

**A GENERAL INVERSE DESIGN PROCEDURE
FOR AERODYNAMIC BODIES**

by

Michael L. Papay

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

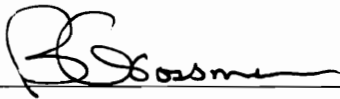
in

Aerospace Engineering

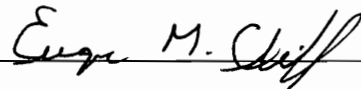
APPROVED:



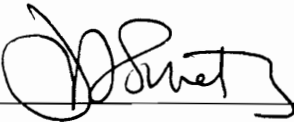
Robert W. Walters, Chairman



B. Grossman



E. M. Cliff



J. A. Schetz



W. Devenport

May, 1994

Blacksburg, Virginia

A GENERAL INVERSE DESIGN PROCEDURE FOR AERODYNAMIC BODIES

by

Michael L. Papay

Dr. Robert W. Walters, Chairman

Aerospace Engineering

(ABSTRACT)

A general inverse design procedure has been developed to use optimization techniques and generic surface descriptions for the purpose of aerodynamic shape design. A variety of flow regimes are examined from 2-D inviscid, subsonic cases to 3-D turbulent, supersonic problems. Surface descriptions have been generalized through the use of B-splines to model a variety of curves and shapes with a minimum of parameters. The process uses a computational fluid dynamics program, GASP (the General Aerodynamic Simulation Program), and several iterative and optimization techniques to examine bodies of interest.

A 2-D inviscid, subsonic airfoil test case demonstrates the ability of the procedure to solve problems governed by elliptic equations. A 3-D, viscous, compressible flow over a forebody/canopy model of a supersonic fighter and its comparison to test data establishes the ability of the method to solve practical problems of interest. Several other test cases are performed, including an axi-symmetric power law body and a 3-D elliptic cone. Unconstrained multi-parameter optimizations have been quite successful in matching target pressure coefficients and reproducing target body shapes.

DEDICATION

This work is dedicated to my wife Kristy, who made the long, arduous process of qualifying exams, classes, research and dissertation writing seem pleasurable and worthwhile. She has the unique ability to listen and understand me, and provide me with much needed motivation to stay off the golf course.

I would like to take this opportunity to express my appreciation to my parents for giving me sixteen years of formal education, and a lifetime of happiness. I admire their ability to juggle jobs, children and a personal life, and I hope that one day I can raise a family with as much love and dedication as they have.

ACKNOWLEDGMENTS

First of all, I would like to thank Dr. Robert Walters for the years of instruction, advice and encouragement throughout undergraduate and graduate school. Although it hurts me to say this, he certainly has more CFD knowledge in his little finger than I will ever possess.

The other members of my committee, Dr. Bernard Grossman, Dr. Gene Cliff, Dr. Joseph Schetz, and Dr. William Devenport, have not only been very helpful in the writing of this dissertation, but also in providing me with a solid background in many facets of aerospace engineering.

Much appreciation is due TRW, Inc. for supporting me on a Ph.D. fellowship while earning my degree. Special thanks to Dr. Robert Strickler and Gary Schelin for their patience.

Last but certainly not least, thanks to all the past and present residents of the Sun Room; for without their support, encouragement and humor, this degree would have been impossible.

TABLE OF CONTENTS

1. Introduction.....	1
1.1 Systems Of Equations.....	1
1.2 The Aerodynamic Analysis Program.....	2
1.3 The Inverse Design Problem.....	4
2. Axisymmetric Inverse Design.....	9
2.1 Axisymmetric Power Law Body - Inviscid.....	9
2.2 Axisymmetric Power Law Body - Viscous.....	15
3. Three Dimensional Inverse Design.....	17
3.1 3-D Elliptic Cone - Inviscid Calculation.....	17
3.2 Secant Method for a System of Equations.....	19
3.3 One-Parameter Optimization.....	27
4. Quasi-Newton Optimization Method.....	32
5. Analytic Forebody Inverse Design.....	35
5.1 Analytic Forebody Grid Generation.....	35
5.2 Comparison with Test Results.....	36
5.3 Three Dimensional Inverse Design - Inviscid.....	38
6. Surface Description Using B-Splines.....	43
6.1 B-Spline Curves.....	43
6.2 Non-Uniform Rational B-Splines.....	45
6.3 Three Dimensional B-Spline Surfaces.....	47

7. Elliptic Multi-Parameter B-Spline Inverse Design.....	48
7.1 Circular Arc Airfoil Test Case	48
7.2 Circular Arc Airfoil Inverse Design.....	51
8. Multi-Parameter B-Spline Inverse Design.....	55
8.1 Multi-Parameter B-Spline Curve Generation.....	55
8.2 Selection of B-Spline Parameters.....	57
8.3 Multi-Parameter B-Spline Inverse Design - Inviscid.....	62
8.4 Starting the Iteration Process.....	63
8.5 Multi-Parameter B-Spline Inverse Design - Turbulent Flow	75
9. Conclusions	81
9.1 Conclusions About the Current Work.....	81
9.2 Recommendations for Future Work.....	82
10. Bibliography	83
11. Appendix.....	92
11.1 Sensitivity Analysis by Automatic Differentiation.....	92

LIST OF FIGURES

1. Power Law Body.....	11
2. Results of Power Law Body Inverse Design, Inviscid.....	14
3. Elliptic Cone/Power Law Body.....	17
4. Three Dimensional Cp-Theta relationship.....	18
5. Elliptic Cone Cross-Section.....	22
6. Pressure Coefficient Influence of Perturbing Theta3.....	23
7. Pressure Coefficient Influence of Perturbing Theta17.....	23
8. Diamond Cross Section.....	24
9. Multiple Solutions for Diamond Cross Section.....	26
10. Elliptic Cone Target and Inverse Design Pressure Contours.....	30
11. Elliptic Cone Pressure Coefficient Difference Contours.....	31
12. Inappropriate Iterative Guess.....	33
13. Analytic Forebody Baseline Grid.....	36
14. Inviscid GASP Calculations vs NASA Test Case.....	37
15. Parameter Variation Effect on Baseline Analytic Forebody.....	39
16. Inviscid Three Parameter Optimization Results.....	41
17. Parameter Variation by Iteration.....	42
18. B-Spline Blending Functions.....	44
19. B-Spline Example.....	45
20. Circular Arc Airfoil.....	48
21. Grid for the Circular Arc Test Case.....	49
21. Circular Arc Pressure Contours.....	50
23. B-Spline Approximation to the Circular Arc Airfoil.....	53

24. Pressure Distribution on the Circular Arc Airfoil	53
25. Grid Interpolation to Evenly Spaced Rays	56
26. Six Control Point, Quadratic B-Spline	58
27. Best B-Spline Fits for Plane 20	61
28. Inverse Design Methodology	65
29. Initial and Final B-Spline Approximations to Forebody, Plane 20	68
30. Inviscid Target C_p vs B-Spline Approximations, Forebody, Plane 20	69
31. Initial and Final B-Spline Approximations to Forebody, Plane 15	70
32. Inviscid Target C_p vs B-Spline Approximations, Forebody, Plane 15	71
33. Final B-Spline Approximations to Forebody, Planes 25, 30 and 35	72
34. Inviscid Target C_p vs B-Spline Approximations, Planes 25, 30 and 35	72
35. Inviscid Target Pressure vs Inverse Solution, Analytic Forebody	73
36. Analytic Forebody Pressure Coefficient Difference Contours	74
37. Turbulent GASP Calculations vs NASA Test Case	77
38. Near Wall Turbulent Velocity Profiles	78
39. Turbulent Target C_p vs B-Spline Approximations, Planes 15 and 20	80

LIST OF TABLES

1. Power Law Body Parameters.....	9
4. Inviscid Power Law Body Input Parameters.....	12
3. Inverse Design CPU Times for the Power Law Body	15
4. Inviscid Elliptic Cone Input Parameters.....	18
5. Analytic Forebody Test Conditions	37
6. Parameter Values and Bounds for Three Parameter Optimization.....	39
7. Inviscid Circular Arc Input Parameters.....	49
8. Circular Arc Airfoil Inverse Design	52
9. Optimal 6 Parameter B-Spline for Plane 20	59
10. Optimal 10 Parameter B-Spline for Plane 20	60
11. Inviscid Analytic Forebody Input Parameters	63
12. Turbulent Analytic Forebody Input Parameters	75

1. INTRODUCTION

The design of aerodynamic shapes through the use of computational fluid dynamics is well within the reach of engineers today because of the many advances in processor speed and technology. The days have all but disappeared when a shape is generated and analyzed by the designer, then modified (based on his experience) and redesigned again and again until a suitable result is found. Through the use of inverse design procedures, the engineer can use his experience to determine the desired performance of the shape at the beginning of the design process, and put the speed of the processor to work solving the problem. However, inverse design methods have traditionally been quite specialized and finding a general method to solve a variety of problems and shapes has been difficult. The present work is aimed at circumventing techniques that are only applicable to certain flows in favor of a more general approach.

1.1 Systems Of Equations

The Navier-Stokes equations govern almost all commonly encountered flows. However, computing a solution to the full Navier-Stokes equations is often impossible or computationally impractical [1]. Several levels of simplification are common in the field, and have been utilized in this study. The Euler equations govern the flow of an inviscid, non heat-conducting fluid and are used to solve all of the inviscid cases presented here. Also, the fluid was assumed to be a perfect gas throughout the flow regime, though this simplification need not have been made, based on the design procedure developed in this study. For the laminar and turbulent test cases, the Reynolds Averaged Navier-Stokes equations were solved, but only the thin-layer contributions in the direction normal to the

wall were considered. This solution methodology assumes that there is no separation or recirculation in the flow.

For a set of integral conservation laws that is elliptic in character, a global technique must be used to solve the entire flowfield at once. If the equations are hyperbolic or parabolic, the mathematical problem to be solved is an initial value or initial boundary value problem. When the entire flowfield is supersonic and steady, the Euler equations become hyperbolic and a space-marching technique can be used to compute the solution by marching outward from the initial data surface while satisfying the boundary conditions [1]. The marching "direction" is in the axial direction of the body, so that the initial data surface is often the first "plane" or cross-section of a body. To correctly apply the space marching technique, it is necessary that the velocity *component* in the marching direction be supersonic. This technique is computationally advantageous, and was applied in several of the test cases. Moreover, for high speed viscous flows with an attached boundary layer, it is also possible to solve the "parabolized" form of the Navier-Stokes equations, again with a space marching procedure, at a considerable reduction in computational cost.

1.2 The Aerodynamic Analysis Program

The flow solver that was utilized in this study was GASP 2.0 (the General Aerodynamic Simulation Program, Version 2.0) [2]. GASP solves the integral form of the Navier-Stokes equations and its subsets including the Thin-layer N-S, Parabolized N-S, and Euler equations using a finite volume formulation. The code supports a wide variety of implicit and explicit time integration schemes and gives the user a range of thermo-chemical modeling options.

In two-dimensional space marching problems, such as the axisymmetric power law body examined here, the linearization of the residual to perform the implicit time integration results in a single block tri-diagonal matrix on each marching plane that can be solved by LU decomposition. In two dimensional global problems, such as the circular arc airfoil, the linearization of the residual is approximated by assuming that the implicit operator can be decoupled in space, resulting in a two factor Approximate Factorization (AF) scheme. This reduces the problem to a system of uncoupled block tri-diagonal problems in each of two directions. In three-dimensional space marching problems, such as the elliptic cone and the analytic forebody, the problem is equivalent to solving a series of two dimensional global problems, and the two factor AF scheme is used in the cross-flow plane. Since all the problems considered here are steady, an implicit time integration scheme has been utilized to achieve the fastest possible convergence rate [2].

When performing space marching calculations in viscous flow regimes, the Parabolized Navier Stokes (PNS) equations are solved by GASP. The PNS calculations were conducted with second order spatial accuracy by retaining the time dependent term in the N-S equations and using a three-point backward differencing scheme that includes information from the two previous planes. Iterations were then performed at each computational plane to reduce the residual of the steady PNS equations.

The nomenclature used to describe the indices for the grids is a standard right-handed ijk coordinate system. For example, $i0$ is the first grid point in the i direction, and $idim$ is the last grid point in the i direction.

Although GASP was utilized as the aerodynamic analysis code in this study because of its general nature and ability to solve problems in a wide variety of flow regimes, any flow solver could be used with this inverse design procedure.

1.3 The Inverse Design Problem

The inverse design problem can basically be summed up as follows: given the desired performance parameters, design the body that produces these parameters. For example, a designer might specify the pressure coefficient (C_p) distribution that produces the lift and drag on an airfoil that meets his needs. The inverse design procedure determines the shape of the airfoil that most closely matches that pressure coefficient distribution. The direct problem, using aerodynamic analysis to determine the C_p from the body shape, is well posed and has received considerable attention in the past. The inverse problem however, determining the body shape from a C_p distribution, suffers from questions about existence, uniqueness and continuous dependence. Whereas in aerodynamic analysis a very small change in body shape produces a small change in the solution, the converse is not necessarily true for inverse design problems. Attempts have been made to address the existence and uniqueness issues of certain problems.

In two-dimensional, inviscid, incompressible flow over an airfoil, Mangler and Lighthill [3] showed that a unique solution exists to the inverse problem if a certain constraint is met regarding the relationship between the free stream and the surface speed. If a closed airfoil profile is required, two other constraints appear [4]. Volpe [5] extended this work to transonic flow over a 2-D airfoil. For viscous flows over three-dimensional bodies of arbitrary shape, no closed form inverse design procedure exists similar to the above work. In fact, since the method of Lighthill is based on the speed on the surface of the airfoil, and a conformal transformation of the airfoil shape to a circle, its usefulness is limited to 2-D, inviscid problems.

Although many more inverse design methods have been developed, few are general enough to perform well in more than one flow regime. As an example, Lee and Mason [6] developed an inverse method for supersonic and hypersonic body design. It performed

well on both 2-D and 3-D inviscid test cases, but like other inverse methods of its type, was limited to inviscid flows. This limitation arises in the method of posing the inverse problem. A surface streamline along the body is used in a surface pressure-body geometry rule to arrive at a corrected body geometry. Although this method provided rapid, reliable convergence for the test cases studied, it was deemed necessary to move away from the surface streamline approach to develop a more general procedure applicable to many flows and geometries.

In studying the inverse design procedures of the past, it becomes clear that there are three main parts to each method. The selection of each of these three segments determines how comprehensive the technique is. The three parts are:

1. a flow solving program, or an approximation that yields flow results
2. a surface description program, or a method of generating the geometry
3. a technique for converging to a desired solution

In Lee and Mason [6] the flow solving code used was *cf3de*, the geometry was generated through a corrected body geometry technique based on the surface pressure-body geometry rules, and a Regula-Falsi technique was used to arrive at a converged solution. Other authors have used even more specialized methods. Dulikravich [7] used a simple Modified Newtonian Theory (good only for hypersonic and supersonic flows), while using an optimizer to design cross sections based on the super-elliptic Lamé function to optimize the shape of hypersonic missiles. Bock [8] designed minimum drag bodies using an Euler space marching code combined with an optimizer (CONMIN) for bodies of revolution at supersonic speeds. Bowcutt, Anderson and Capriotti [9] designed hypersonic waveriders performing in viscous flow regimes using a non-linear simplex

method [10]. Most recently, Burgreen and Baysal [11] have employed a Bezier-curve surface description of three dimensional wings to obtain inverse solutions in inviscid flows. Other geometry generation techniques for airfoils involve superimposing combinations of basis airfoils until a desired geometry is reached, for example, the work of Vanderplaats [12] and Davis [13]. The work of Davis was extended by Ratcliff and Carlson [14] in curvilinear coordinates. Birckelbaw [15] used a Bezier curve fitting routine coupled with a Navier-Stokes flow solver to perform 2-D airfoil design in flows including viscous effects. Selig and Maughmer [16] used a multi-dimensional Newton iteration technique and an inviscid airfoil design method coupled with a direct integral boundary-layer analysis technique to perform 2-D airfoil design.

In their 'Smart Aerodynamic Optimization' concept [17], Aidala, Davis and Mason used a combination of ideas. Design variables, or shape functions, were generated by various inverse methods and applied in a numerical optimization technique to perform inverse design analysis.

Much of the work in inverse design has been in the transonic wing design area. Since Volpe and Melnik [5], Xia [18] has utilized a full potential method, a geometry correction and a residual-correction method for the inverse design of transonic airfoils and wings. Campbell and Smith [19-22] have done a considerable amount of work on the direct/iterative approach to transonic wing design. Each of these methods has merit within its framework, but none to date are capable of a generic body shape in any specified flow regime.

Many inverse design procedures are also limited to pressure coefficient matching because of body shape rules and algebraic procedures inherent in the technique. This is not all bad, since C_p can provide a wealth of information about the performance of a body, including the design of a pressure gradient to maintain laminar flow at high velocities. The

method described here uses the pressure coefficient for design purposes, but in principle the procedure could have used skin friction coefficient, heat transfer coefficient, or a number of other variables as an objective function.

As described in this work, the use of B-splines for surface generation is an improvement in inverse design techniques. B-splines and a subset, Non-Uniform Rational B-splines, are currently in use throughout the aircraft design industry as Computer Aided Design (CAD) tools. The ACSYNT (AirCraft SYNThesis) Institute, composed of eight U.S. aerospace companies, several NASA and Navy centers, and Virginia Tech's CAD Laboratory, have developed a feature-based, parametric computer aided aircraft conceptual design code called ACSYNT. This code features a highly interactive graphical user interface, and automatically generated B-spline surface models for both detail design and preliminary aircraft design [23].

Non-uniform bicubic B-spline surface models in ACSYNT guarantee that the surfaces generated for each component feature have curvature continuity except where tangent or positional continuity is required. At surface intersections, B-spline fillets are used to produce curvature continuous fillets over most regions [23]. A procedure is underway for skinning the complete aircraft component model with a curvature continuous surface through the use of non-uniform hierarchical B-spline surfaces [24].

The widespread use of B-spline modeling in aircraft design makes their investigation attractive in inverse design procedures. Moreover, B-splines exhibit a desirable feature for optimization, a large amount of shape control with few parameters. Most of the computational cost in any inverse design procedure is in the function evaluation routine, the flow solver [12]. Because the calculation of a gradient through finite differences involves multiple calls to the flow solver, and it is often necessary to compute gradients for the optimization routines, a low number of parameters is desirable.

Other methods of obtaining the necessary derivatives, such as the sensitivity analysis performed by Burgreen and Baysal [11] are limited to specific aerodynamic shapes. In their study of three-dimensional wings in transonic flow, a discrete sensitivity analysis approach was used to decrease the CPU time from a finite difference formulation.

The general nature of the procedure described here is what makes it unique. Through the use of a general flow solver, an optimizer able to solve a wide range of non-linear, constrained problems, and a flexible B-spline surface generator, a design tool has been developed that can solve a vast array of problems for the aerospace engineer.

