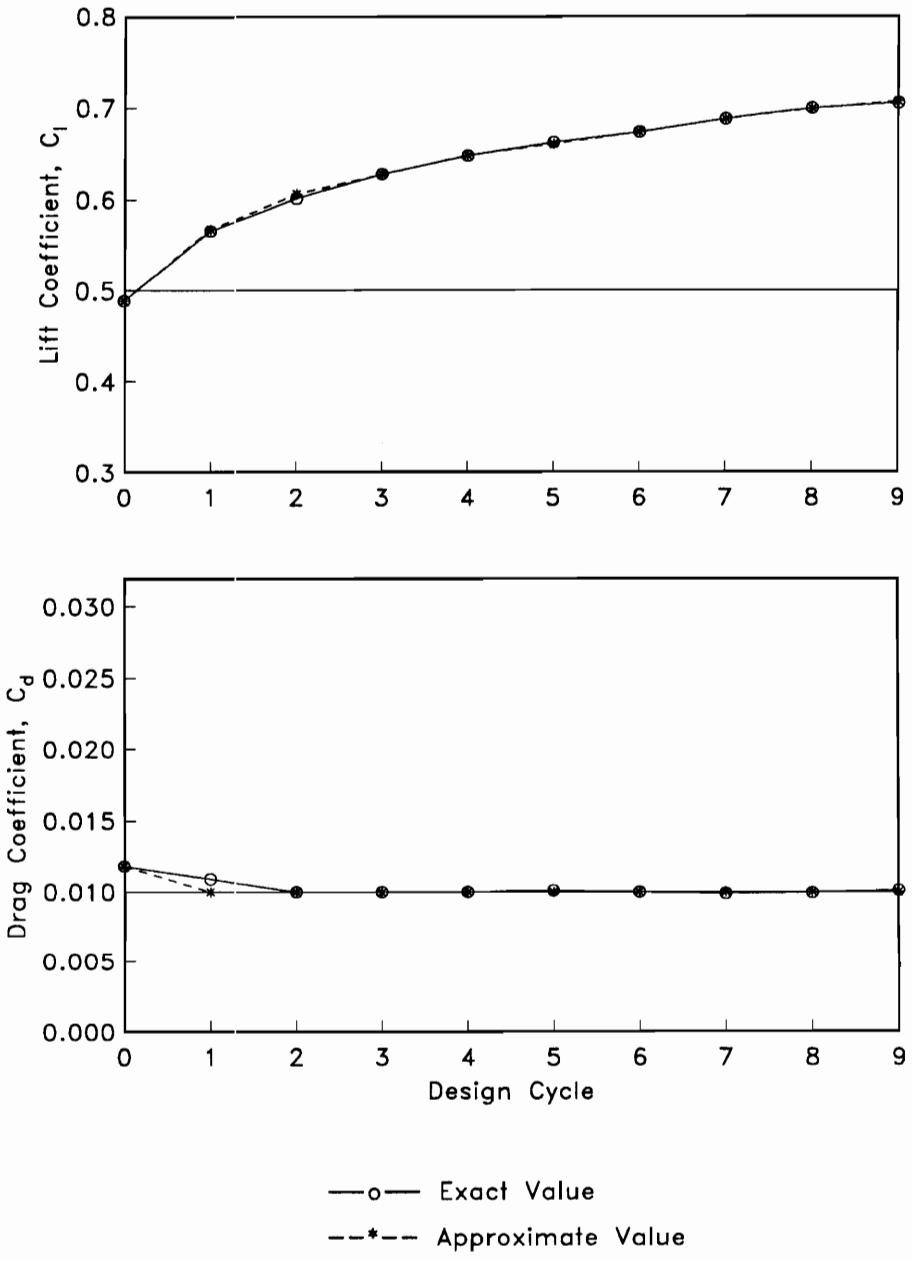



**Figure 17a.** Euler Convergence History (Table 10a) : Strategy C, Approximate Optimization with Absolute Move Limits.

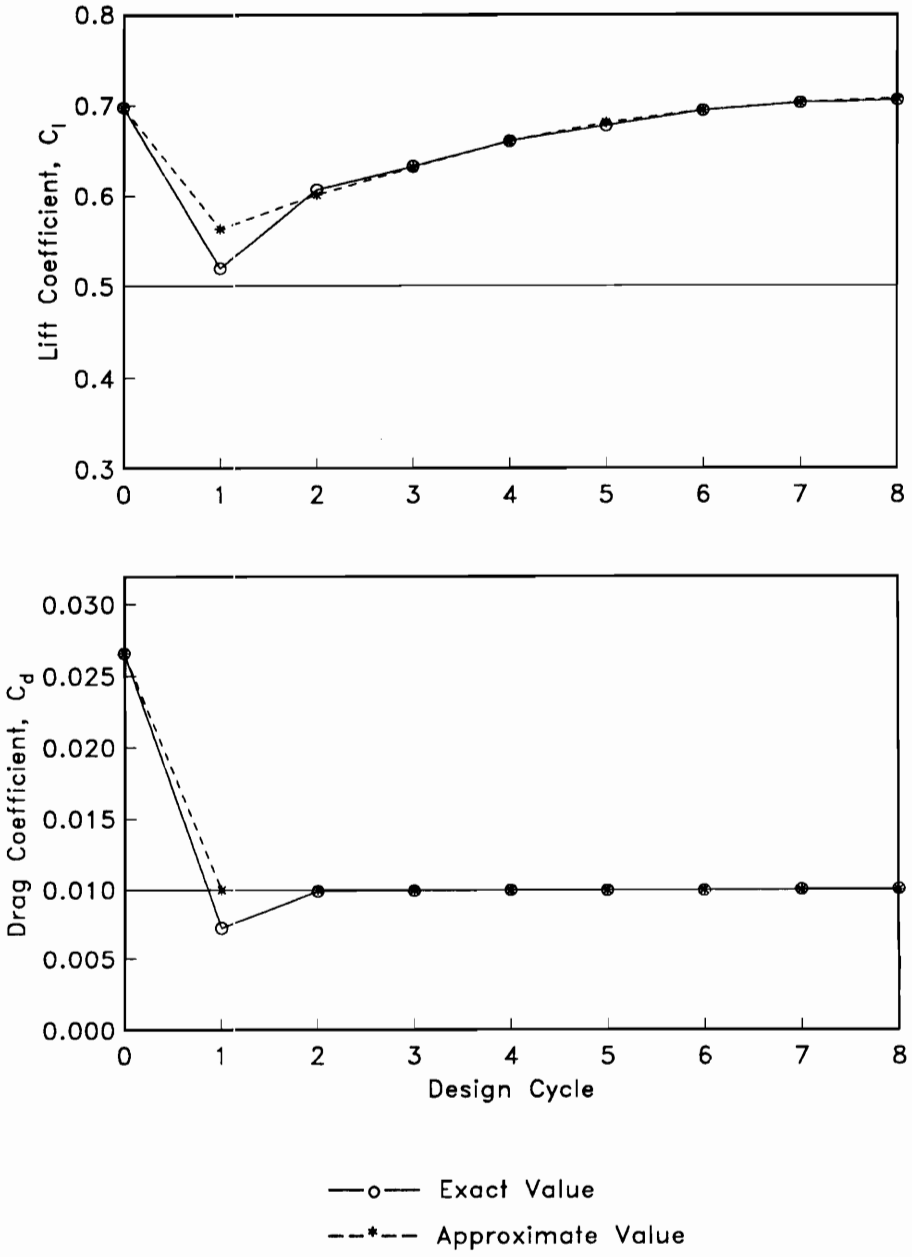



**Figure 17b.** Euler Convergence History (Table 10b) : Strategy C, Approximate Optimization with Absolute Move Limits.

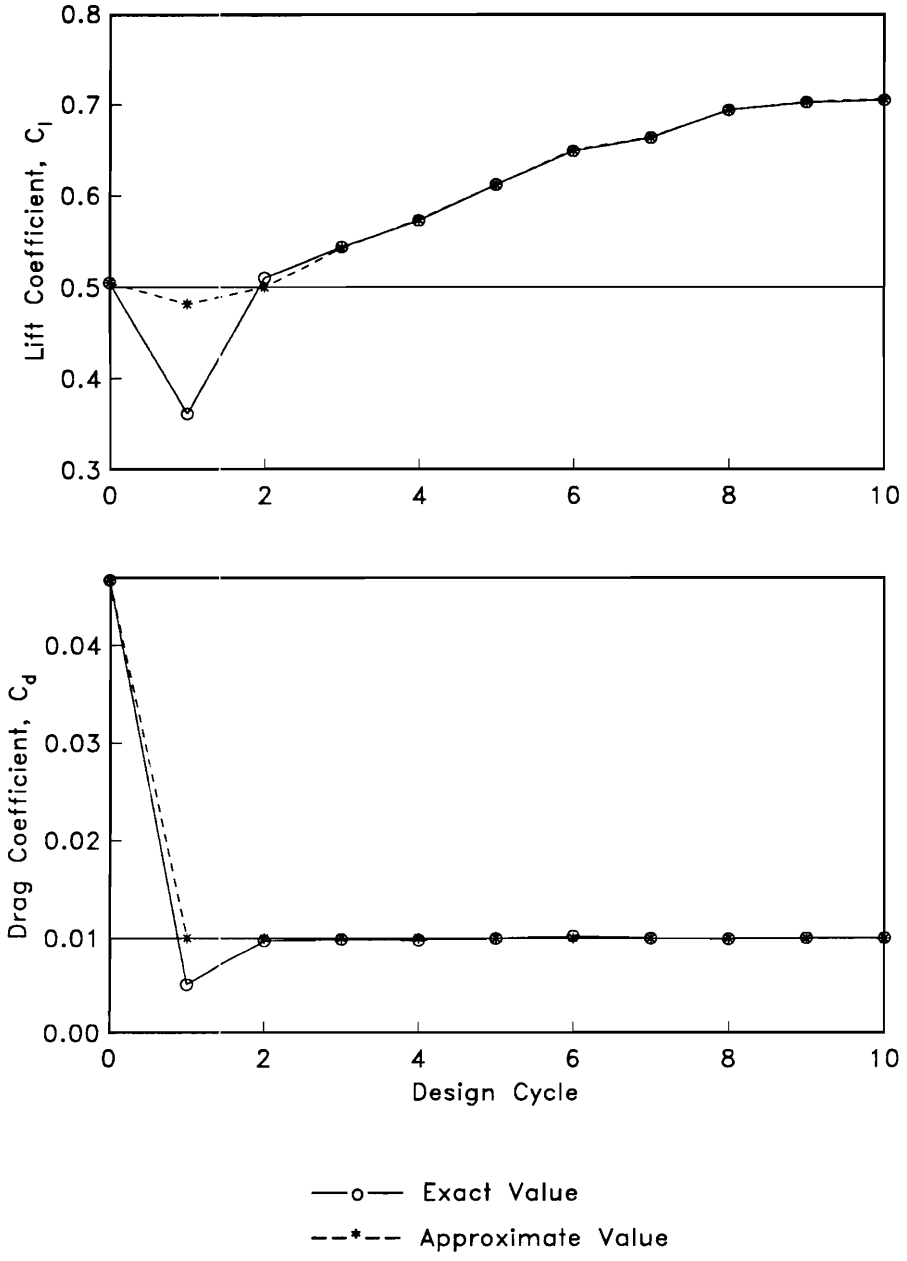