2. AXISYMMETRIC INVERSE DESIGN

2.1 Axisymmetric Power Law Body - Inviscid

In order to develop a more general approach towards inverse design, one that could be applied to inviscid and viscous flows, a test case was employed that was used in the inviscid analysis of Lee and Mason [6]. A power law body with the equation

$$r = Ax^n \quad (1)$$

and the parameters shown in Table 1 was generated. This body was the subject of intense scrutiny by Lee and Mason and provided a good reference point from which to begin the current analysis. This shape is aerodynamically sharp [25], meaning it may be geometrically blunt near the nose, but behave as a sharp body and have low drag. The body tapers to a smooth aft section.

Table 1. Power Law Body Parameters

A	0.1667
n	0.75
Fineness Ratio	3.0
Mach Number	6.28

A view of the body is shown in Figure 1; not all grid lines were shown to improve the clarity of the picture. The surface of the body, defined by the $j0$ boundary, was

computed using equation 1, and the outer edge of the computational grid, the $jdim$ boundary, was designed to be large enough to ensure that the shock was completely within the grid at the analyzed Mach number. Since GASP utilizes a finite volume spatial discretization in which the state variables are stored at the centers of each cell, it is necessary to create a grid that has two grid points in the transverse, or i direction, for an axisymmetric body. The spacing in the axial direction was based on the square of the k coordinate

$$x_k = \left(\frac{k}{kdim} \right)^2 \quad \text{where} \quad k = 1, 2, 3 \dots kdim \quad (2)$$

and a hyperbolic tangent stretching function (to be discussed in more detail later) was used to cluster the grid points near the wall of the body, in anticipation of solving the problem with the assumption that the flow was laminar and the resolution of the boundary layer near the wall would become important.

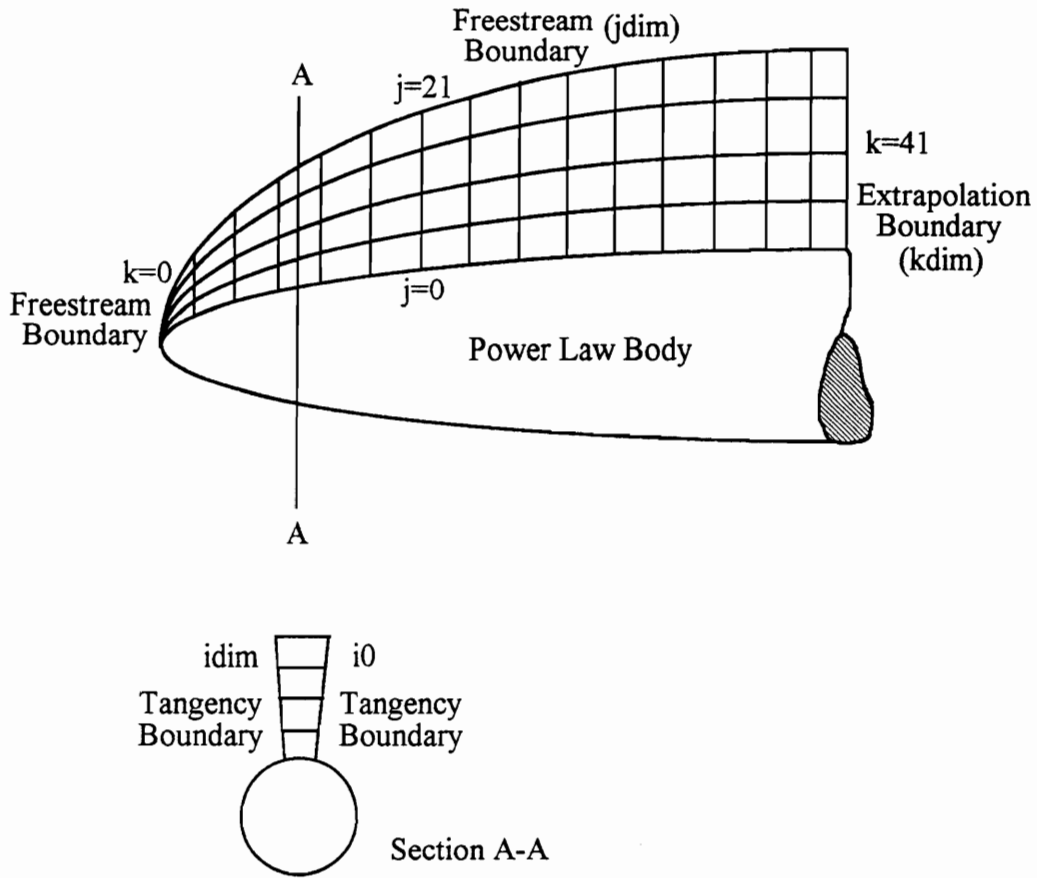


Figure 1. Power Law Body

To perform the inverse design, a target pressure distribution must first be generated. For demonstration purposes this was done by running GASP in the space marching mode with the parameters given in Table 4. The variables $idim$, $jdim$ and $kdim$ are the number of grid points in the transverse, normal to the body and axial directions, respectively (Figure 1).

Table 2. Inviscid Power Law Body Input Parameters

idim (transverse direction)	2
jdim (direction normal to body)	21
kdim (streamwise direction)	41
i0 Boundary Condition	Tangency
idim Boundary Condition	Tangency
j0 Boundary Condition	Tangency
jdim Boundary Condition	Fixed at free stream
k0 Boundary Condition	Fixed at free stream
kdim Boundary Condition	First order extrapolation

The boundary conditions are typical GASP input parameters. The tangency boundary condition assumes that there is no flow normal to the local face, and is appropriate for inviscid problems with solid walls [2]. The boundary condition which fixes the flow quantities at the free stream is appropriate for the exterior points of the grid in the j direction, and is a reasonable assumption at the nose of the body. At the tail of the body ($kdim$ boundary), the flow quantities are extrapolated using a first order approximation.

GASP was utilized to compute the solution on the first three k planes of the target body to get a starting point for the inverse calculations. This was done to facilitate the use of the computationally advantageous space marching option in GASP. Since the space marching mode uses the solution from the previous plane as an initial estimate for the solution on the current plane, instead of the free stream quantities, the CPU time to converge to the solution on the current plane is usually reduced. Therefore, the first three planes were converged on the target body, and the inverse design procedure was started on the fourth plane, using the secant method to determine the angle of the new segment.

For the secant method, two points are needed to start the iterative process. The first point, at the $n-1$ iteration

$$(\theta^{n-1}, C_p^{n-1}) \quad (3)$$

was found by using the information already known about the problem, the target pressure distribution. An approximation that relates the two variables of interest is Newton's Sine Squared Law [26]

$$C_p = 2\sin^2 \theta \quad (4)$$

which is a good approximation for C_p in high supersonic and hypersonic flows with a high density and an essentially inviscid shock layer. Newtonian flow theory is based upon the concept of an infinitesimally thin shock layer which coincides with the surface of the body, with the assumption that there is no friction between the layer and the body surface [27].

The angle θ^{n-1} , which is relative to the local horizontal, was used to generate a new plane. C_p^{n-1} is the pressure coefficient resulting from the GASP computation on this new plane. A small perturbation was made, $\Delta\theta$, to get θ^n and C_p^n in a similar manner. The secant method was then used to find the next point in the iterative scheme.

$$\theta^{n+1} = \theta^n - \frac{\theta^n - \theta^{n-1}}{C_p^n - C_p^{n-1}} (C_p^n - C_p^{target}) \quad (5)$$

This process was repeated until the plane was converged to a satisfactory tolerance,

$$\frac{|C_p - C_p^{target}|}{C_p^{target}} < \varepsilon \quad (6)$$

where $\varepsilon=1.0 \times 10^{-5}$. The entire iterative process was repeated on each plane until the end of the body was reached. The results are shown in Figure 2. Of course, the first three points are exact, since they are actually comparing the target body with itself, but the agreement is good for the rest of the body as well.

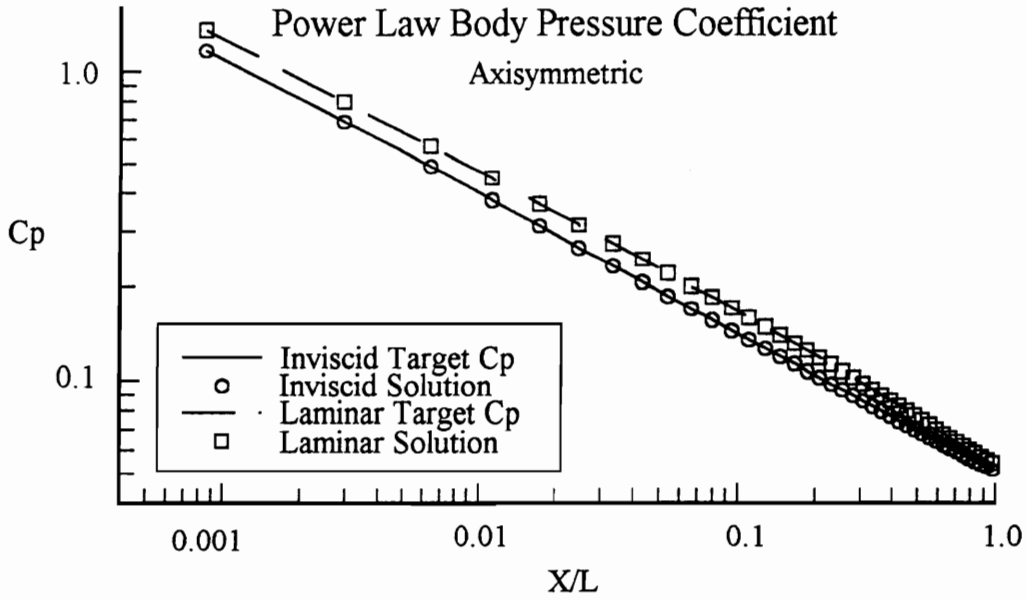


Figure 2. Results of Power Law Body Inverse Design, Inviscid

2.2 Axisymmetric Power Law Body - Viscous

The same body was computed with new flow parameters for the viscous demonstration. Since the inverse design procedure makes no assumptions regarding streamlines on the body surface, or any other limiting assumptions, the viscous test case was simply a matter of changing a few input parameters and performing the calculations again. A Reynolds number of 1.0×10^6 was chosen to be in the range of laminar flow for Mach 6.28 [28]. The parameters that were changed for the laminar case are consistent with a viscous flow analysis; the j_0 boundary condition was set to be a no slip condition, instead of no flow normal to the wall, and the free stream density was adjusted to match the desired Reynolds number. The inverse design procedure follows the same secant method as described for the inviscid case. CPU times in Table 3 are for the High-Speed Processor Subsystem VonNeumann, which is the Cray Y-MP C-90 supercomputer at the Numerical Aerodynamic Simulation Program at NASA Ames.

Table 3. Inverse Design CPU Times for the Power Law Body

Inviscid Planes 1-3	Inviscid Planes 4-40	Laminar Planes 1-3	Laminar Planes 4-40
3.96 sec	63.26 sec	8.58 sec	64.79 sec

The results for the laminar case are also shown in Figure 2. In both the inviscid and the laminar case, the pressure coefficients were matched to within the specified tolerance on each plane, and the original body geometry was recovered.

As discussed in Lee [10], for an axisymmetric body at supersonic/hypersonic velocities, the solution can be found to be unique if $C_p(\theta)$ is a single-valued function of θ . The Newtonian flow assumption, and several other surface pressure-body geometry rules examined by Lee, are monotonic functions of θ . Of course, in inviscid flow, many solutions may satisfy the non-dimensionalized target C_p by similarity, but for a given body length, there exists only one body shape that satisfies the target C_p at the conditions given in this test case.

3. THREE DIMENSIONAL INVERSE DESIGN

3.1 3-D Elliptic Cone - Inviscid Calculation

The natural extension from a two dimensional case was to proceed to a fully three dimensional test case. In Lee's previous work, [10], an elliptic cone was used as a three dimensional test case. Because a conic body has the same solution on each plane, it is trivial to examine more than one plane at an angle of attack of 0° . Lee used a non-zero angle of attack to simulate a fully three dimensional body; a different approach was taken in the present work. The cross section of the body was an elliptic cone, but the body varied in the streamwise direction as a power law body. A cross section of the body and the associated computational grid is shown in Figure 3.

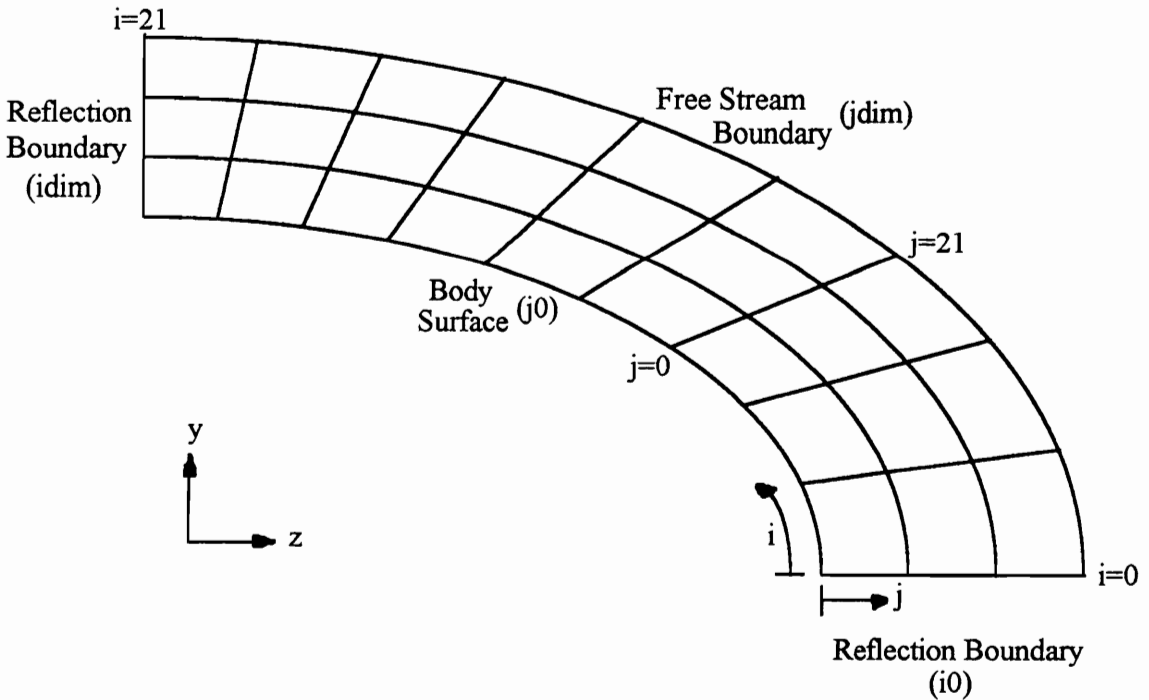


Figure 3. Elliptic Cone/Power Law Body

In Figure 3, the i direction was around the body, called the transverse direction, and the j direction was normal to the body. The k and x direction were into the plane of the page, or the axial direction. The body had one-fourth symmetry, which means it can be reflected about the x - z axis and the x - y axis to display the full body. The $i0$ and $idim$ boundary conditions in Table 4 are therefore set to model this symmetry. Since this case is inviscid, the $j0$ boundary condition is set to tangency, equivalent to no flow normal to the surface. As in the inviscid axisymmetric power law body case, the $k0$ boundary is set to the free stream, and the $kdim$ boundary is extrapolated from the interior points using a first order approximation. The $jdim$ boundary is again designed to contain the entire shock at the design Mach number.

Table 4. Inviscid Elliptic Cone Input Parameters

$idim$ (transverse direction)	21
$jdim$ (direction normal to body)	21
$kdim$ (streamwise direction)	21
$i0$ Boundary Condition	x - z plane symmetry
$idim$ Boundary Condition	x - y plane symmetry
$j0$ Boundary Condition	Tangency
$jdim$ Boundary Condition	First order extrapolation
$k0$ Boundary Condition	Fixed at free stream
$kdim$ Boundary Condition	First order extrapolation

The ratio of the width to height of the elliptic cone cross section was 2:1. The objective of this three-dimensional design was to match the pressure coefficient at the center of the cell faces on the body by adjusting the angles that the cell vertices made to the local horizontal. To simplify the number of parameters, the assumption was made to

keep the angle ϕ , in the circumferential direction, the same for each i location (Figure 4). The angle ϕ is the angle measured from the x - z axis to the position of the i^{th} coordinate. Therefore, θ_1 is at $\phi=0^\circ$ and θ_{idim} is at $\phi=90^\circ$, although for simplicity Figure 4 does not show the full quarter plane.

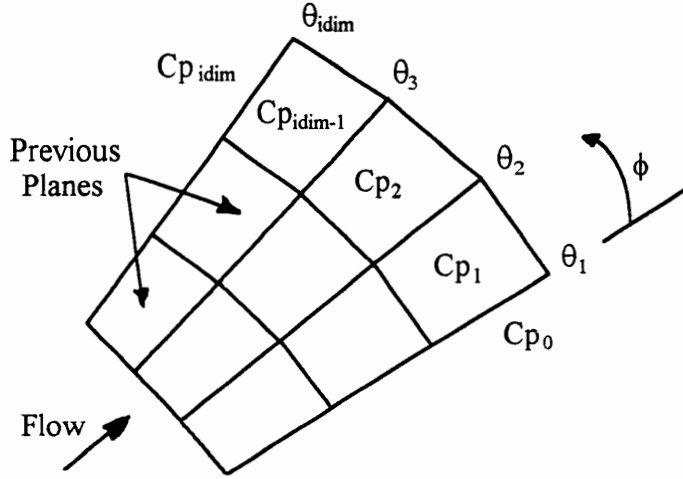


Figure 4. Three Dimensional C_p , θ relationship

The reflection boundary conditions assume that $C_{p0}=C_{p1}$ and $C_{p\text{idim}}=C_{p\text{idim}-1}$. As a simple example, an inverse design procedure operating on Figure 4 would attempt to match the pressure coefficients at each of the three internal points by adjusting the four θ parameters.

3.2 Secant Method for a System of Equations

To solve the system of equations generated by the problem, the secant method was again selected as an iterative scheme. The development of the secant method for a system of equations is straightforward. For the vector function

$$F(\theta) = 0 \quad (7)$$

a Taylor Series expansion yields

$$\mathbf{F}(\theta) = 0 = \mathbf{F}(\theta)^k + (\theta^{k+1} - \theta^k) \mathbf{F}'(\theta^k) + \frac{(\theta^{k+1} - \theta^k)^2}{2!} \mathbf{F}''(\theta^k) + \dots \quad (8)$$

Assuming that the $\Delta\theta^2$ term is negligible, Newton's method for a system becomes

$$\mathbf{F}'(\theta) \Delta\theta = -\mathbf{F}(\theta) \quad (9)$$

To obtain the secant method, the derivatives appearing in the matrix $\mathbf{F}'(\theta)$ are evaluated numerically. For this application, the forcing function is simply

$$\mathbf{F}(\theta) = C_p(\theta) - C_p^{target} \quad (10)$$

In matrix form, the method can be expressed as

$$\left| \begin{array}{cccccc} \frac{d\mathbf{F}_1}{d\theta_1} & \cdot & \cdot & \cdot & \cdot & \cdot & \frac{d\mathbf{F}_1}{d\theta_{idim}} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \frac{d\mathbf{F}_i}{d\theta_i} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{d\mathbf{F}_{idim}}{d\theta_1} & \cdot & \cdot & \cdot & \cdot & \cdot & \frac{d\mathbf{F}_{idim}}{d\theta_{idim}} \end{array} \right| \left| \begin{array}{c} \Delta\theta_1 \\ \Delta\theta_2 \\ \cdot \\ \Delta\theta_i \\ \cdot \\ \Delta\theta_{idim-1} \\ \Delta\theta_{idim} \end{array} \right| = - \left| \begin{array}{c} \mathbf{F}_1(\bar{\theta}) \\ \mathbf{F}_2(\bar{\theta}) \\ \cdot \\ \mathbf{F}_i(\bar{\theta}) \\ \cdot \\ \mathbf{F}_{idim-1}(\bar{\theta}) \\ \mathbf{F}_{idim}(\bar{\theta}) \end{array} \right| \quad (11)$$

where, for a typical entry, the notation

$$\frac{d\mathbf{F}_1}{d\theta_1} \quad \text{implies} \quad \left. \frac{\partial \mathbf{F}_1}{\partial \theta_1} \right|_{\theta_2, \theta_3, \dots, \theta_{dim}} \quad (12)$$

The Jacobian matrix was generated using backward differences for each entry.

$$\frac{\partial \mathbf{F}_i}{\partial \theta_i} = \frac{\mathbf{F}_i^k - \mathbf{F}_i^{k-1}}{\theta_i^k - \theta_i^{k-1}} \quad (13)$$

In order to compute the entries in each column, a baseline case was first computed in which all the values of θ were set to their values from the previous iteration. Then, one at a time, the θ 's were varied while the others were held constant, and an entire column was filled. Note that the process requires $n+1$ evaluations of the flow field by the flow solver for each iteration of the secant method. Obviously, this is not a computationally attractive procedure.

When the structure of the Jacobian matrix was examined in this manner, a technique was discovered that generated a substantial savings in CPU time [29]. The perturbed value of θ (by a small amount $\Delta\theta$) only influenced the pressure coefficients of the nearby cells. The influence can be seen for two different transverse locations in Figures 6 and 7, while Figure 5 shows the position of the variables of interest. The variable θ is the angle that the line segment connecting the i^{th} grid point on the current plane with the i^{th} grid point on the previous plane makes with the local horizontal. All the pressure coefficient values are evaluated at the center of the cell. For example, in

Figure 6, perturbing θ_3 only significantly affects the C_p at the adjacent cells that are within a cell or two.

By only taking into account the pressure coefficient fluctuation of a few cells on either side of the perturbed θ , several θ 's may be perturbed at one time, and their corresponding columns in the Jacobian filled. For example, θ_1 , θ_8 and θ_{15} could all be perturbed together by $\Delta\theta$, and columns 1, 8 and 15 filled, since perturbing θ_1 only affected the C_p at cells 1, 2, and 3, and perturbing θ_8 only affected the C_p at cells 6, 7, 8, 9 and 10, and so on. This reduced the number of function calls from $n+1$ to $n/3+1$, a significant cost savings.

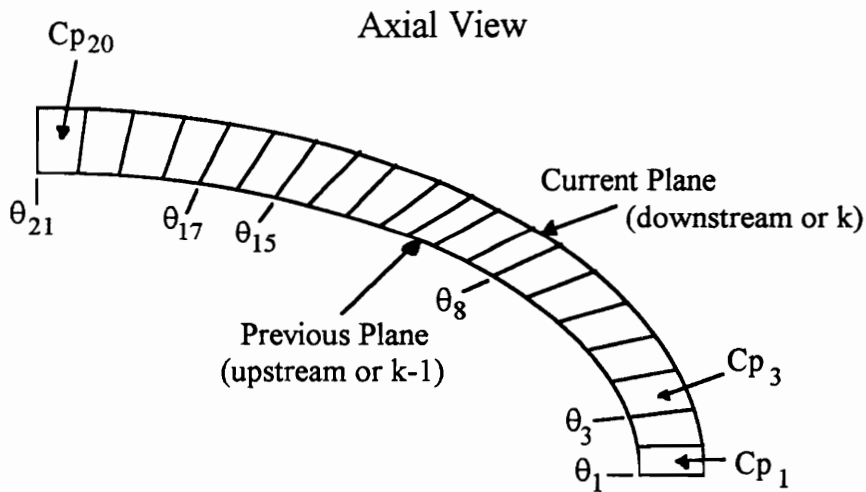


Figure 5. Elliptic Cone Cross-Section

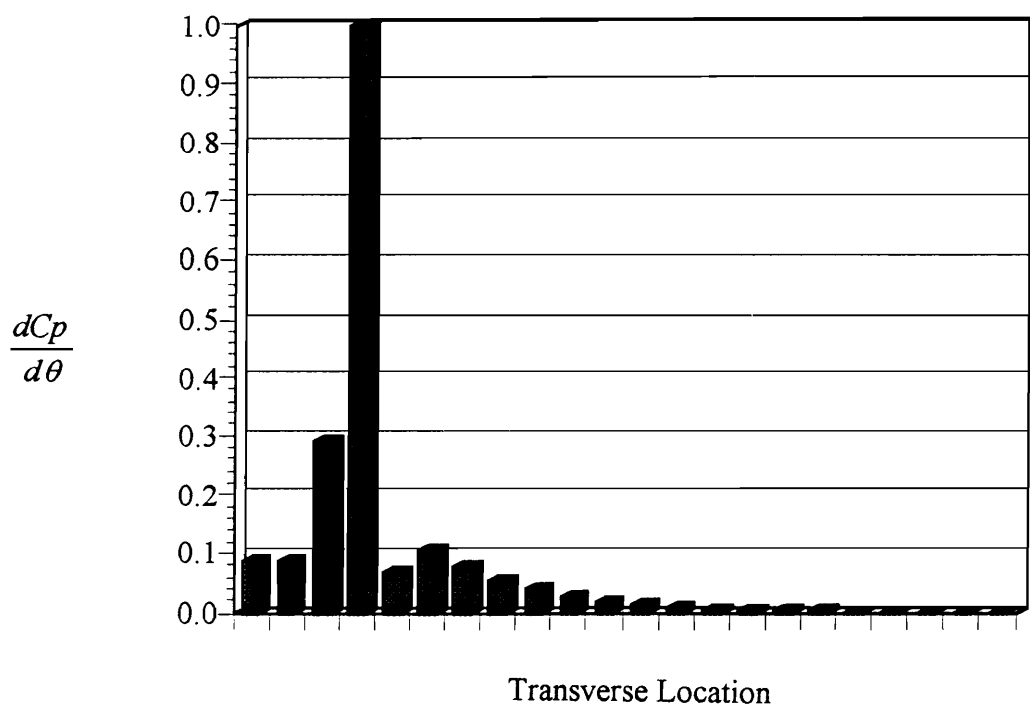


Figure 6. Pressure Coefficient Influence of Perturbing θ_3

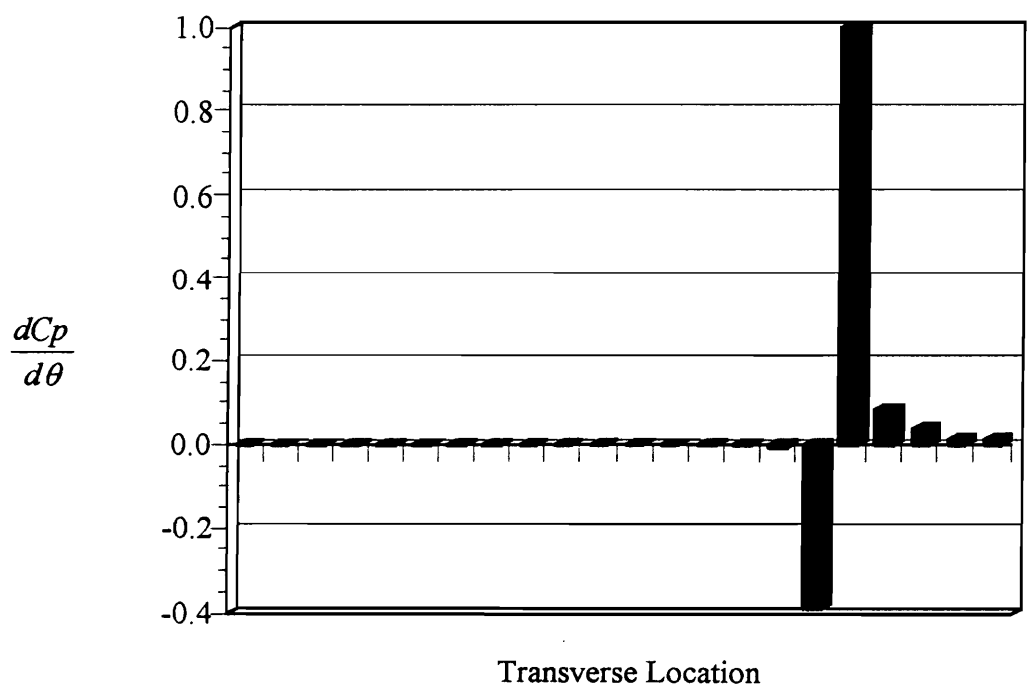


Figure 7. Pressure Coefficient Influence of Perturbing θ_{17}

Although a variety of additional simplifications were attempted, utilizing bi-diagonal, tri-diagonal, and penta-diagonal matrices formed by various averaging methods, few of the attempts succeeded satisfactorily. Most often, oscillations developed in the solution vector, causing instabilities and eventual divergence.

An investigation into the structure of the problem revealed the reason for the instability. To simplify the problem, and remove any possible boundary condition influence, a diamond cross section body was selected for analysis (Figure 8). This reduced the number of parameters to four, a manageable quantity.

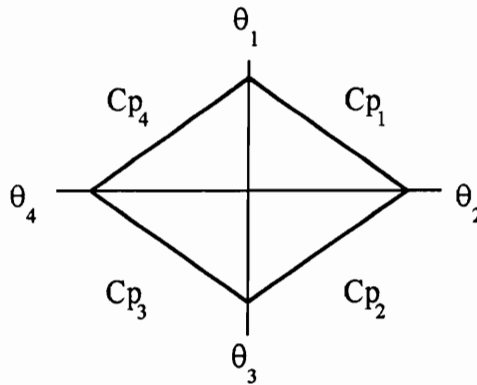


Figure 8. Diamond Cross Section

Upon analysis of the problem using the above procedure, it appeared that there were infinitely many solutions. The matrix $dF/d\theta$ was generated by changing one θ at a time, and the column pertaining to that θ was filled. When the eigenvalues of the $dF/d\theta$ matrix were investigated [30], one of them was shown to be near zero. This led to the conclusion that one of the variables was extraneous, or there were infinitely many solutions. By pinning one of the values, for instance θ_1 , at its known value, the matrix $dF/d\theta$ was no longer indeterminate, and the correct solution could be found. However, this technique is obviously not useful in inverse design, due to the lack of knowledge

about the body. Since infinitely many solutions existed, an attempt was made to find the solution vector of minimum length using a Singular Value Decomposition technique [31]. This effort also failed, as the method was unable to determine the correct solution.

It was then discovered that the manner in which the surface is described in GASP was causing the instability. With structured CFD solvers, it is common practice to define the cell faces on a surface with quadrilaterals, whether or not the four points defining the quadrilateral are co-planar. If the four points do not lie in a plane, then GASP determines the least-square plane for the purpose of computing the unit normal vector. This seemingly minor detail led to the multiple solutions for the problem as shown in Figure 9. The first sketch shows a correct solution to the diamond shaped problem. In this axial view, the four pressure coefficients on the face are matched to the target C_p , and the face between the previous plane and the current plane creates a plane on which the four corner points lie. In the following two views, the pressure coefficients may be the same as those in the first sketch, but the four corner points of the face are non-planar.

Observe that since the geometry in Figure 9 is indeed symmetric about both the x and y axis, the C_p distribution will be the same on each face for a zero angle of attack case. Thus by simple scaling of the geometry, one can match a large number of pressure coefficients within the physically possible set. This barrier led to a search for a new and much more practical solution methodology for the three dimensional problem.

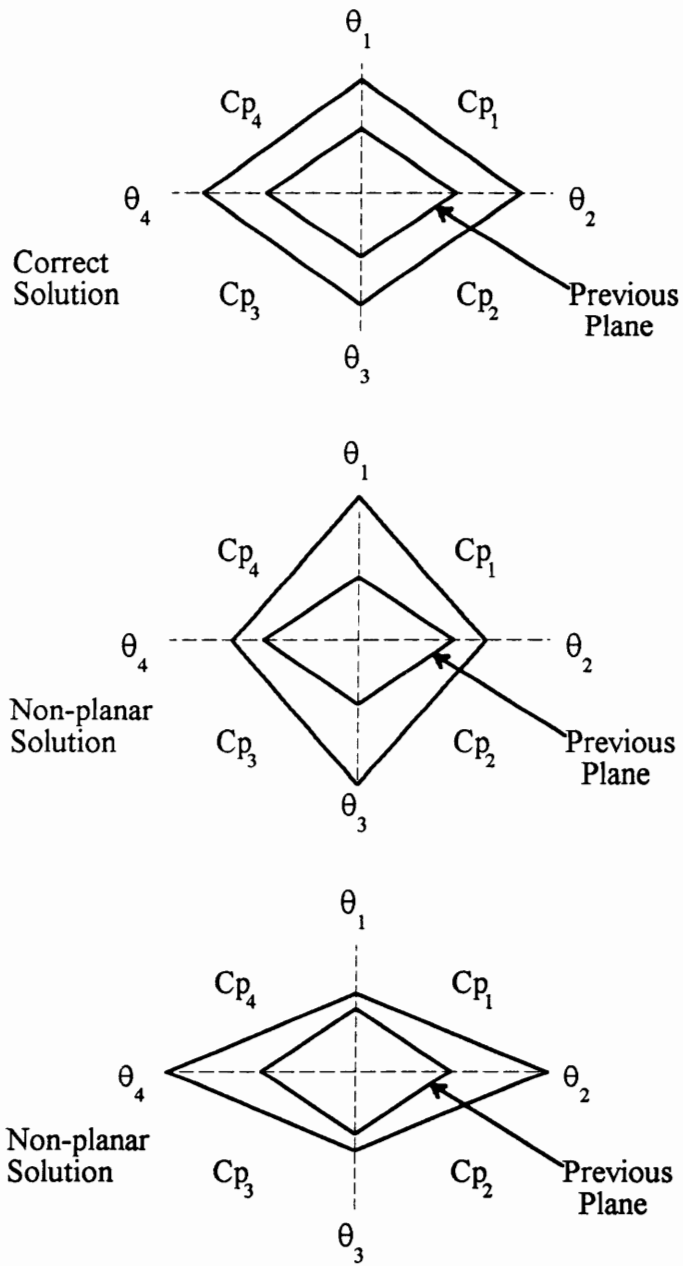


Figure 9. Multiple Solutions for Diamond Cross Section

3.3 One-Parameter Optimization

A different solution approach was taken at this point in the development. Rather than attempt to directly solve for the solution θ_i at each point, the problem was set up to minimize the difference between the target pressure distribution and the pressure distribution generated by the inverse design code using a geometric scaling variable λ .

$$\text{minimize } f(\lambda) \text{ where } f(\lambda) = \sum_i \left(C_{p_i} - C_{p_i}^{target} \right)^2 \quad (14)$$

In certain simple geometric cases, the surface description of the body can be described by a single parameter. The elliptic cone is an example of such a case, since each of the cross sections are similar. This simple example problem was examined before moving on to more general surface descriptions. The diamond case was first re-examined to determine if the methodology was sound. Since the diamond and the elliptic cone are essentially an identical case, different only in the number of grid points used to describe the body, the elliptic cone will be discussed as an example of the single parameter method. Each cross section of the elliptic cone test case are ellipses, therefore λ is simply chosen to be the ratio of the semi-major axis of the ellipse on the current plane to the semi-major axis on the previous plane. To avoid the non-planar issue discussed previously, the ratio of semi-minor axis to semi-major axis was held constant at 1:2.

By observation, in an inviscid flow, if the pressure coefficient on the current plane is greater than the C_p on the previous plane, $\lambda > 1$. To determine the correct λ , a one-dimensional search algorithm was employed. For the elliptic cone test case, a quadratic fit through three points generated a good line search algorithm [32].

A drawback of the quadratic fit method is that three points are needed to start the iteration, and the only information available as a first guess is the result from the previous

plane. Examination of the elliptic cone/power law body shows that $1.0 < \lambda < 2.0$ for planes 4-40. By using λ from the previous plane, and two other choices $\pm \Delta\lambda$, the three starting points for the iterative process were achieved. These points were used to create the quadratic passing through the points,

$$q(\lambda) = \sum_{i=1}^3 f_i \frac{\prod_{j \neq i} (\lambda - \lambda_j)}{\prod_{j \neq i} (\lambda_i - \lambda_j)} \quad (15)$$

and a new point, λ_4 , was determined as the point where the derivative of the quadratic goes to zero [32],

$$\lambda_4 = \frac{1}{2} \frac{b_{23}f_1 + b_{31}f_2 + b_{12}f_3}{a_{23}f_1 + a_{31}f_2 + a_{12}f_3} \quad (16)$$

where

$$a_{ij} = \lambda_i - \lambda_j \quad \text{and} \quad b_{ij} = \lambda_i^2 - \lambda_j^2 \quad (17)$$

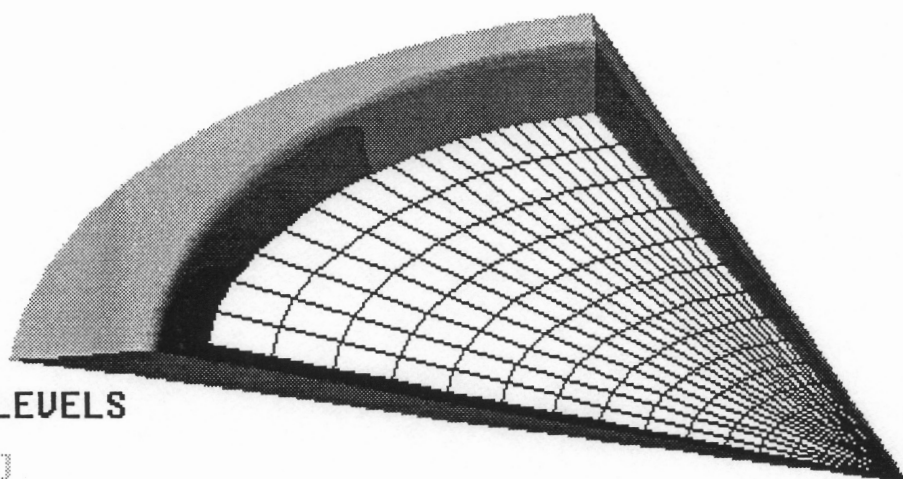
This method converged rather slowly, as the quadratic was fairly flat for the elliptic cone/power law body, but a solution was found for every plane; the target and inverse design pressure contours are shown in Figure 10. Figure 11 shows the contours of the difference between the target pressure coefficient and the inverse design solution pressure coefficient. Two views are given, the first is a view of the $i=21$ plane, or the computational grid in the x - y plane. The second plot is the pressure coefficient difference contours in the y - z plane, at the $k=21$ location, or the aft end of the body. In each plot, the

difference between the target C_p and the inverse design C_p is on the order of 10^{-5} at most points in the flowfield. Most of the larger differences take place near the location of the shock wave.