**Figure 17c.** Euler Convergence History (Table 10c) : Strategy C, Approximate Optimization with Absolute Move Limits.

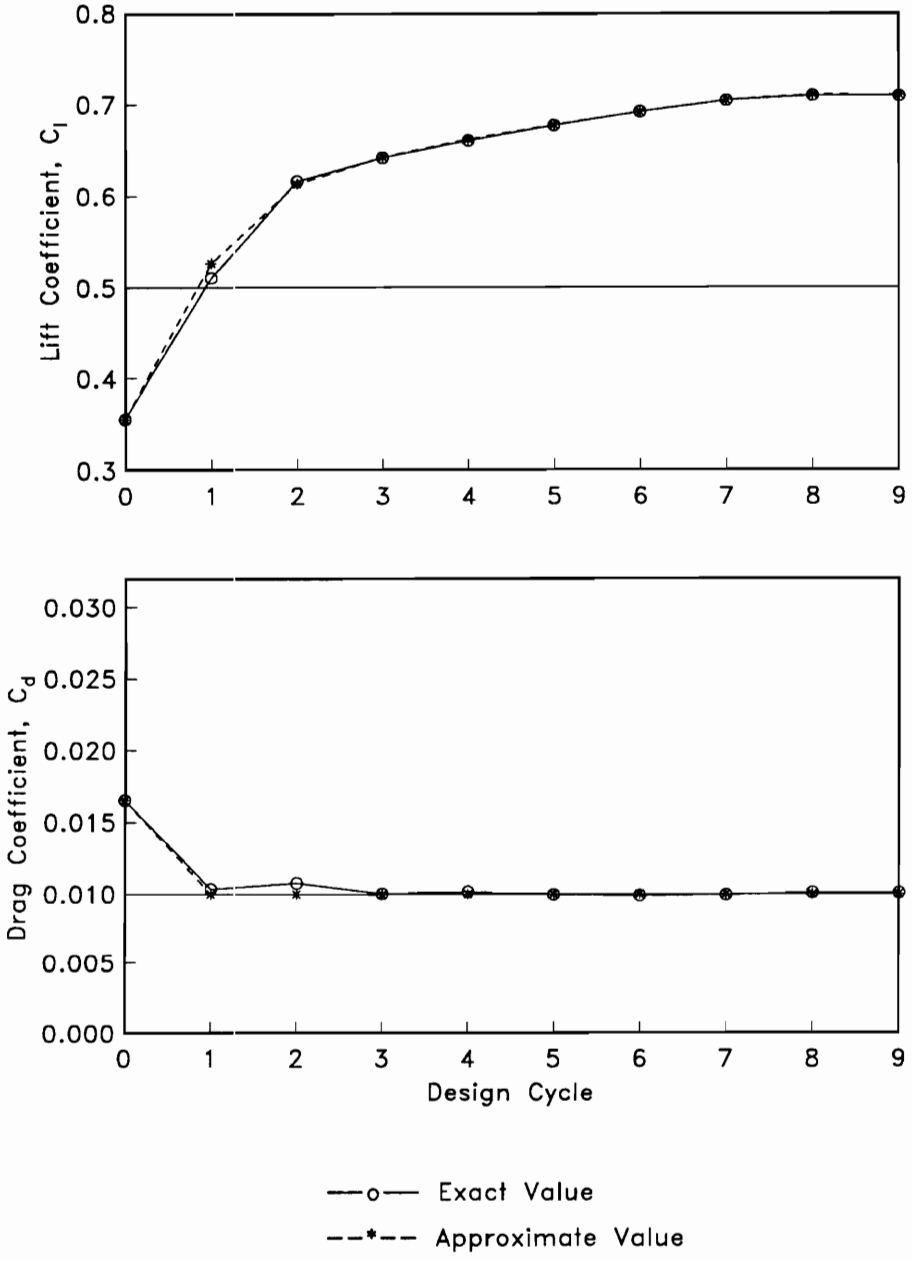



**Figure 17d.** Euler Convergence History (Table 10d) : Strategy C, Approximate Optimization with Absolute Move Limits.

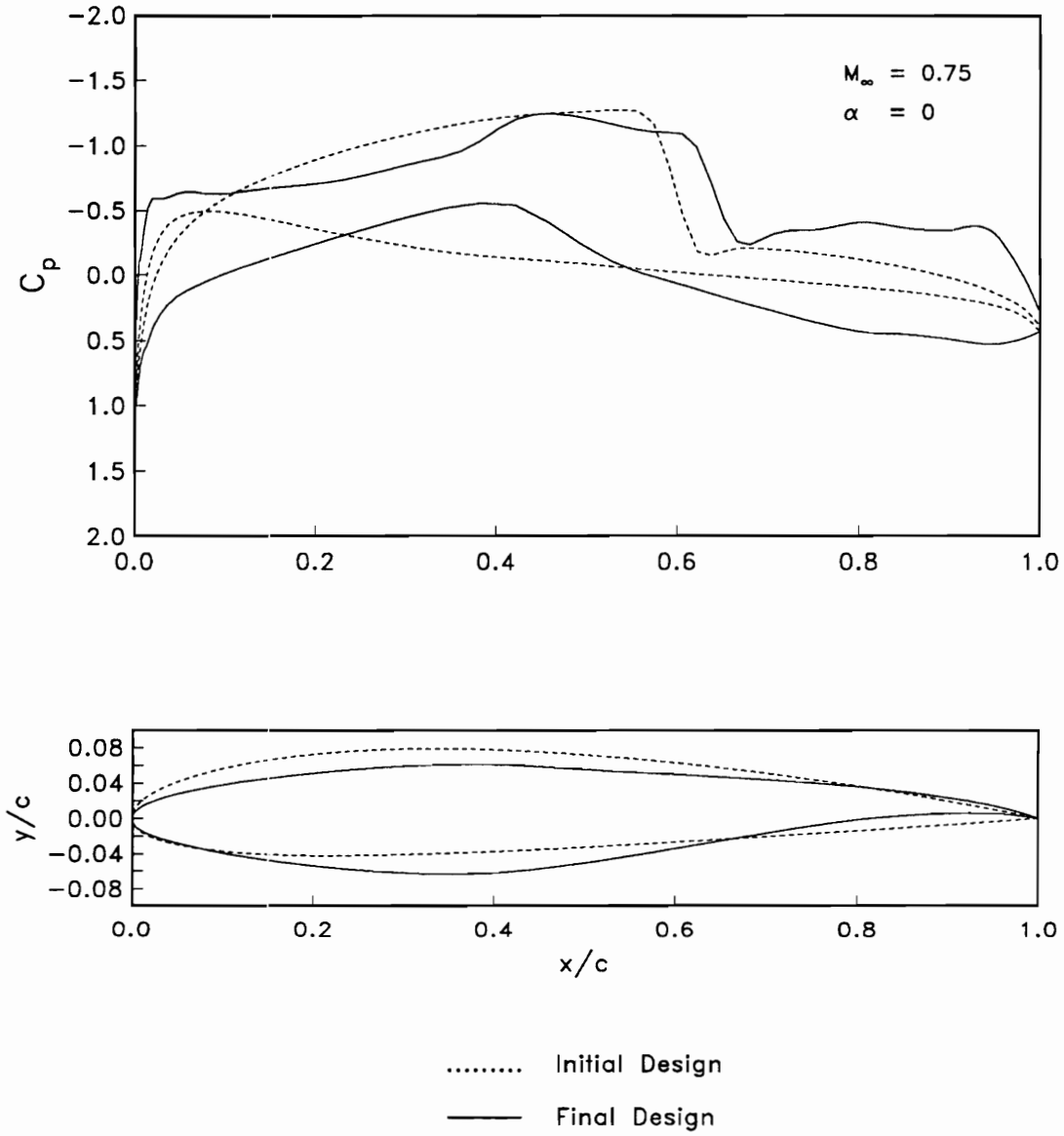



**Figure 18a.** Euler Design Results (Table 10a) : Strategy C, Approximate Optimization with Absolute Move Limits.

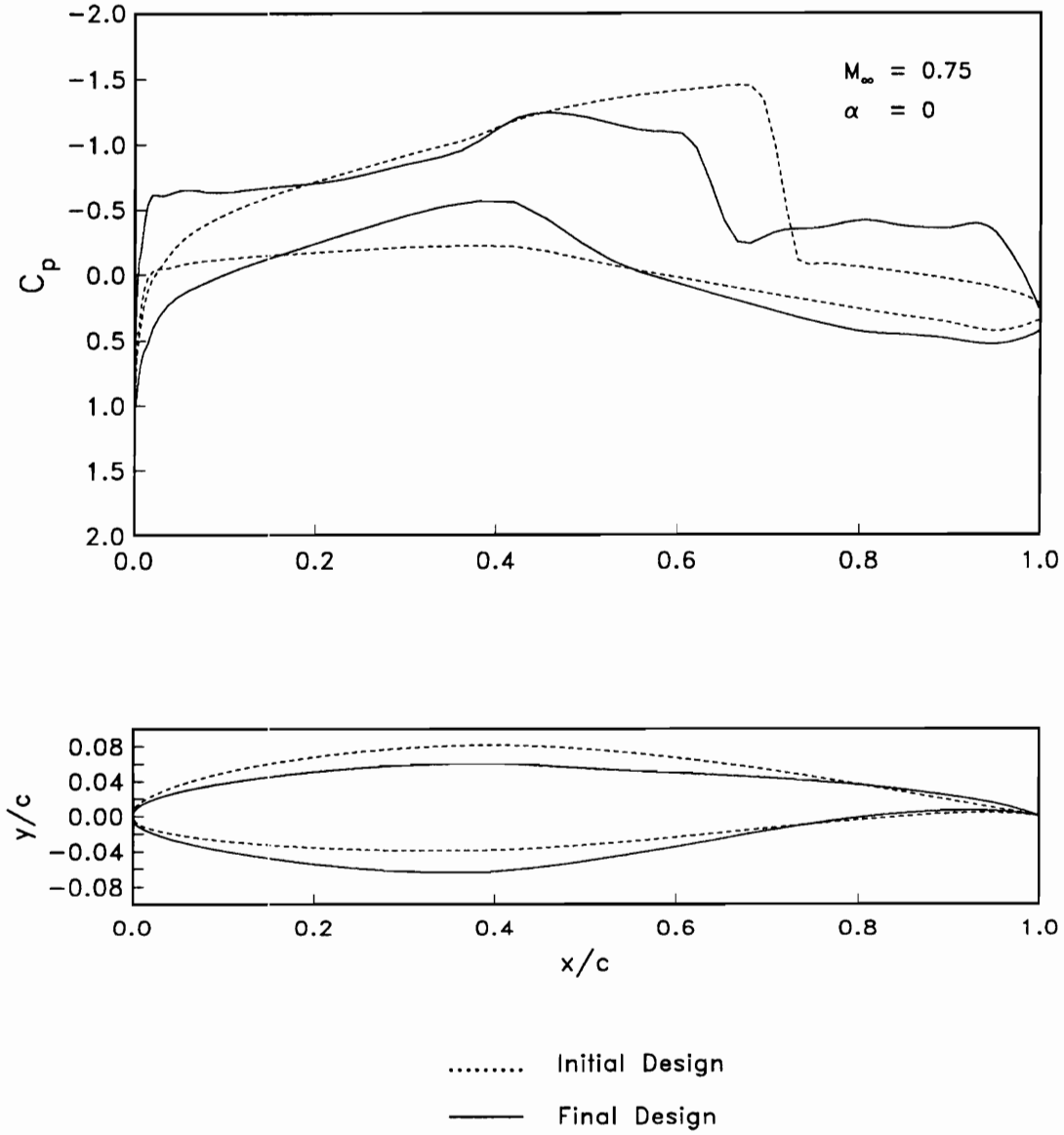