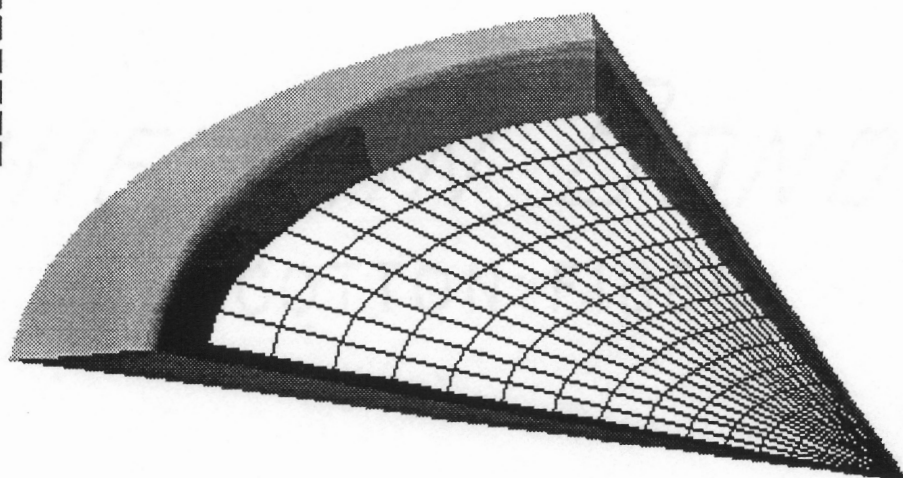
The convergence of the optimization could possibly have been improved, but more challenging problems lay ahead, and little effort was expended on the single parameter optimization. Moreover, the single parameter optimization is not very general since only bodies of similar cross section at each axial plane can be solved using this method, or similar problems where the surface geometry can be adequately described by a single parameter. To perform the multi-parameter optimization, an existing optimization program was utilized.

CONTOUR LEVELS

1000.000
2000.000
3000.000
4000.000
5000.000
6000.000
7000.000
8000.000
9000.000
10000.00
11000.00



Target Pressure Contours



Inverse Solution Pressure Contours

Figure 10. Elliptic Cone Target and Inverse Design Pressure Contours

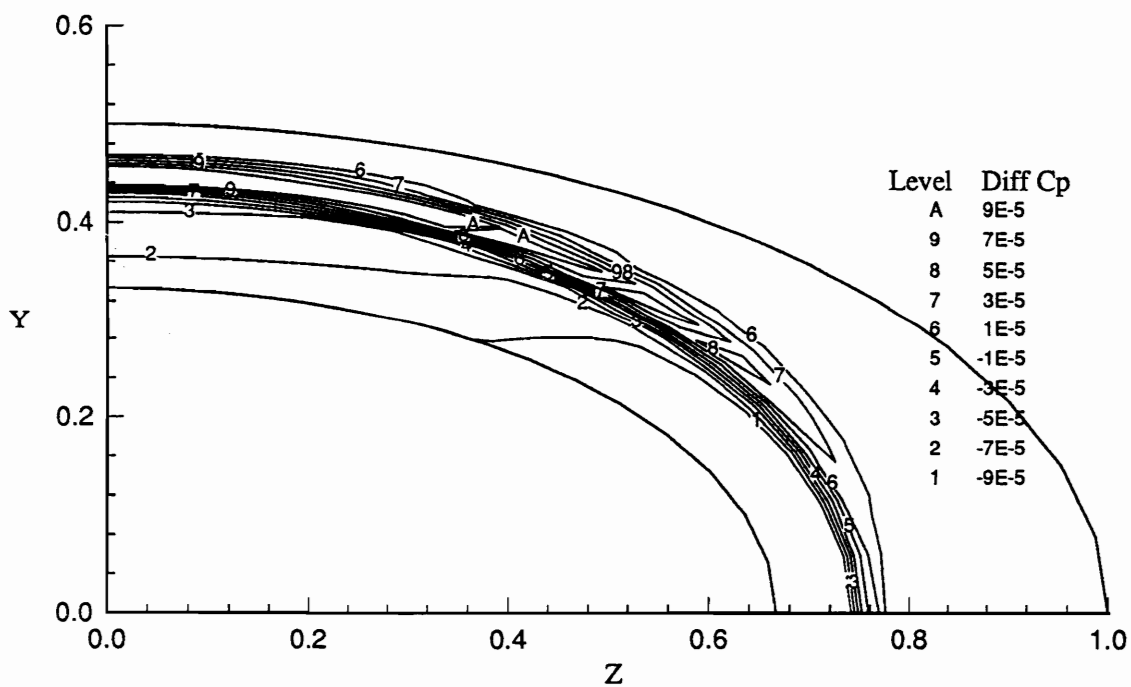
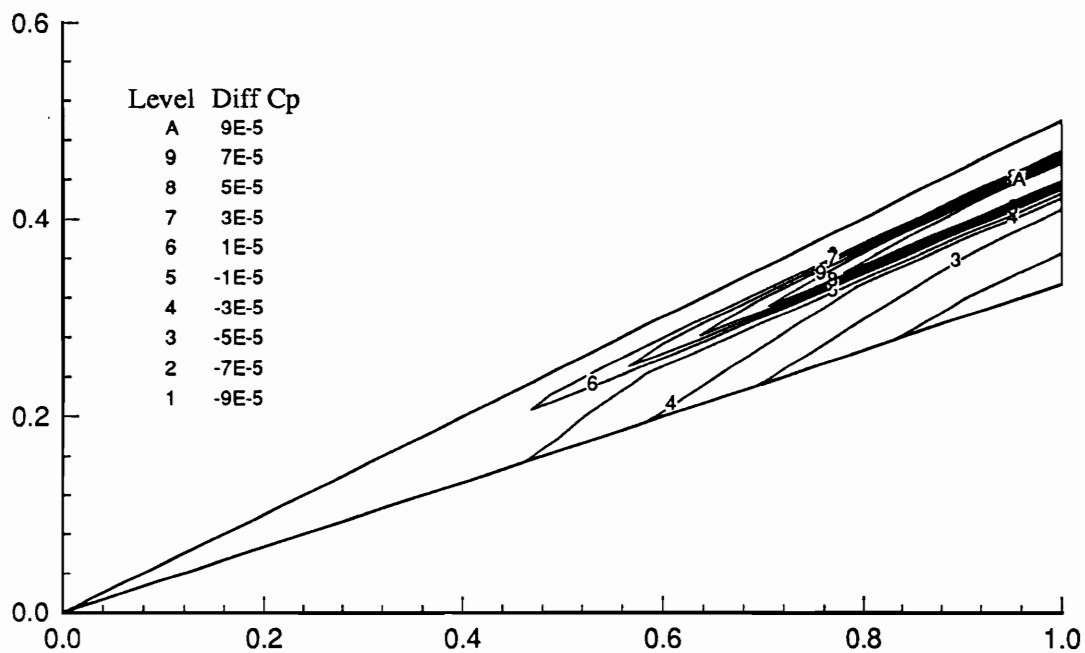


Figure 11. Elliptic Cone Pressure Coefficient Difference Contours

4. QUASI-NEWTON OPTIMIZATION METHOD

To perform the multi-parameter optimization required for a more general inverse design method, the Fortran subroutine NLPQL, due to Schittkowski, was utilized [33]. In general, multi-parameter optimization problems encountered in this work were unconstrained, and did not take advantage of the full usefulness of the code. Without constraints, the method reduced to a Quasi-Newton method.

A Quasi-Newton method is based on the fact that an approximation to the curvature of a non-linear function can be computed without explicitly forming the Hessian matrix [34]. This method is an improvement over the steepest-descent algorithm, which takes a step in the direction of the negative gradient, but exhibits poor convergence. If the Hessian matrix was available, a step in the Newton direction would yield the best convergence, this is known as Newton's method. A discrete approximation to the Hessian would be time consuming, since the function evaluation routine, the flow solver, consumes the majority of the CPU time in inverse design problems. Therefore, for these unconstrained cases, the Quasi-Newton method, which forms an approximation to the Hessian based on information from previous iterations, requires only first derivative information and converges more quickly than a steepest descent algorithm. The first derivative of the objective function with respect to the design parameters is performed through central differencing.

Although no modeling constraints were placed on the problem, upper and lower bounds were necessarily placed on the parameters in the optimization space, due to geometric constraints. For example, if a supersonic flowfield around a relatively smooth body is being solved, and the next iteration produces a large change in the body geometry, the flow solver may have difficulty converging to an appropriate solution on that plane

(Figure 12). This figure was generated by changing the surface characteristics of one plane in the streamwise direction of an otherwise smooth body. At supersonic velocities, most flow solving programs would have difficulty finding a reasonable solution to this shape.

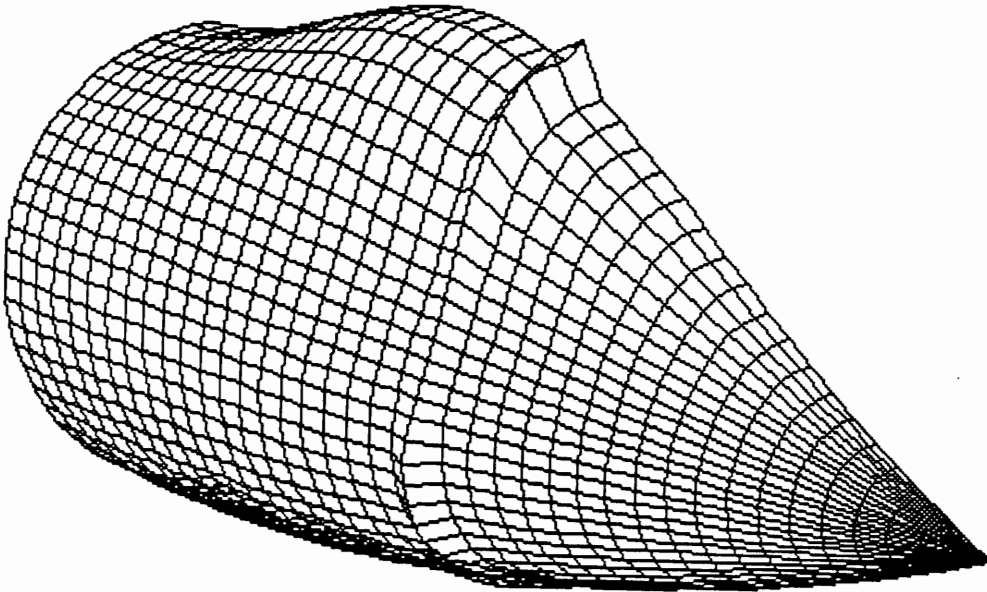


Figure 12. Inappropriate Iterative Guess

This became a problem when performing the supersonic multi-parameter optimization. Several "trust-region" techniques exist in which the step size for the next iteration can be scaled to remain in a region that the user "trusts" will give a reasonable solution, but none were implemented into this effort [35]. Instead, the upper and lower

bounds on the optimized variables were adjusted to put limitations on the iteration step size. This technique was successful, but time consuming.

The NLPQL algorithm is general enough to handle most types of non-linear optimization problems involving inverse design. An example of a problem that might take advantage of all the capabilities of the optimizer might be the inverse design of a wing section, where constraints are placed on the root section, yet improved aerodynamic performance is desired based on changing other wing cross sections.

5. ANALYTIC FOREBODY INVERSE DESIGN

5.1 Analytic Forebody Grid Generation

In 1979 NASA conducted a series of tests on various forebody configurations that could be described analytically [36]. The body selected for this computational procedure is described by the following equations:

$$\frac{r_1(x)}{l} = \left(1 - \frac{x}{2l}\right) \left(\frac{x}{l}\right) \tan(20^\circ) \quad (18)$$

$$\frac{r}{r_1} = \frac{1}{\lambda_1 - \lambda_2 \cos(2\theta) + \lambda_3 (\cos\theta - \cos 3\theta)} \quad (19)$$

$$\frac{y}{r_1} = \left[1 + \left(1.35 \frac{r}{r_1} - 1 \right) \sin^2 \left(\frac{\pi x}{l} \right) \right] \sin \theta \quad (20)$$

$$\frac{z}{r_1} = \left[1 + \left(1.35 \frac{r}{r_1} - 1 + \frac{0.35}{\cos\theta} \right) \sin^2 \left(\frac{\pi x}{l} \right) \right] \cos \theta \quad (21)$$

where $\lambda_1=1.25$, $\lambda_2=0.25$ and $\lambda_3=0.13174$. Figure 13 shows the body generated by these equations. The body is a generic description of a forebody and canopy of a supersonic fighter aircraft.

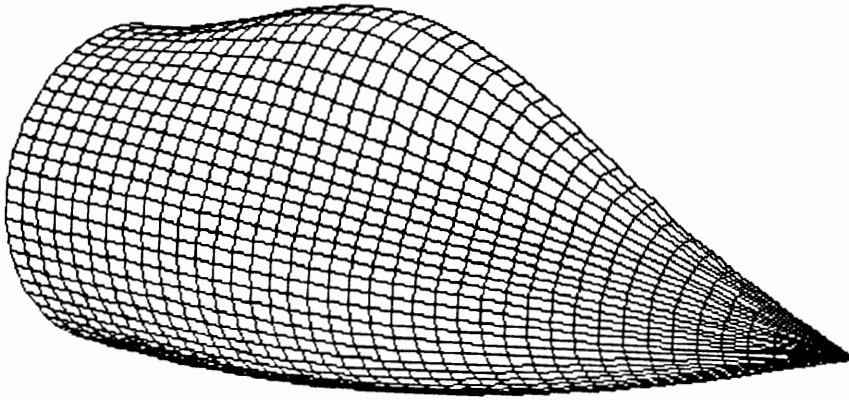


Figure 13. Analytic Forebody Baseline Grid

5.2 Comparison with Test Results

A baseline perfect gas, inviscid case at Mach 1.70 was computed with GASP. Since the inviscid problem is assumed to be supersonic everywhere, the Euler equations were solved, allowing the problem to be marched in the streamwise direction, one plane at a time. Figure 14 shows the comparison of the transverse distribution of the pressure coefficient from the GASP run and the NASA Langley test case.

The solution and the test data are compared at three different streamwise locations on the body, and the agreement is reasonable. The test conditions listed in Table 5 were matched for the GASP calculations, except of course the Reynolds number, which is meaningless for an inviscid calculation.

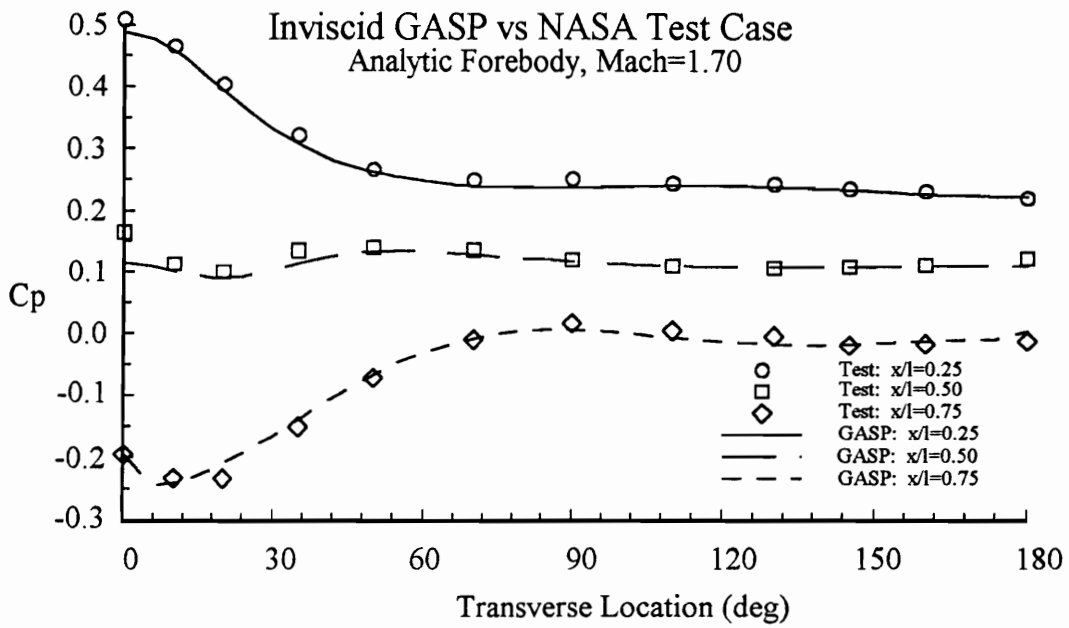


Figure 14. Inviscid GASP Calculations vs NASA Test Case

Table 5. Analytic Forebody Test Conditions

Reynolds Number	2.33×10^6 1/ft
Total Temperature	339 K
Total Pressure	56.403 kPa
Dynamic Pressure $_{\infty}$	23.126 kPa
Static Pressure $_{\infty}$	11.443 kPa
Model Length	14 inches

Although the GASP results did not exactly match the test results, this did not affect the inverse calculations, since the inverse design procedure had as its target the

baseline CFD solution. That is to say, the experiment was only used to verify the baseline solution, it was not used in the actual design method.

5.3 Three Dimensional Inverse Design - Inviscid

A simple three parameter unconstrained optimization was performed on the analytic forebody to test the ability of the NLPQL optimizer to work with the flow solver. The three parameters used in this case were λ_1 , λ_2 and λ_3 as defined in equation 19. The effect of changing each of these parameters can be seen in Figure 15. Changing λ_1 smoothes the top surface and places a hump on the bottom of the body, perturbing λ_2 increases the size of the canopy on the top, and varying λ_3 fills out the bulge on the side of the body. To begin the inverse design process, the entire baseline body was generated analytically, and the target pressure coefficients were computed using GASP.

When space marching in GASP, information from the previous plane is used to initialize the solution on the current plane. For this calculation, the first three planes were computed from the baseline body, and the fourth plane was used to test the optimization techniques. Practical limits based on body geometry were used as upper and lower bounds on the three parameters; Table 6 lists the baseline value, the starting value, and the upper and lower limits.

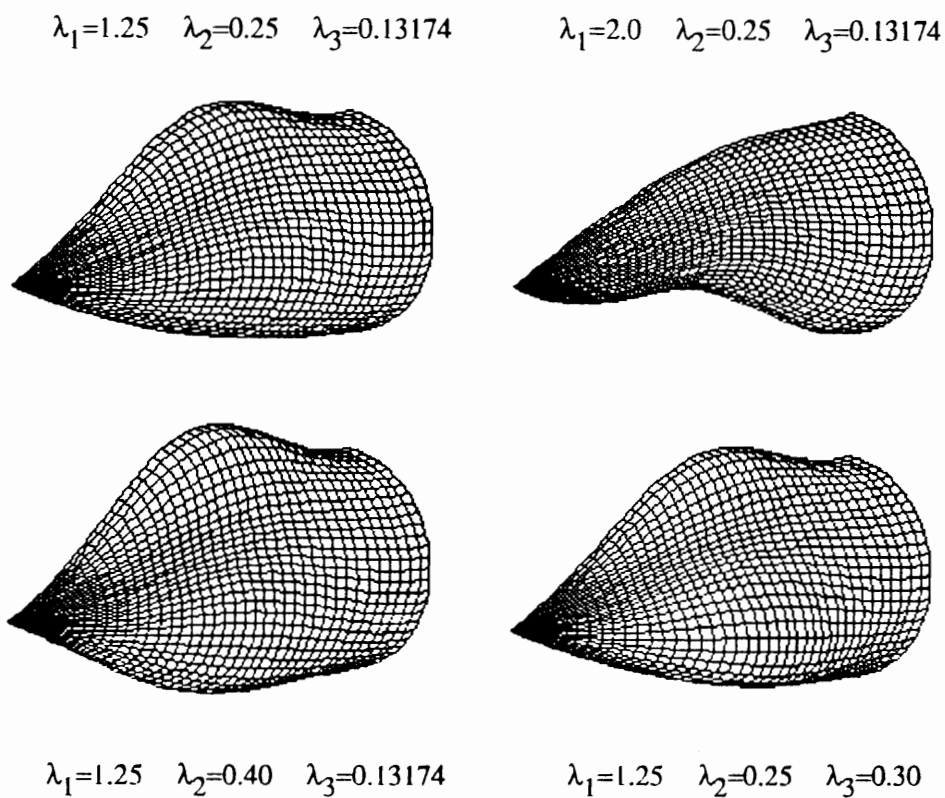


Figure 15. Parameter Variation Effect on Baseline Analytic Forebody

Table 6. Parameter Values and Bounds for Three Parameter Optimization

Parameter	Lower Bound	Baseline Value	Starting Value	Upper Bound
λ_1	1.0	1.25	1.5	2.0
λ_2	0.1	0.25	0.3	0.5
λ_3	0.1	0.13174	0.2	0.3

The optimization process was then started, with the objective function

$$F(C_p, C_p^{target}) = \sum_{i=1}^{idim} \left(C_{p_i} - C_{p_i}^{target} \right)^2 \quad (22)$$

where i is in the transverse direction around the body. This is a simplification of the real problem to be solved

$$F(C_p, C_p^{target}) = \sum_{k=1}^{kdim} \sum_{i=1}^{idim} \left(C_{p_i} - C_{p_i}^{target} \right)^2 \quad (23)$$

which would minimize the difference between the target C_p and the inverse design C_p at each point on the entire body.

Each time a parameter was changed, a new grid for that plane was generated, based on the current parameters. The optimization routine called the flow solver, and generated gradients based on central differences. The gradient was then used to calculate the direction in which to search for the optimal set of variables. The NLPQL algorithm employs an Armijo type bisection method combined with quadratic interpolation to perform a line search along the direction specified by the gradient to produce a sufficient decrease of a merit function based on the Lagrange multipliers [33]. The convergence history of the 3-parameter inverse design process is shown in Figure 16. Note that each design iteration requires a minimum of seven function evaluations (one baseline + two gradient calculations for each parameter). More function evaluations may be required in the line search algorithm.

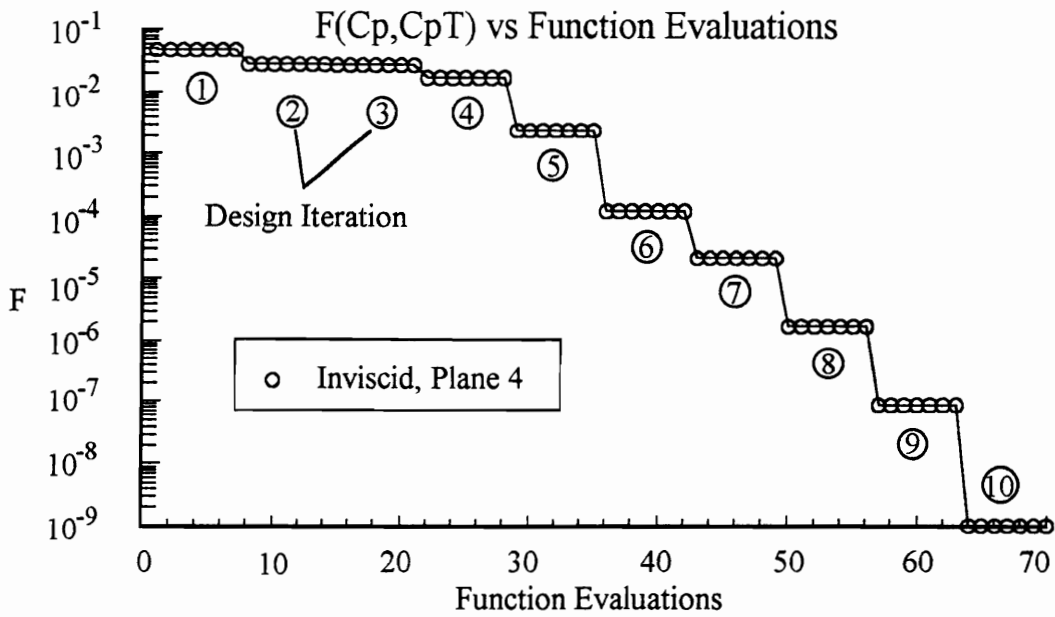


Figure 16. Inviscid Three Parameter Optimization Results

It appeared pointless to run the same calculation on any more planes since in a real design process the parameters from the previous plane would be chosen as the starting point for the next plane, from which immediate convergence would result for this case, due to the nature of the parameters chosen and the analytic description of the body. The variation of each of the three parameters can be seen in Figure 17. Most of the changes occur by the sixth design iteration, near the point where the objective function has been reduced by four orders of magnitude. In performing further work on this body, the solutions from the optimizer were only converged four orders of magnitude.

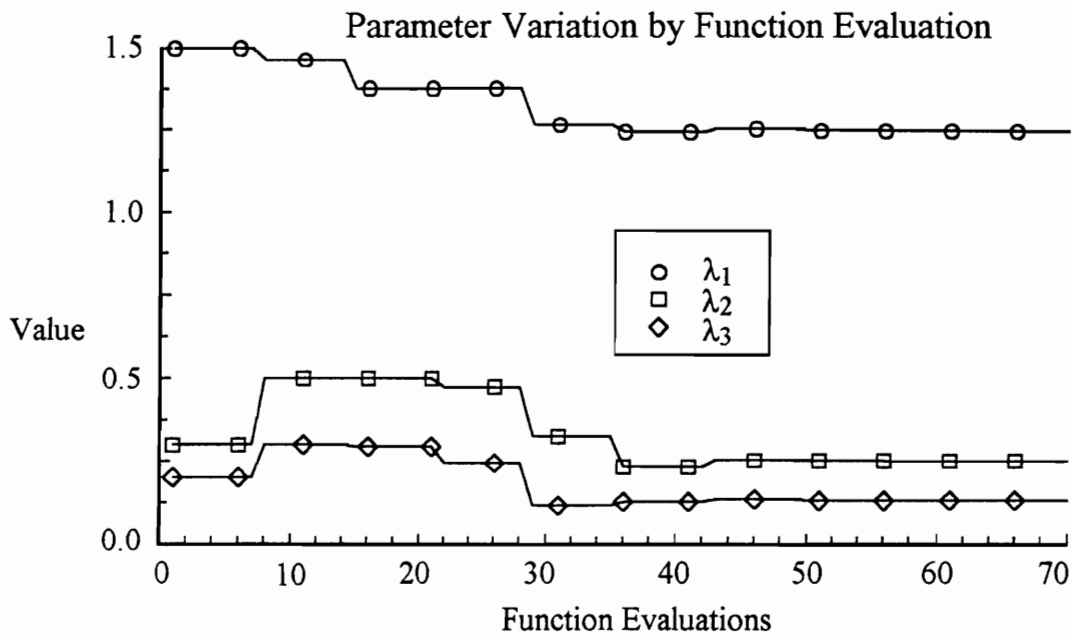


Figure 17. Parameter Variation by Iteration

Once it was determined that the optimizer was performing well with GASP, a more general approach to the surface description was sought. As discussed in Chapter 1, B-spline curves are particularly well suited to general problems of this nature, and are of particular interest to the aerospace designer.

6. SURFACE DESCRIPTION USING B-SPLINES

6.1 B-Spline Curves

B-spline curves are a well known method of describing generalized curves and surfaces that have a small number of parameters but a large amount of shape control. B-splines apply a set of blending functions to $n+1$ control points \mathbf{p}_i by

$$\mathbf{p}(u) = \sum_{i=0}^n \mathbf{p}_i N_{i,k}(u) \quad (24)$$

The parameter k controls the degree of the polynomial (degree = $k-1$) of the line segment between each of the control points [37]. The blending functions can be computed recursively from

$$\begin{aligned} N_{i,1}(u) &= 1 \text{ if } t_i \leq u < t_{i+1} \\ N_{i,1}(u) &= 0 \text{ otherwise} \end{aligned} \quad (25)$$

and

$$N_{i,k}(u) = \frac{(u - t_i)N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u)N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}} \quad (26)$$

The t_i are called knots and relate a parametric variable u to the control points. For a curve that is not a closed figure, and has the end points fixed at the end control points, the knot values must be chosen to be

$$\begin{aligned}
t_i &= 0 & \text{if } i < k \\
t_i &= i - k + 1 & \text{if } k \leq i \leq n \\
t_i &= n - k + 2 & \text{if } i > n
\end{aligned} \tag{27}$$

where

$$0 \leq i \leq n + k \tag{28}$$

and the parametric variable u is defined by

$$0 \leq u \leq n - k + 2 \tag{29}$$

An example of the blending functions for $n=5$ (six control points) and $k=4$ (cubic line segments) is shown in Figure 18.

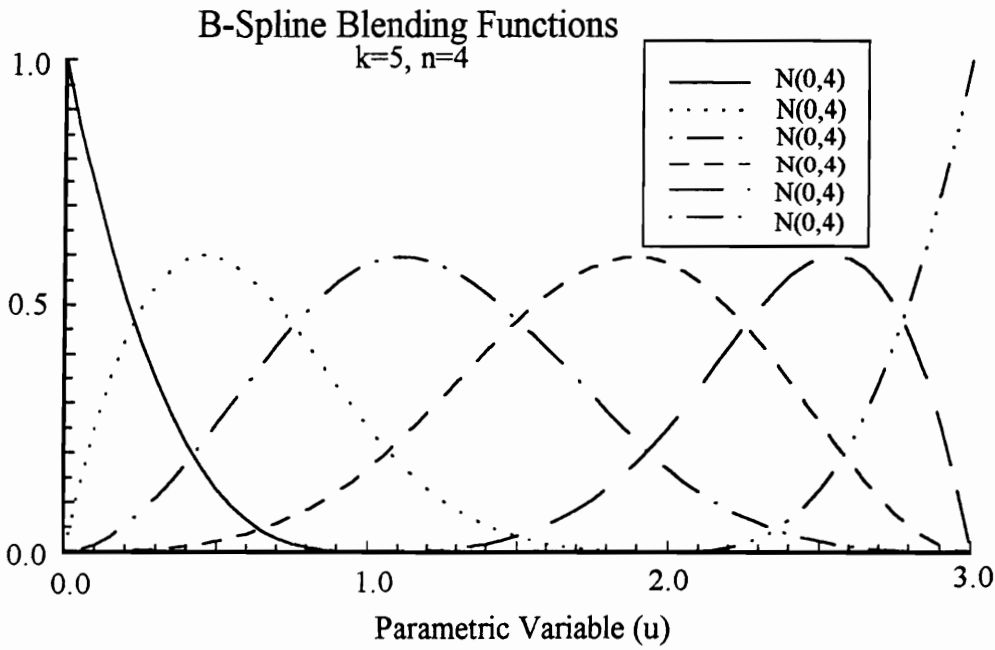


Figure 18. B-Spline Blending Functions

An example of a four control point ($n=3$) cubic B-Spline is shown in Figure 19. Note the relationship between the control points p_i and the shape of the curve. As the control points are moved further from the curve, the B-spline is stretched accordingly. The curve also passes through the end points, due to the choice of knot values.

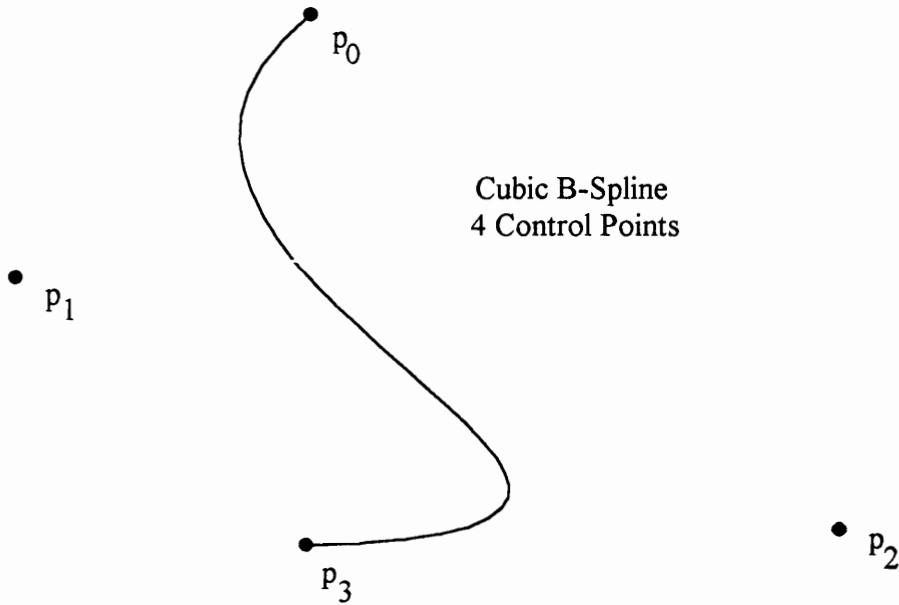


Figure 19. B-Spline Example

6.2 Non-Uniform Rational B-Splines

Non-Uniform Rational B-Splines (NURBS) are currently in use throughout the aerospace industry as Computer-Aided Design tools and as surface generation techniques. NURBS are easily generated from the above formulation by changing equation 24 to

$$\mathbf{p}(u) = \frac{\sum_{i=0}^n h_i \mathbf{p}_i N_{i,k}(u)}{\sum_{i=0}^n h_i N_{i,k}(u)} \quad (30)$$

where the variables h_i are weights that are associated with each control point. The standard B-spline derivation can be recovered by setting the weights equal to one, since the blending functions sum to one identically. The NURB can be manipulated by either moving the control points, or modifying the weight associated with the control point. Increasing the weight tends to pull the B-spline curve toward that control point.

NURBS have the added flexibility of being able to exactly describe a quadratic or cubic curve, whereas general B-splines can only approximate these polynomial functions. This description was not intended to include a discussion on the trade-offs associated with using the different B-splines. Although in principle this inverse design method could handle the introduction of NURBS, they were not considered in this work due to the increased number of parameters, the weights, associated with the B-splines. Since an increase in the number of parameters leads to an increase in CPU time and problem complexity, NURBS are mentioned here only for completeness and because of their regular usage in current CAD communities.

6.3 Three Dimensional B-Spline Surfaces

The formulation of a B-spline surface also follows directly from the B-spline curve generation [37]. The shape of the B-spline surface approximates the polyhedron formed by its vertices \mathbf{p}_i , where

$$\mathbf{p}(u, w) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{p}_{i,j} N_{i,k}(u) N_{j,l}(w) \quad (31)$$

as in equation 24. The $N_{i,k}(u)$ and $N_{j,l}(w)$ are the blending functions as before, where k and l are the degree of the polynomials. These B-spline surfaces could be used in the inverse design procedure to generate an entire three-dimensional surface and optimize on its shape by modifying the positions of the control points, $\mathbf{p}_{i,j}$.

7. ELLIPTIC MULTI-PARAMETER B-SPLINE INVERSE DESIGN

7.1 Circular Arc Airfoil Test Case

All the calculations to this point have utilized the space marching mode of GASP. This mode is only useful in design cases that involve supersonic velocities in the inviscid region, so that information passes from one streamwise plane to the next in only one direction. For subsonic cases, a global strategy must be used to solve elliptic problems. A simple two dimensional test case was chosen to demonstrate the usefulness of the inverse design code for subsonic, inviscid applications.

The inviscid flow of a perfect gas over a circular arc airfoil in a channel was analyzed at Mach 0.3 (Figure 20).

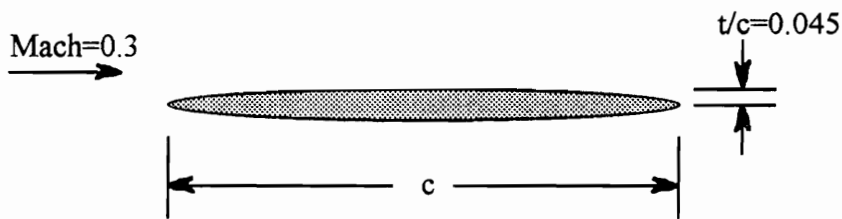


Figure 20. Circular Arc Airfoil

The parameters for the inviscid calculations are shown in Table 7 [2]. Two grid points are required in the direction orthogonal to the streamwise and normal directions due to the fact that GASP uses a finite volume formulation. The inflow and outflow boundary conditions, $i0$ and $idim$, use a subsonic boundary condition where the flow quantities are computed as a function of freestream quantities and interior values. Riemann invariants are used to properly extract information at the boundaries [2]. The $jdim$ boundary also

uses this farfield formulation. The $j0$ boundary assumes no flow normal to the surface of the airfoil or the channel.

Table 7. Inviscid Circular Arc Input Parameters

idim (streamwise direction)	61
jdim (direction normal to body)	31
kdim	2
i0 Boundary Condition	subsonic inflow
idim Boundary Condition	subsonic outflow
j0 Boundary Condition	tangency
jdim Boundary Condition	subsonic
k0 Boundary Condition	First order extrapolation
kdim Boundary Condition	First order extrapolation

The 61x31 grid is shown in Figure 21. The grid is clustered near the leading and trailing edge of the airfoil to resolve the physics of the problem. Since this case is inviscid, there is no clustering of the grid in the direction normal to the wall.