**Figure 18b.** Euler Design Results (Table 10b) : Strategy C, Approximate Optimization with Absolute Move Limits.

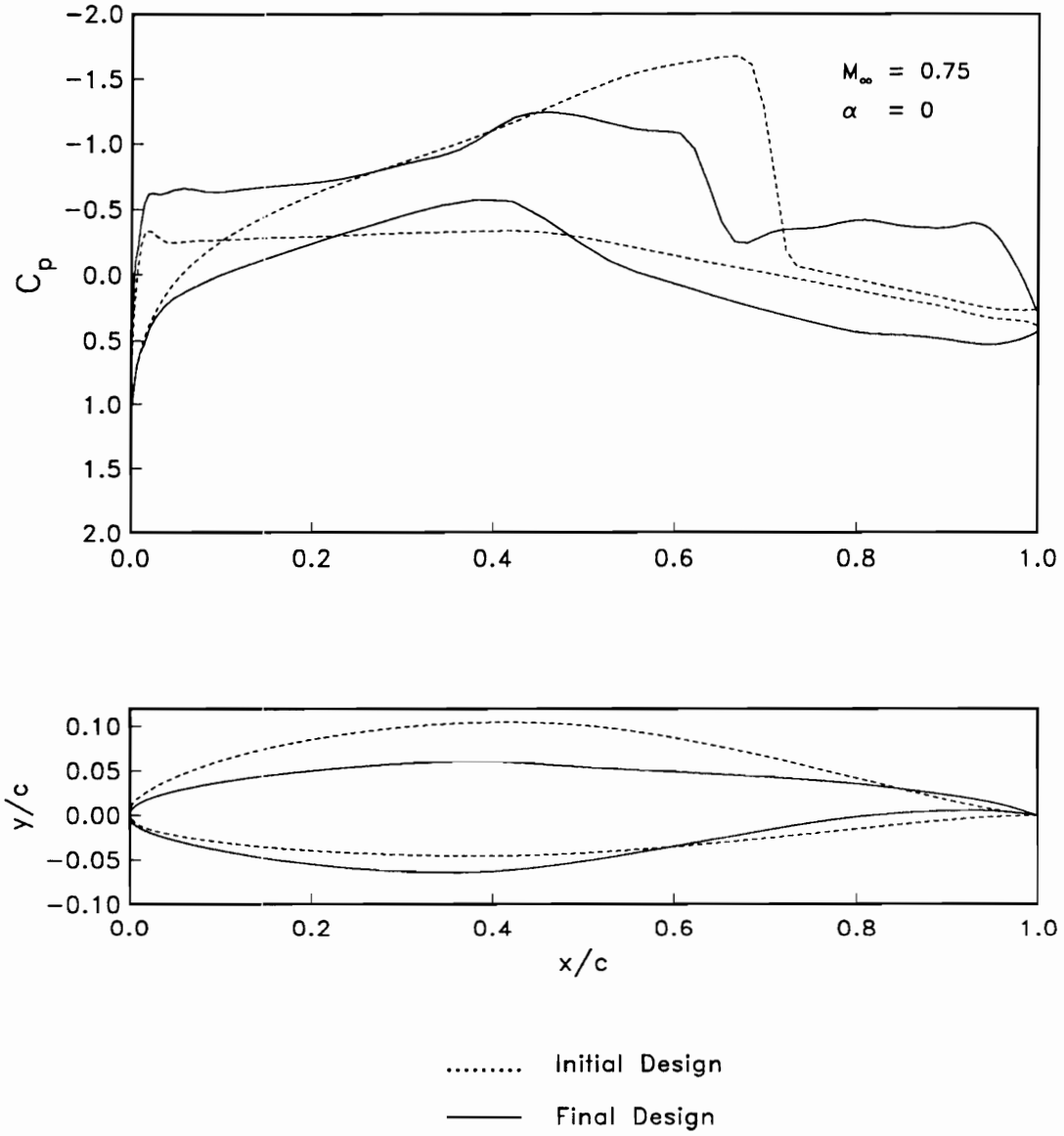



**Figure 18c.** Euler Design Results (Table 10c) : Strategy C, Approximate Optimization with Absolute Move Limits.

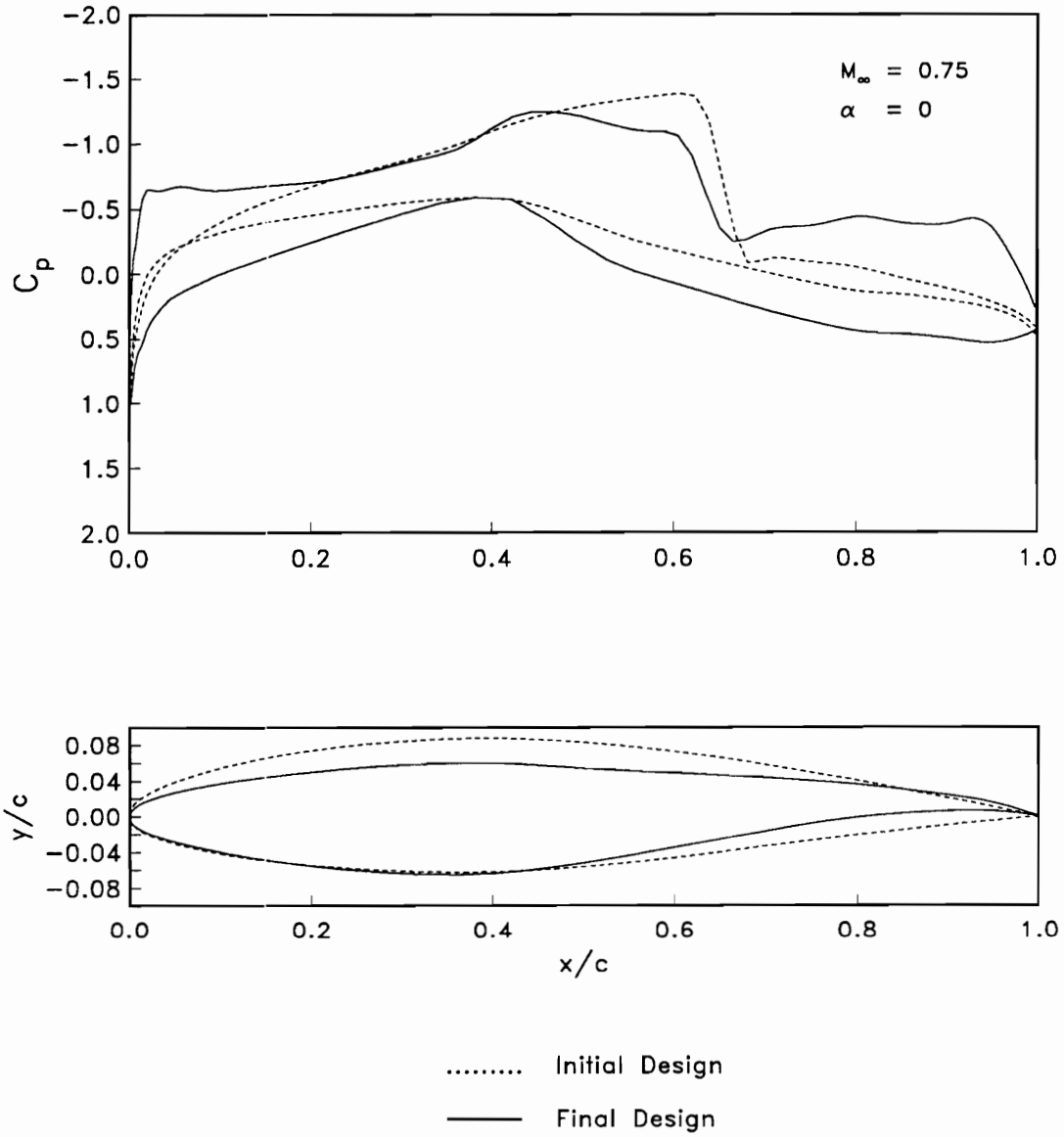



**Figure 18d.** Euler Design Results (Table 10d) : Strategy C, Approximate Optimization with Absolute Move Limits.

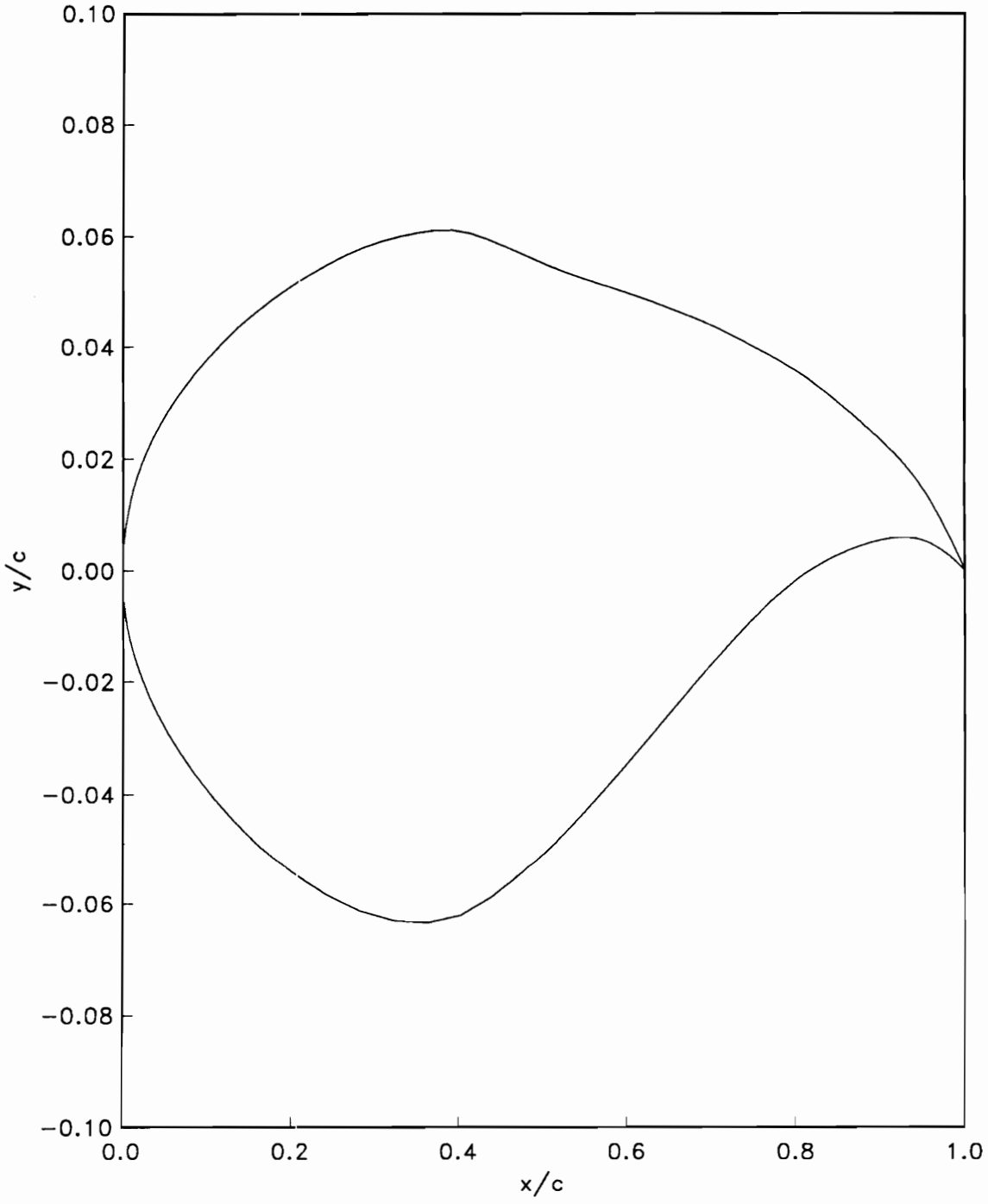