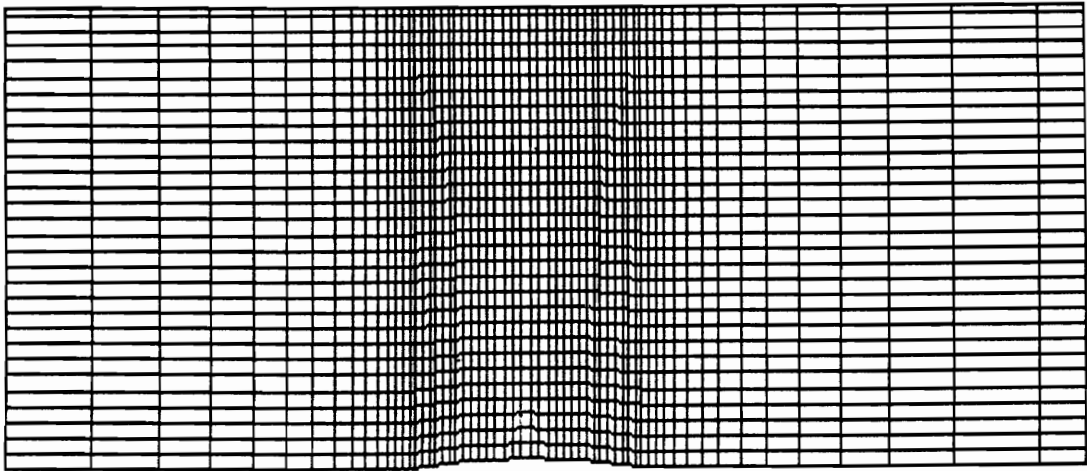


Figure 21. Grid for the Circular Arc Test Case

The circular arc can be adequately described by a B-spline with 5 control points ($n=4$). Two of the control points were fixed at the leading and trailing edges, the other three were allowed to float in the radial direction, and the second and fourth points were allowed to move in the angular direction. This yielded a total number of optimization parameters of five. Each of the spline segments is a 4th order polynomial ($k=5$).

An interpolation routine was needed to ensure that the objective function

$$f = \sum_{x=0}^1 \left(C_{p_x} - C_{p_x}^{target} \right)^2 \quad (32)$$

was comparing pressure coefficients at similar positions. The leading and trailing edges of the airfoil are at $x=0$ and $x=1$, respectively, and C_{p_x} denotes the pressure coefficient on the airfoil surface at the corresponding x location. The B-spline generated curve was interpolated to the x location of the target shape for this purpose. In an actual design implementation, the pressure coefficients generated by the inverse design code would be interpolated to the location at which the target data was given in a similar manner.

Contours from the baseline pressure calculation for the circular arc problem are shown in Figure 21. The pressure contours near the leading and trailing edges are not very smooth, because the tangency boundary condition does not adequately model the flow at this low Mach number. The version of GASP that was employed in running this test case did not include the Dadone-Grossman curvature correction boundary condition technique [39] that Godfrey [40] has shown to smooth the pressure contours.

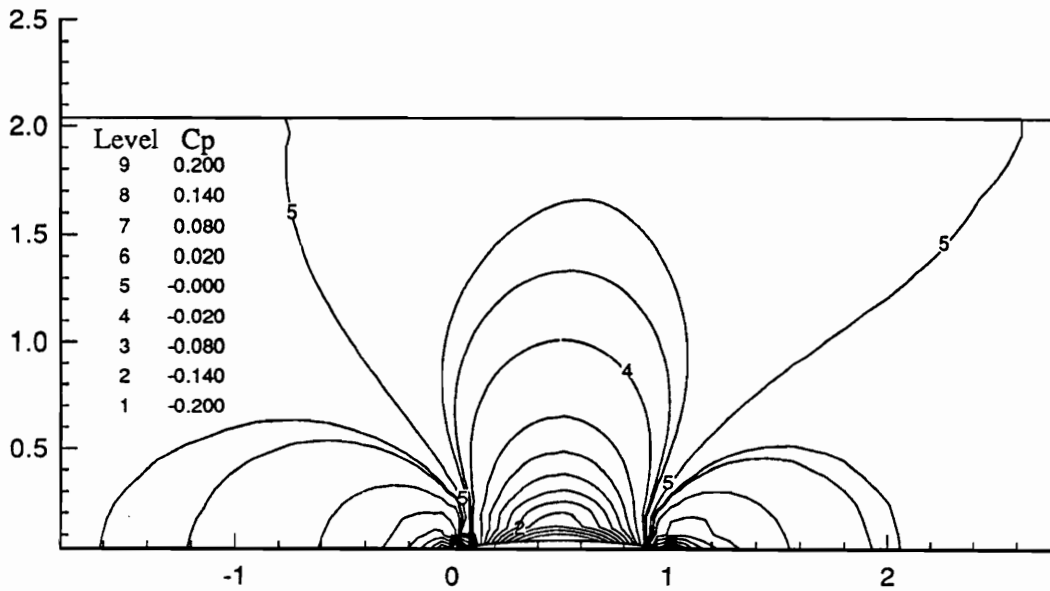


Figure 22. Circular Arc Pressure Contours

The initial approximation to the circular arc airfoil was a B-spline generated curve that was intentionally designed to be asymmetric and have a t/c ratio of 0.7, more than 1.5 times the t/c ratio of the target body, to determine how well the code could converge from an initial solution that was far away from the target body. The objective function for the optimizer, equation 32, compares the C_p on the B-spline generated surface with the C_p on the target body. It does not take the pressure distribution on the wall of the channel into account in the objective function.

7.2 Circular Arc Airfoil Inverse Design

The inverse design process was started as before, and the initial B-spline and pressure distribution were recorded. For each iteration, the optimizer took a central derivative of each parameter (the location of the B-spline control points) to get a gradient direction. A step size was computed in this direction and a new body surface based on the B-spline was generated. A new grid was then generated and GASP was utilized to get the

pressure coefficients and the objective function. The objective function and cumulative CPU time on the Cray Y-MP C-90 are listed in Table 8. After three iterations, the B-spline and pressure distribution were examined, and the objective function had been reduced significantly, although the optimization algorithm had overshoot the target solution, as seen in Figure 24.

Table 8. Circular Arc Airfoil Inverse Design

Iteration	Objective Function	Gradient ²	Cumulative CPU seconds
0	0.518	N/A	4.5
3	0.0809	2.0	214.5
13	0.00037	8.4x10 ⁻⁶	748.5
23	0.00033	1.8x10 ⁻⁵	1285.5

After ten additional iterations were performed, the pressure distribution was almost indistinguishable from the target Cp and the B-spline approximation is coincident with the target surface (Figure 23). Ten more iterations were performed to see if the solution could be improved. As seen in Table 8, the solution is not significantly improved, and the L2 Norm of the gradient

$$\|gradient\|^2 = \nabla F^2 \tag{33}$$

actually increased, where

$$\nabla F = \left| \begin{array}{cccc} \frac{\partial F}{\partial \lambda_1} & \frac{\partial F}{\partial \lambda_2} & \cdots & \frac{\partial F}{\partial \lambda_n} \end{array} \right| \quad (34)$$

and n is the total number of parameters in the optimization scheme. Since the objective of the optimizer was to find a minimum, this should occur where the gradient is near zero. This led to the conclusion that the solution after 13 iterations was the best that could be achieved with the number of parameters and the degree of the B-spline that was used.

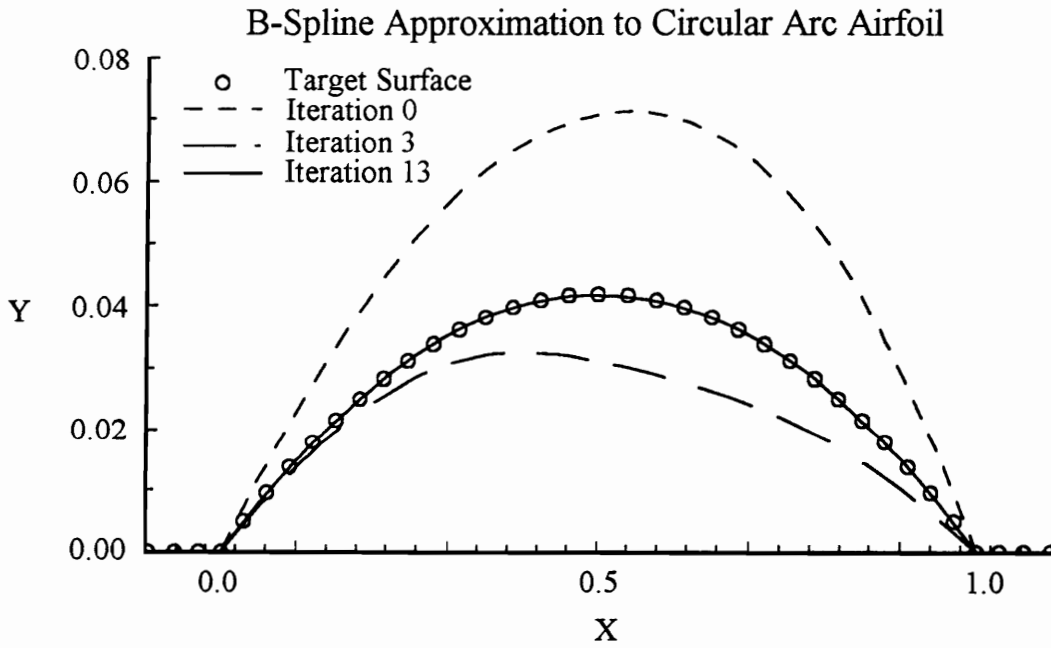


Figure 23. B-Spline Approximation to the Circular Arc Airfoil

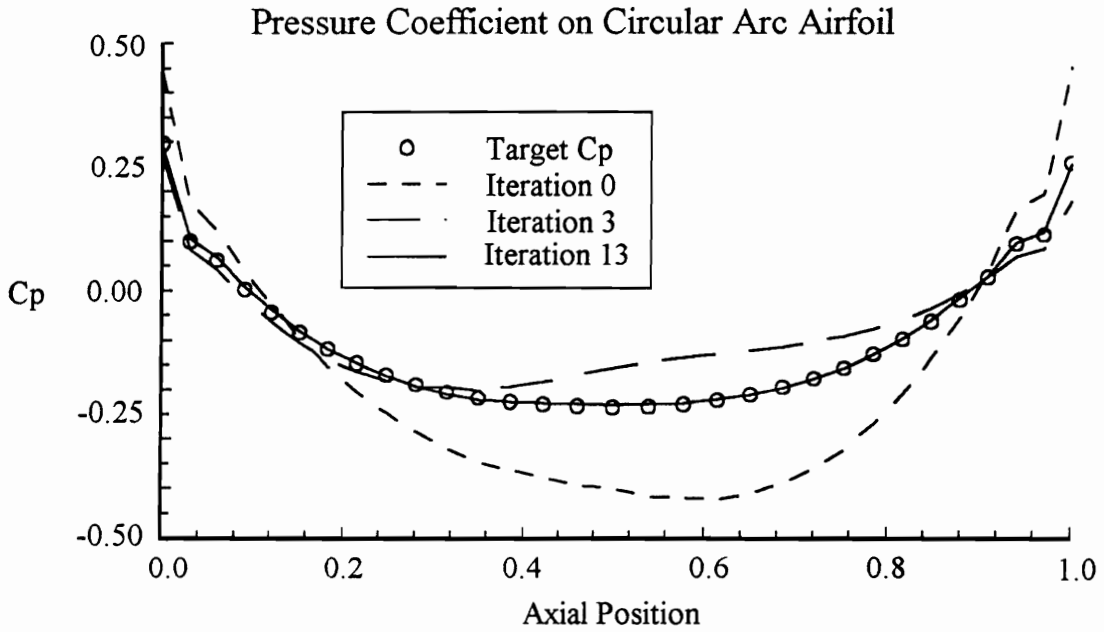


Figure 24. Pressure Distribution on the Circular Arc Airfoil

The inverse design method was quite robust running this subsonic test case. The upper and lower bounds on the optimization parameters covered a wide range, and the optimizer did not reach any intermediate solutions that were up against a parameter boundary. This makes the procedure much easier for the designer, since very little user interaction is required, other than the input of an initial guess.

This test case was only intended to show the ability of the procedure to effectively solve an elliptic test case. Many inverse design methods are available that are much more computationally efficient for subsonic airfoils in an inviscid flow. However, many of them have limitations on finding the inverse solution, because the speed on the surface of the airfoil is a parameter and viscous computations are not achievable [3], [5]. Therefore, while this procedure may not be the most efficient for this particular problem, the flexibility of the tool makes it attractive.

8. MULTI-PARAMETER B-SPLINE INVERSE DESIGN

8.1 Multi-Parameter B-Spline Curve Generation

The analytic forebody was again studied for the full three dimensional B-spline analysis. In general, the body surface at each plane was simulated by a curve generated with a B-spline, and the parameters of the B-spline were optimized to meet the objective function. This objective function was again

$$f = \sum_{i=1}^{idim} (C_{p_i} - C_{p_i}^{target})^2 \quad (35)$$

where C_p is the pressure coefficient generated at each cell center in a transverse, or i , direction around the body. The actual problem that should be solved is to minimize the objective function, equation 35, at all the planes on the body. A simplification was made by space marching the body and minimizing the objective function a plane at a time. This simplification resulted in a major reduction in CPU time.

Because the B-spline curve often generated grid points that were in a different transverse location than those on the target body, an interpolation routine was used to ensure that all the grid points fell on rays emanating from the origin at evenly spaced angles (Figure 25). Because the original grid on the left hand side of Figure 25 has rays that do not emanate from the origin, when comparing the pressure coefficients generated by the inverse design procedure with the target pressure coefficient the i^{th} face may not be at the same location and the objective function may produce false results. Therefore, both the target grid and the B-spline generated grid (at each iteration) were interpolated to the evenly spaced cross section as shown on the right hand side of Figure 25. This

interpolation routine used a simple arithmetic average between the nearest neighboring cells on the target body to compute a new radius for the B-spline generated curve.

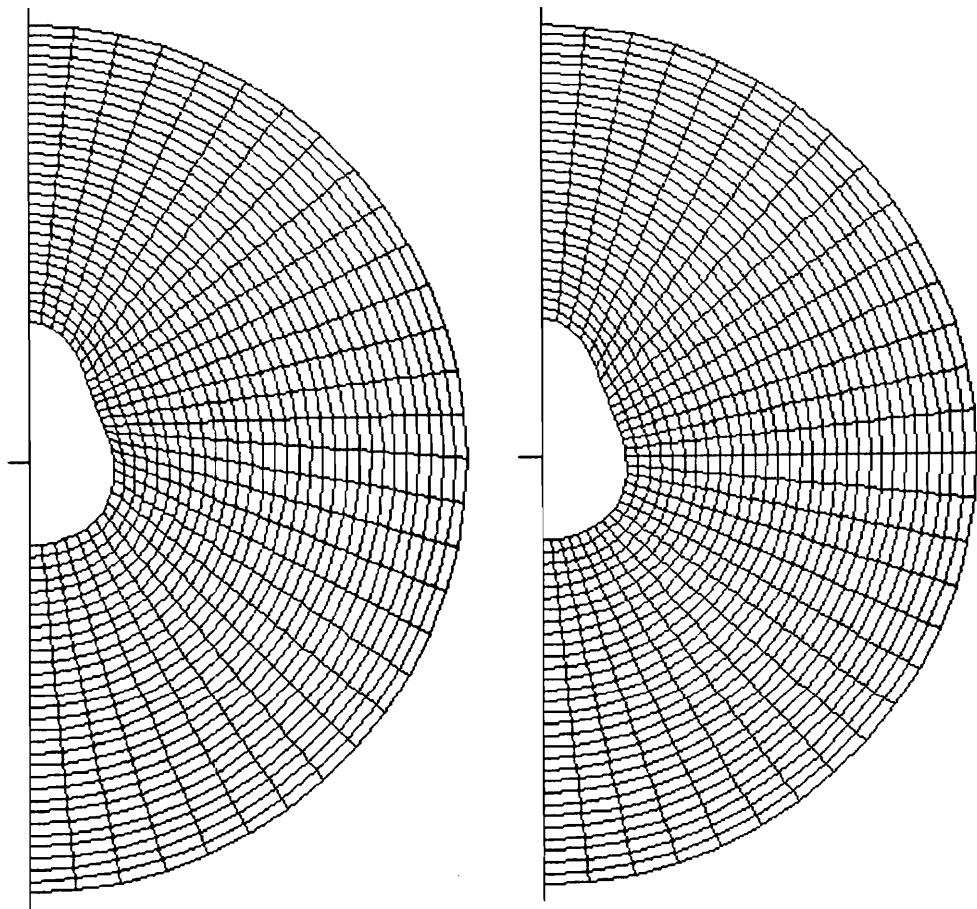


Figure 25. Grid Interpolation to Evenly Spaced Rays

8.2 Selection of B-Spline Parameters

To adequately describe a curve on the surface of the analytic forebody with a B-spline, both the degree of the spline and the number of control points must be high enough to obtain a good approximation. For this analysis, six control points were chosen as an initial guess to keep the number of optimization parameters at a reasonable level. An arbitrary selection was made to study plane 20 since it was at the mid point of the body and has several curvature changes in the cross section view. The entire cross flow grid for plane 20 is shown in Figure 25.

A quadratic B-spline ($k=3$) with 6 control points ($n=5$) is insufficient to describe the surface of the body. The polynomial for each of the five segments between the points is 2nd order and has C^1 continuity. The curve is completely enclosed in the convex hull created by the polygon connecting the points, and is tangent to the vector $\mathbf{p}_1-\mathbf{p}_0$ and $\mathbf{p}_5-\mathbf{p}_4$ at the end points (Figure 26).