**Figure 19.** Detail Airfoil Geometry of the 9th Design in Table 10a.

# APPENDIX A. ESTIMATION OF EQUIVALENT LIFT COEFFICIENT

## *A.1 Kuhn-Tucker Conditions*

The typical optimization problem which includes inequality constraints only is

$$\begin{aligned} &\textit{minimize} \quad f(\bar{X}) \\ &\textit{such that} \quad g_j(\bar{X}) \geq 0 \qquad j = 1, \ldots, n_g \end{aligned} \tag{A.1}$$

where $f$ is the objective function, $g_j$ is the j-th inequality constraint of which total number is $n_g$ and $\bar{X}$ is the vector of $n$ design variables $\bar{X} = (X_1,\ X_2, \ldots, X_n)^T$. The necessary conditions for a minimum known as *Kuhn-Tucker conditions* for the problem (A.1) may be as follows:

> A point $\bar{X}$ is a local minimum of the problem (A.1) only if a set of non-negative $\lambda_j$ may be found such that

$$\begin{aligned} &1. \quad \frac{\partial f}{\partial X_i} - \sum_{j=1}^{n_g} \lambda_j \frac{\partial g_j}{\partial X_i} = 0 \qquad i = 1, \ldots, n \\ &2. \quad \lambda_j = 0 \quad \text{if j} - \text{th constraint is not active.} \end{aligned} \tag{A.2}$$

## *A.2 Computing the Lagrange Multipliers and Equivalent Lift Coefficient*

The Kuhn-Tucker conditions are often used to check whether a candidate minimum point satisfies the necessary conditions. However, more valuable thing in practical applications is that the conditions can be used for estimating the sensitivity of the optimum solution to small changes in the problem definition. The equivalent lift coefficient used for the global convergence checking in chapter 6 also

were calculated following this application. In both cases the Lagrange multipliers at a given point are needed to calculate.