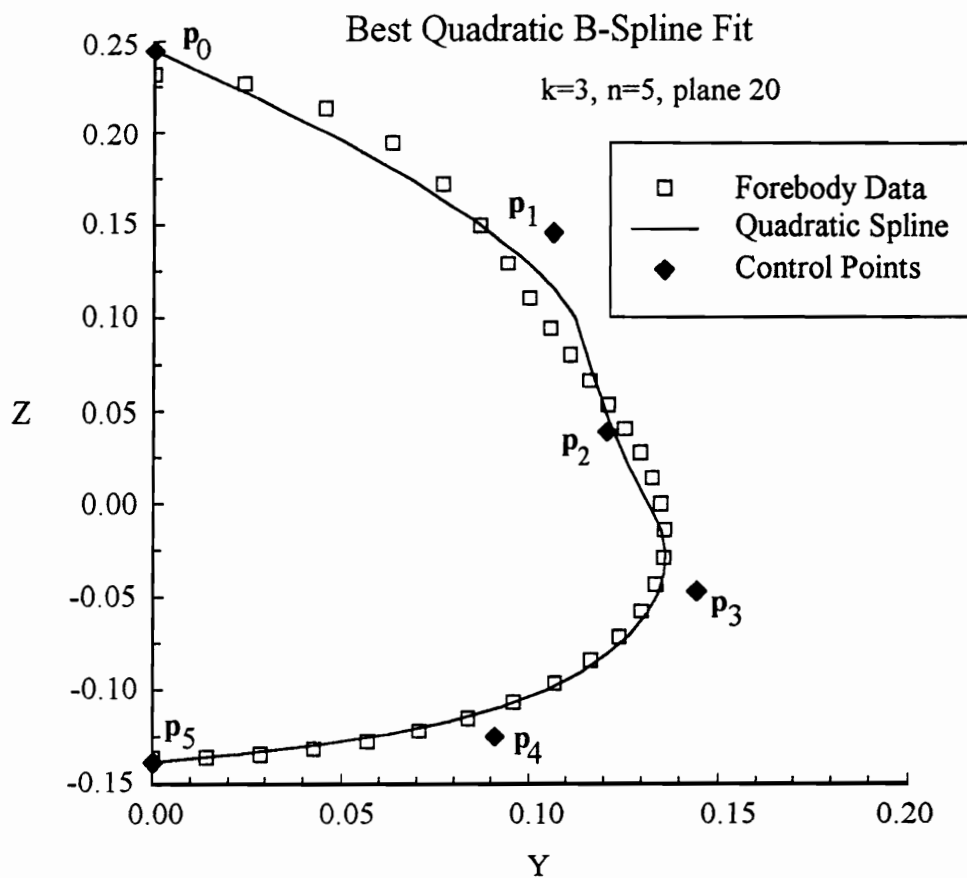


Figure 26. Six Control Point, Quadratic B-Spline

The tangency condition at the end points limited the quadratic spline's usefulness and it was eliminated from further consideration. A cubic B-spline ($k=4$) was next examined with the four interior control points fixed at the angular locations,

$$\frac{\pi}{5}, \frac{2\pi}{5}, \frac{3\pi}{5} \text{ and } \frac{4\pi}{5} \quad (36)$$

and the angular location of the end points fixed at 0 and π . At the time, the cubic spline seemed to exhibit insufficient flexibility and a quartic spline ($k=5$) was chosen as the baseline. Subsequently, a method was developed to determine the best number of control

points and the best degree of the spline. By generating B-splines with various numbers of control points and polynomial degrees, the position of the control points could be optimized to determine the best fit. For example, using a second order B-spline with 6 control points, the NLPQL algorithm was employed to determine the optimum location of the control points. The objective function was

$$f = \sum_{i=1}^{idim} \left[(y_i - y_{b_i})^2 + (z_i - z_{b_i})^2 \right] \tag{37}$$

where y_b and z_b are the locations of the grid points on the target body, and *idim* equals 31, the number of points in the transverse direction for the target body. Table 9 shows the results that were generated for the second, third and fourth order splines.

Table 9. Optimal 6 Parameter B-Spline for Plane 20

B-spline Order	Objective Function
2	7.2×10^{-4}
3	5.3×10^{-4}
4	4.1×10^{-4}

Upon examination of these results, and the corresponding plots of the B-spline against the actual forebody data, it was determined that a small increase in the number of parameters could make a large impact on the ability of the spline to fit the target surface.

The new parameters for the optimization were the angular location of the interior points, four total, since the end points still remained fixed at 0 and π . Including the distance from the origin of each of the six control points, the total number of parameters was increased to ten, still a manageable number for a flow solver/optimization problem,

but a low number for a complex body shape. The improvement in the objective function (equation 37) for the ten parameter B-spline is shown in Table 10.

Table 10. Optimal 10 Parameter B-Spline for Plane 20

B-spline Order	Objective Function
2	1.1×10^{-5}
3	2.0×10^{-5}
4	7.1×10^{-7}

The heuristic approach seems to have paid off, at least for plane 20; the analysis shows that the quartic B-spline is at least one order of magnitude better than the cubic at matching the surface data from the original analytic forebody. Figure 27 shows the cubic and quartic approximations to plane 20 of the forebody; the differences are slight, but are especially noticeable near the inflection points. The position of the control points used to generate the B-spline is also shown in the graphs.

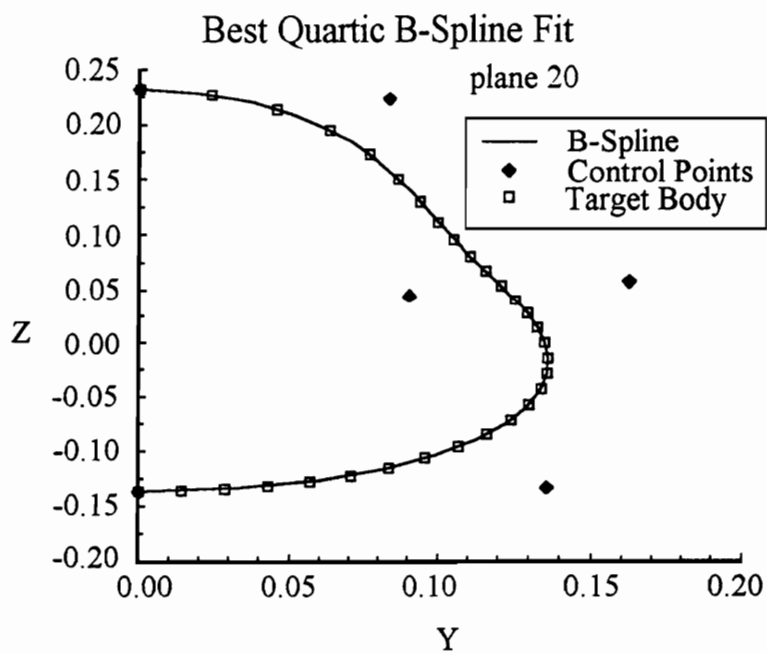
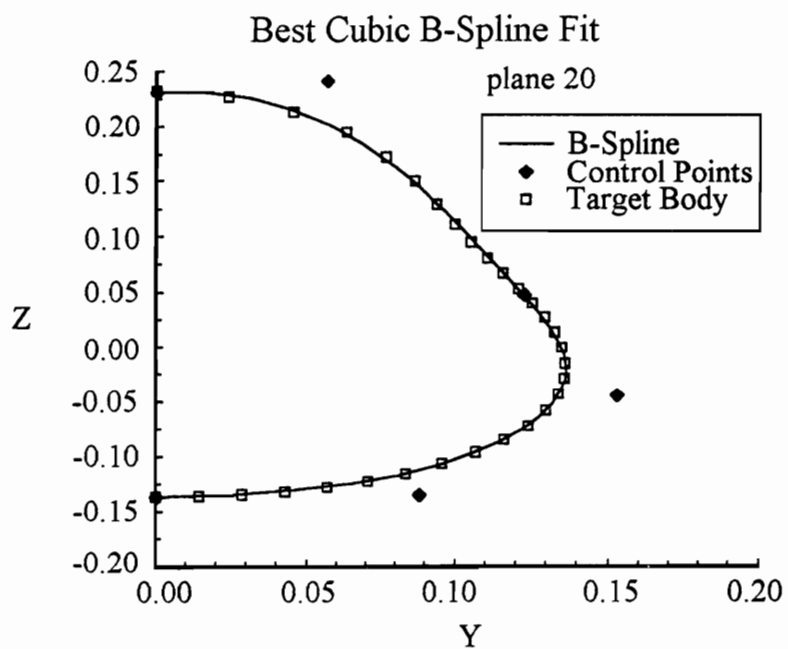


Figure 27. Best B-Spline Fits for Plane 20

Of course, this analysis is merely an exercise to show the usefulness of some of the various B-splines, since a designer would not normally have a-priori knowledge of the body shape. However, this method could be useful in the event that a shape optimization was being performed to improve an existing design, in which case a baseline curve may provide some insight into the shape of the final body. A variation of this technique will be used later in the inverse design analysis.

8.3 Multi-Parameter B-Spline Inverse Design - Inviscid

The inviscid target pressure distribution generated for an earlier test case was used as the design objective for the B-spline optimization (Figure 14). The test case was computed at Mach 1.70 and only inviscid calculations were performed at this point in the study. The body was space marched, with the first three planes computed from the target body to get the calculations started for the reasons described previously. Although the forward section of the body is nearly a circular arc, and could be described by fewer B-spline segments, the same number of control points and the degree of the B-spline were held constant for the length of the body to simplify the design procedure. At this point, a starting guess for the ten parameters is needed for the optimizer.

Table 11. Inviscid Analytic Forebody Input Parameters

idim (transverse direction)	31
jdim (direction normal to body)	31
kdim (streamwise direction)	41
i0 Boundary Condition	x-z plane symmetry
idim Boundary Condition	x-z plane symmetry
j0 Boundary Condition	Tangency
jdim Boundary Condition	First order extrapolation
k0 Boundary Condition	Fixed at free stream
kdim Boundary Condition	First order extrapolation

8.4 Starting the Iteration Process

A simplistic approach to finding an initial guess for the current plane was to use the parameters from the previous plane, assuming that the body was smooth and very few large changes were taking place from one plane to the next. In reality, for the analytic forebody the parameters from the previous plane produce a rather poor guess compared to one that can be generated from information that is known. By using the target pressure distribution, and Newton's Sine Squared Law (equation 4), or a variation of it, a better initial guess was obtained. Because Newton's Sine Squared Law is only good for high supersonic and hypersonic velocities, the Modified Newtonian law was used [26],

$$C_p = C_{p_{max}} \sin^2 \theta \quad (38)$$

Admittedly, this method is valid only for blunt bodies at high Mach numbers, the previously mentioned constraints of Newton's Sine Squared law hold, and it does not

work at all for regions of negative C_p , but the savings in CPU time from obtaining a marginally better initial guess is quite large, on the order of several full design iterations. An improvement in initial guess techniques would be appropriate for cases other than this particular body to make the inverse design procedure more general. Once again, upper and lower bounds were attached to the optimization parameters to keep the B-spline approximation in an area where GASP would have a reasonable chance of finding a feasible solution. This part of the process requires user interaction, and also needs to be improved, as discussed later. A flow chart of the solution process is presented in Figure 28. Note the exchange of information between the three main components of the inverse design process.

The section of the flow chart that contains the 'Compute Gradients' box is a simplification of the actual inverse design procedure. To compute the gradients for the optimization routine, each parameter λ is perturbed slightly by a value $+\Delta\lambda$, a new B-spline and grid are generated, and the flow solver is called to compute a solution. The parameter is then perturbed by $-\Delta\lambda$, and the procedure is repeated. The partial derivative is then computed from

$$\frac{\partial F}{\partial \lambda} = \frac{F|_{\lambda+\Delta\lambda} - F|_{\lambda-\Delta\lambda}}{2\Delta\lambda} \quad (39)$$

This procedure is computationally expensive, since two function evaluations are required for each parameter to obtain an approximation to the first derivative.

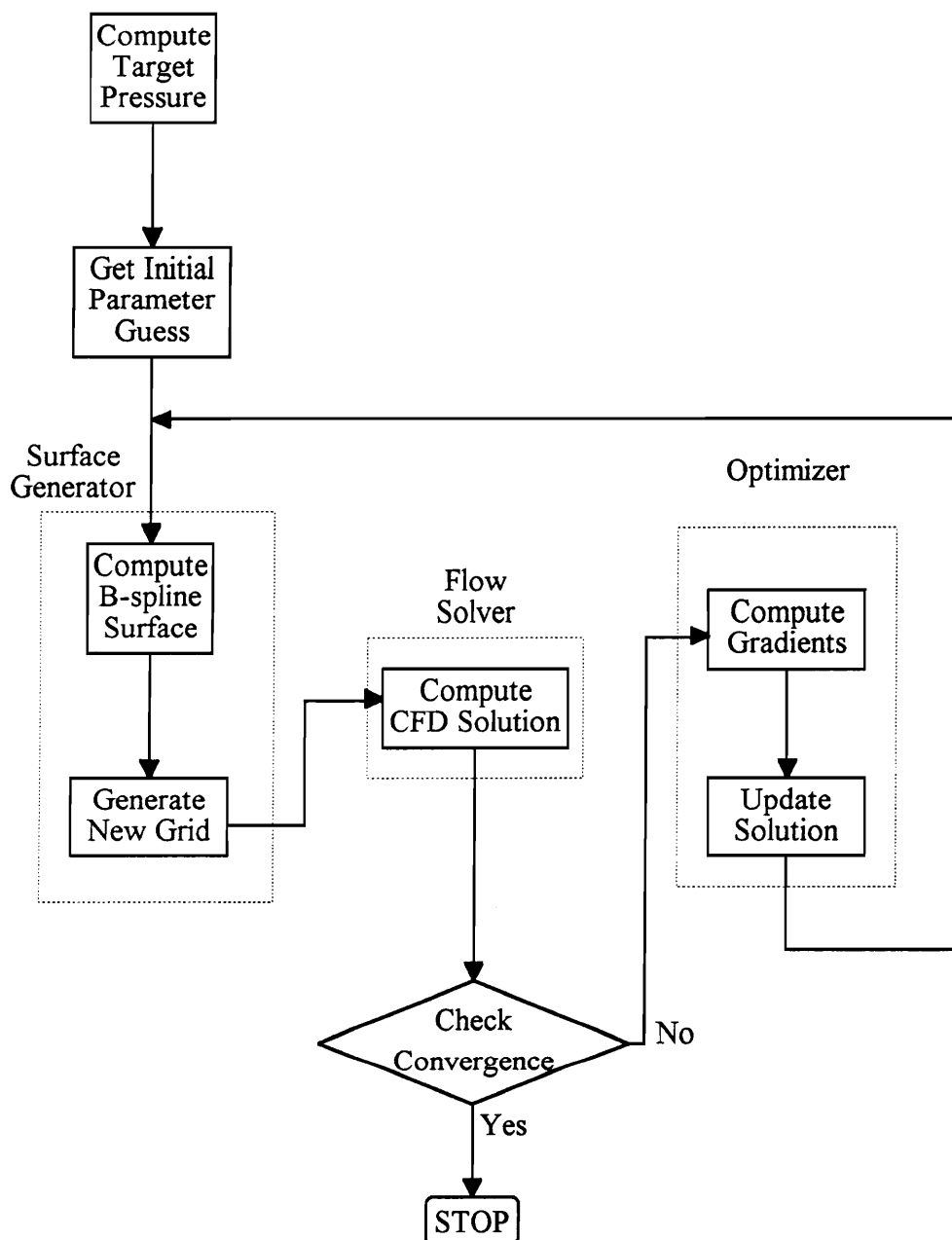


Figure 28. Inverse Design Methodology

Each time the NLPQL optimizer computed the next step in the iterative sequence, at least $2n+1$ function calls were made to GASP, where n is the number of parameters in the optimization process. The first function call evaluated the objective function at the

current set of parameters, the following $2n$ calls were used to evaluate the gradient, obtained by taking a central derivative using some small $\Delta\lambda_i$, on the order of 10^{-6} . An attempt was made to utilize a one-sided derivative, and save n function evaluations, but the gradient direction computed using this method was insufficient to obtain convergence. All further function calls were in performing the line search algorithm to determine how large of a step to take in the descent direction.

The following figures present some results of the inviscid inverse design over the analytic forebody. Figure 29 shows the initial 10 parameter 4th order B-spline approximation to plane 20, with the associated control points and the actual forebody data. It also shows the final design, a converged solution from the optimizer, which means that the solution can no longer be improved with the given number of parameters and order of the B-spline. Figure 30 shows the pressure coefficient vs the transverse location around the body for the initial and final B-spline approximations. Notice that even though the initial B-spline approximation in Figure 29 appears to be close to the body, the pressure coefficient is up to 60% off from the design value. For this case, an inviscid flow over a relatively smooth body at supersonic speeds, the following observations can be made about Figures 29 and 30;

- where the B-spline approximation is inside the actual body design, the corresponding pressure coefficient is lower than the target value,
- where the B-spline approximation is outside the actual body design, the corresponding pressure coefficient is higher than the target value,
- the 4th order B-spline with 6 control points accurately describes this body and approximates the solution to within 10^{-4} of the objective function (equation 35).

Figures 31 and 32 show the same results for plane 15 of the analytic forebody. The results for plane 20 were actually calculated first, as a test case, and then the rest of the body was returned to and solved, including re-solving plane 20. This is the reason that the initial B-spline guess for plane 15 is much better than plane 20, the improved initial guess algorithm had been implemented to save CPU time. Figures 33 and 34 show three more planes of the body, planes 25, 30 and 35. Agreement is not as good as the previous planes, since the fourth order B-spline was not able to match the body as well as the previous cross sections.

The remainder of the body was designed using the same method, and the three dimensional pressure contours were plotted with those of the target body (Figure 35). To determine the variation between the target body and the inverse design solution, the C_p difference was plotted in Figure 36. The difference between the target C_p and the inverse design solution is on the order of 10^{-5} at most points in the flow. The first plot shows the C_p differential on the $i=0$ plane, or the x - z plane, which is at a transverse location of 0° . The second plot shows the C_p differential on the exit plane, the y - z plane.