Eq. A.2 in matrix form is

$$\mathbf{g} - \mathbf{N}\lambda = 0\,, \tag{A.3}$$

where g is the gradient vector of f, and the matrix bf N is defined by

$$n_{ij} = \frac{\partial g_j}{\partial X_i} \qquad j = 1,\ldots,r \quad i = 1,\ldots,n \tag{A.4}$$

with the total $r$ active constraints. Typically $r$ is less than $n$, so that Eq. A.3 is overdetermined with $n$ equations and $r$ unknowns. We assume that the gradients of the constraints are linearly independent so that N is of rank $r$. If the Kuhn-Tucker conditions are satisfied exactly, then only single solution exists. In this case, a subset of $r$ equations from $n$ equations can be selected to solve for the Lagrange multipliers. However this approach may be susceptible to amplification of errors. Instead a least-squares approach can be used to solve the equations. If we define a resiual vector u as

$$\mathbf{u} = \mathbf{N}\lambda - \mathbf{g}\,, \tag{A.5}$$

a least squares solution of Eq. A.4 will minimize the square of the Euclidean norm of the residual with respect to $\lambda$

$$\begin{aligned} \|\mathbf{u}\|^2 &= (\mathbf{N}\lambda)^T(\mathbf{N}\lambda) \\ &= \lambda^T\mathbf{N}^T\mathbf{N}\lambda - 2\lambda^T\mathbf{N}^T\mathbf{g} + \mathbf{g}^T\mathbf{g}\,. \end{aligned} \tag{A.6}$$

Differentiating Eq. A.6 with respect to each one of the Lagrange multipliers to minimize $\|\mathbf{u}\|^2$ leads

$$\lambda = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{g}\,. \tag{A.7}$$

The design problem considered in the present study can be written, from Eq. 2.1, as:

$$
\begin{aligned}
&\textit{maximize} \quad C_l(\bar{X}) \\
&\textit{such that} \quad C_{d_l} - C_d(\bar{X}) \geq 0 \\
&\qquad\qquad A(\bar{X}) - A_{min} \geq 0\,,
\end{aligned}
\tag{A.8}
$$

where $\bar{X}$ is the vector of design parameters $\bar{X} = (X_1,\ X_2, \ldots, X_N)^T$ specifying the airfoil geometry, $C_d$ is the drag coefficient due to wave drag, $C_{d_l}$ is the prescribed upper limit on wave drag, $A$ is the airfoil cross-sectional area non-dimensionalized by $c^2$, with c the airfoil chord and $A_{min}$ is the minimum required area.

Since this problem has 4 design variables and two active constraints in each approximate design cycle, the gradient vector $\mathbf{g}$ and the matrix $\mathbf{N}$ are respectively,

$$
\mathbf{g} = \begin{pmatrix} -\partial C_l/\partial X_1 \\ -\partial C_l/\partial X_2 \\ -\partial C_l/\partial X_3 \\ -\partial C_l/\partial X_4 \end{pmatrix}, \quad \mathbf{N} = \begin{bmatrix} -\partial C_d/\partial X_1 & \partial A/\partial X_1 \\ -\partial C_d/\partial X_2 & \partial A/\partial X_2 \\ -\partial C_d/\partial X_3 & \partial A/\partial X_3 \\ -\partial C_d/\partial X_4 & \partial A/\partial X_4 \end{bmatrix}. \tag{A.9}
$$

In each design cycle, an optimum solution is obtained based on the approximations constructed there. Suppose we found small amount of constraint violations, after performed an exact analysis at the end of each approximate optimization. Then the equivalent lift coefficient to consider small changes in the problem definition due to those violations can be estimated as

$$
C_{l_{eqv.}} = C_l - \lambda_1 \Delta C_d - \lambda_2 \Delta A\,, \tag{A.10}
$$

where $C_l$ is the lift coefficient obtained by the exact analysis, $\Delta C_d$ is the amount of the violation in the wave drag constraint and $\Delta A$ in the area constraint and both $\lambda_1$ and $\lambda_2$ are the Lagrange multipliers corresponding to wave drag and area constraints respectively. More details and useful examples in the applications of Lagrange multipliers are well provided in Haftka *et al.*[118]

# VITA

The author was born in Pusan, Korea on February 9, 1957. He entered the Seoul National University in February 1975 and received his B.S. degree in Aeronautical Engineering in February 1979. He received his M.S. degree in Aeronautical Engineering from KAIST (Korea Advanced Institute of Science and Technology) in February 1981.

He worked at the Korea Institute of Aeronautical Technology as a research engineer and at the Aerospace Division of Korea Air Lines as a design and manufacturing engineer for more than six years.

He started his Doctoral curriculum at the Aerospace and Ocean Engineering Department, Virginia Polytechnic Institute and State University in August 1987.

He is currently a member of Korean Scientists and Engineers Association.