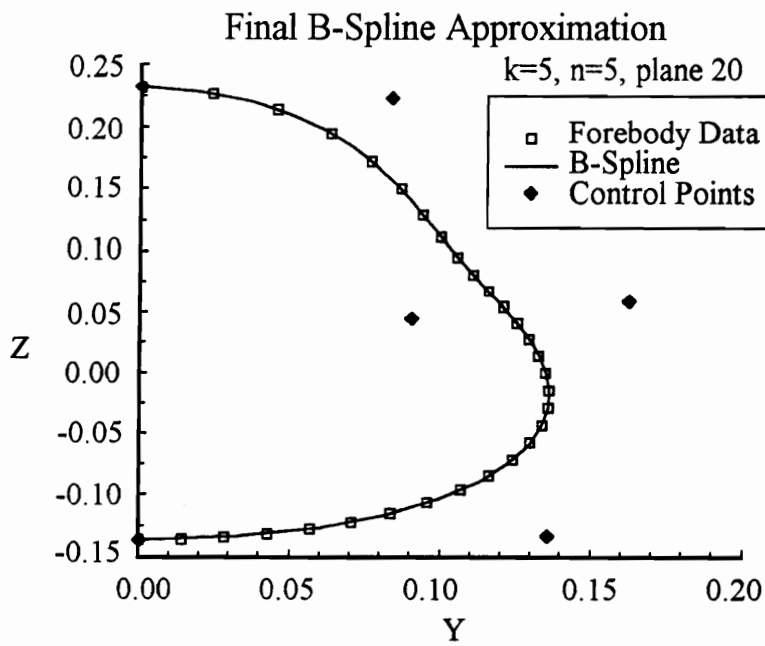
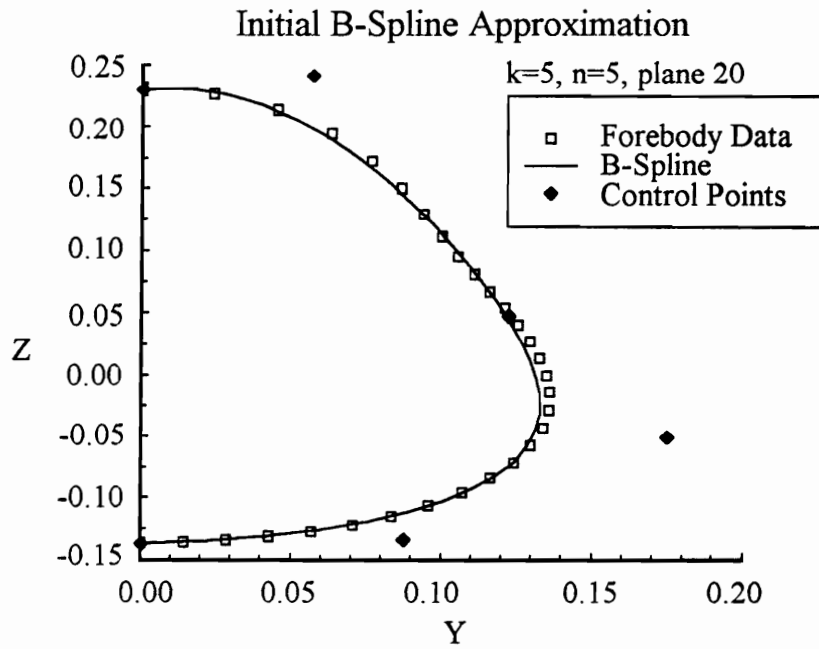


Figure 29. Initial and Final B-Spline Approximations to Forebody, Plane 20

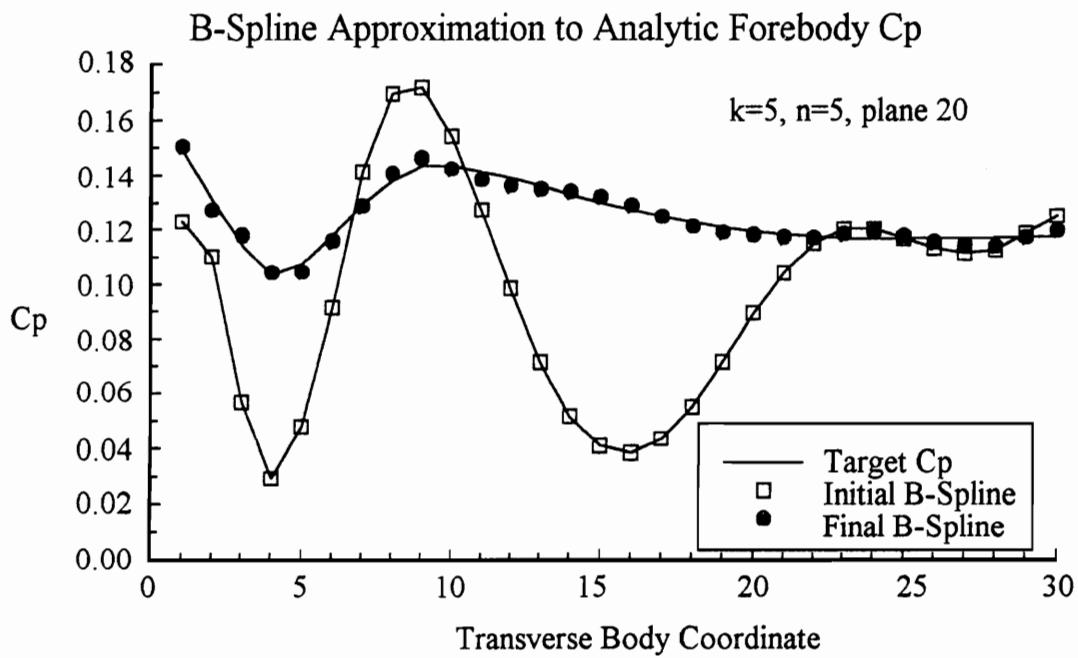


Figure 30. Inviscid Target C_p vs B-Spline Approximations, Forebody, Plane 20

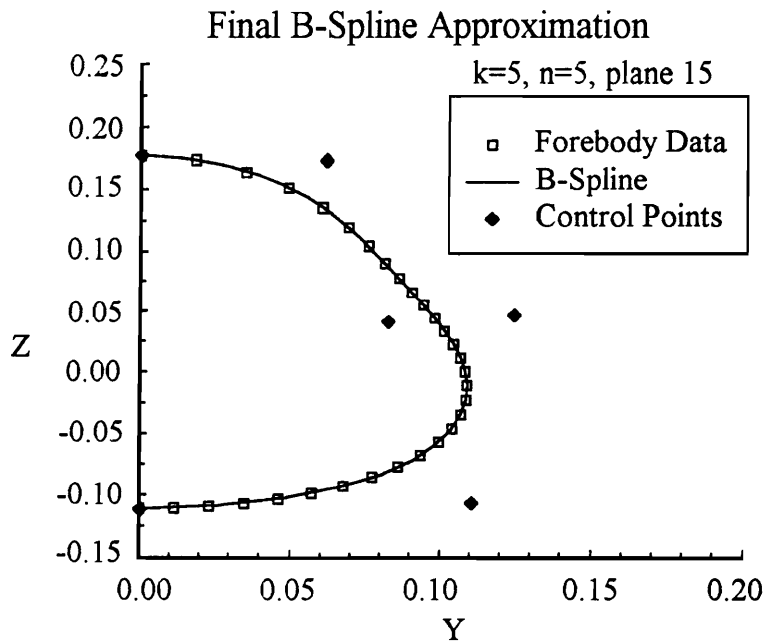
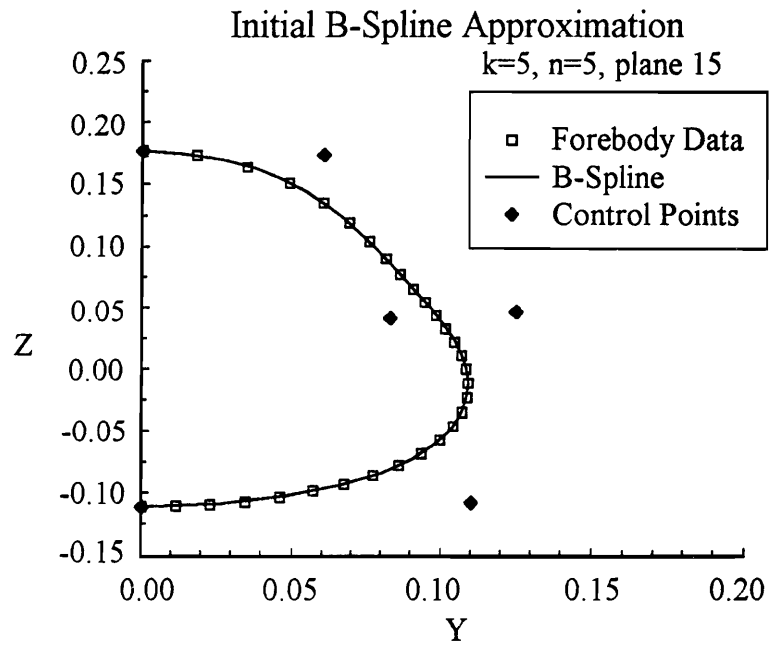


Figure 31. Initial and Final B-Spline Approximations to Forebody, Plane 15

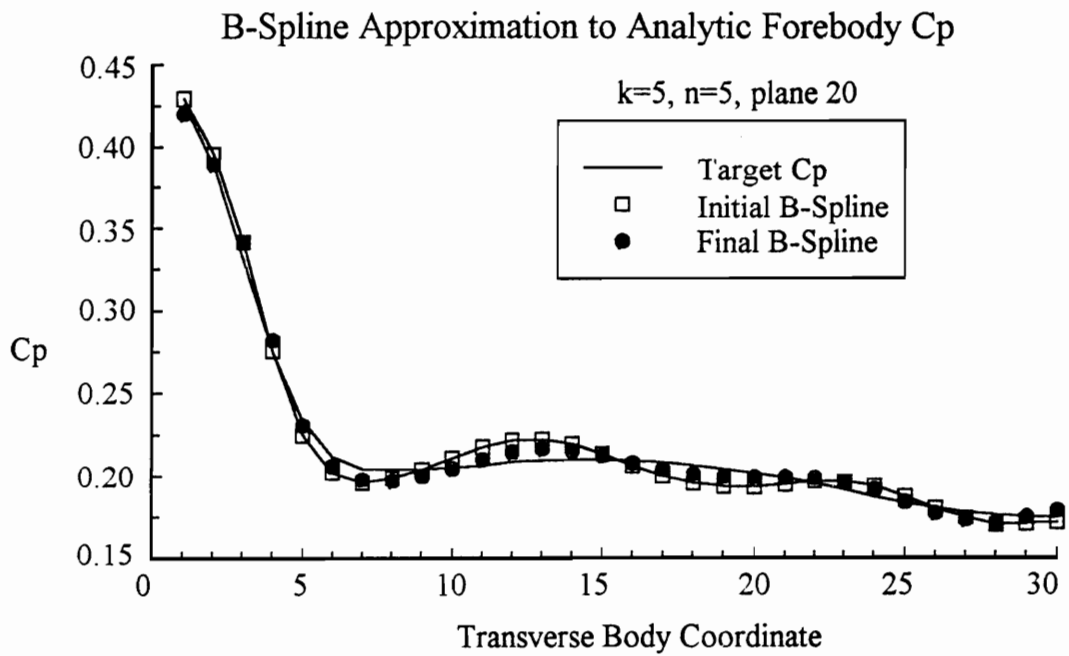


Figure 32. Inviscid Target Cp vs B-Spline Approximations, Forebody, Plane 15

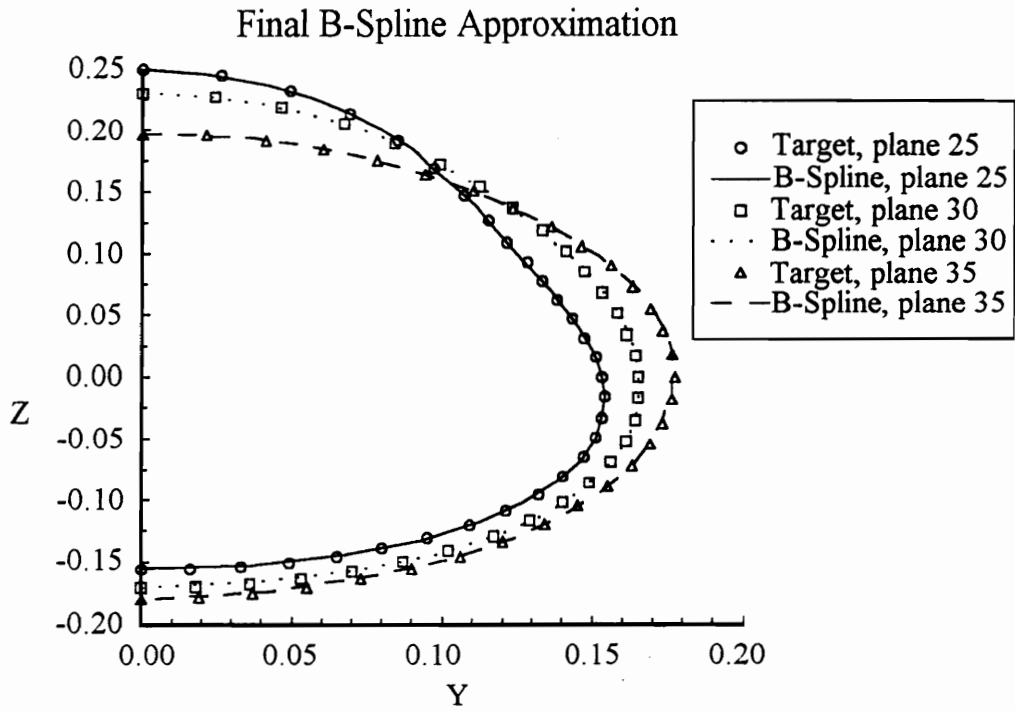


Figure 33. Final B-Spline Approximations to Forebody, Planes 25, 30 and 35

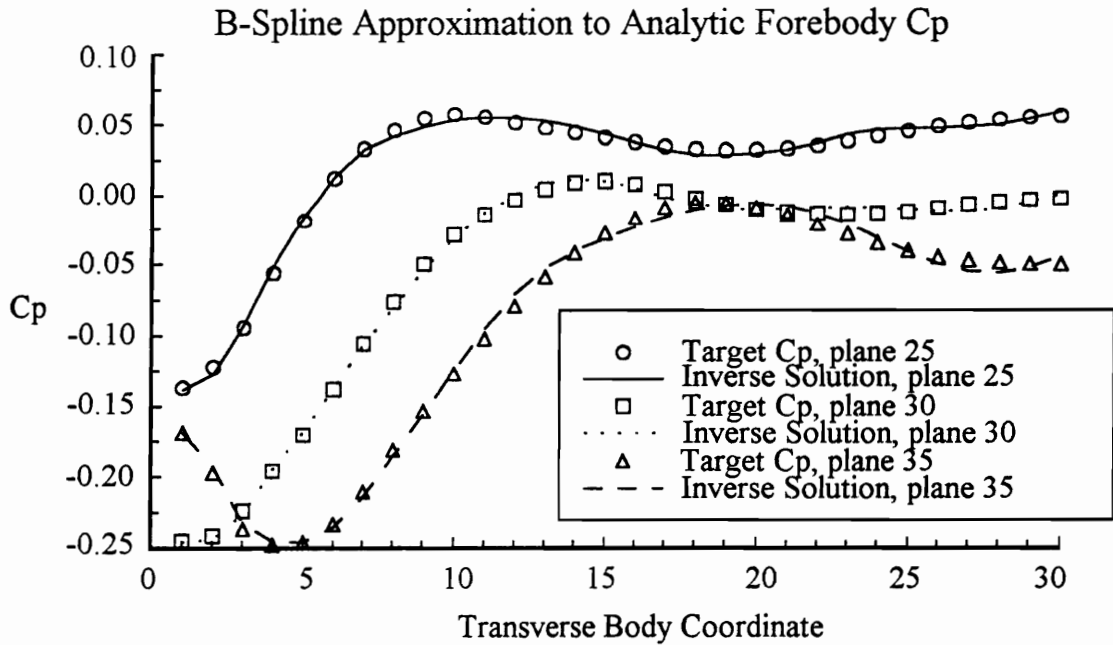
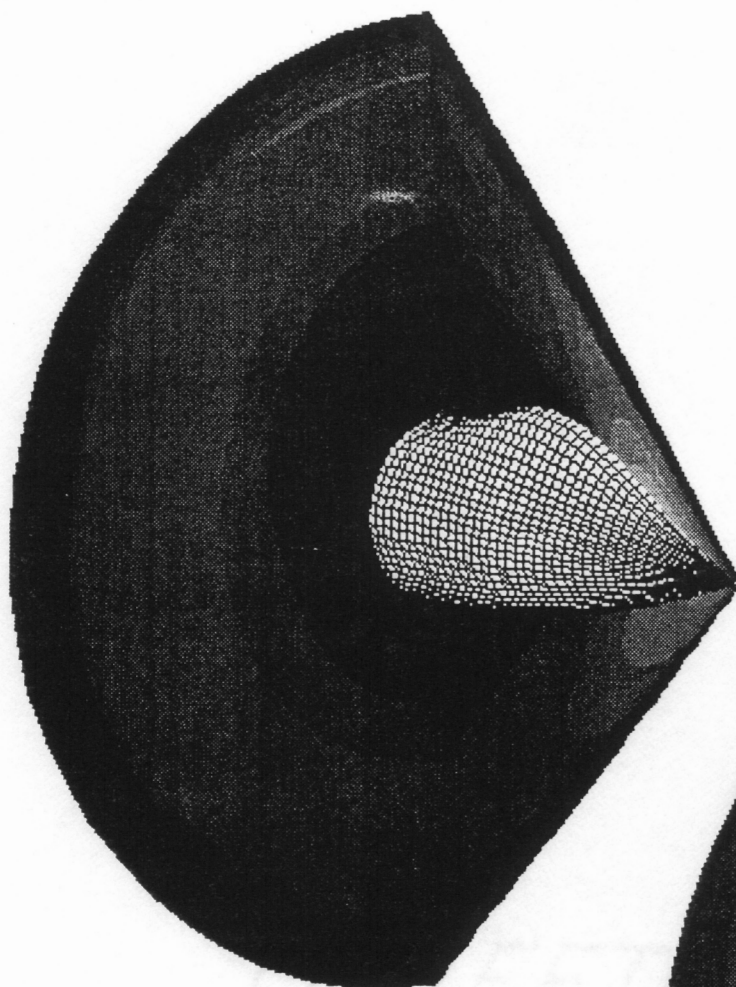
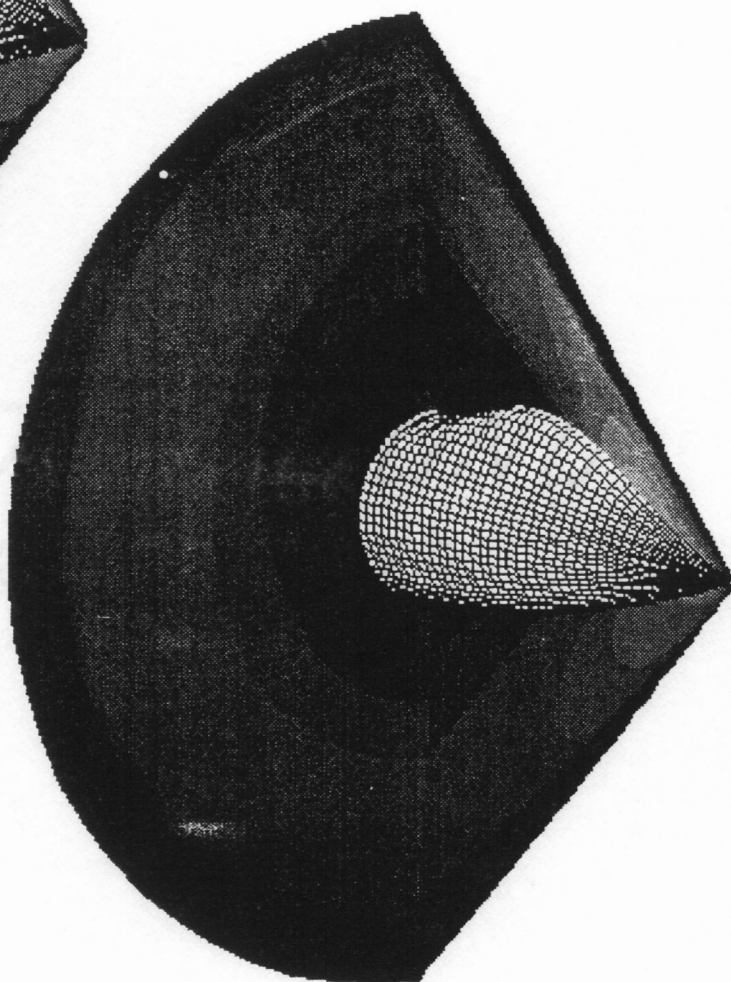


Figure 34. Inviscid Target Cp vs B-Spline Approximations, Planes 25, 30 and 35



Target Body and Pressure Contours



Inverse Design Body and Pressure Contours

Figure 35. Inviscid Target Pressure vs Inverse Solution, Analytic Forebody

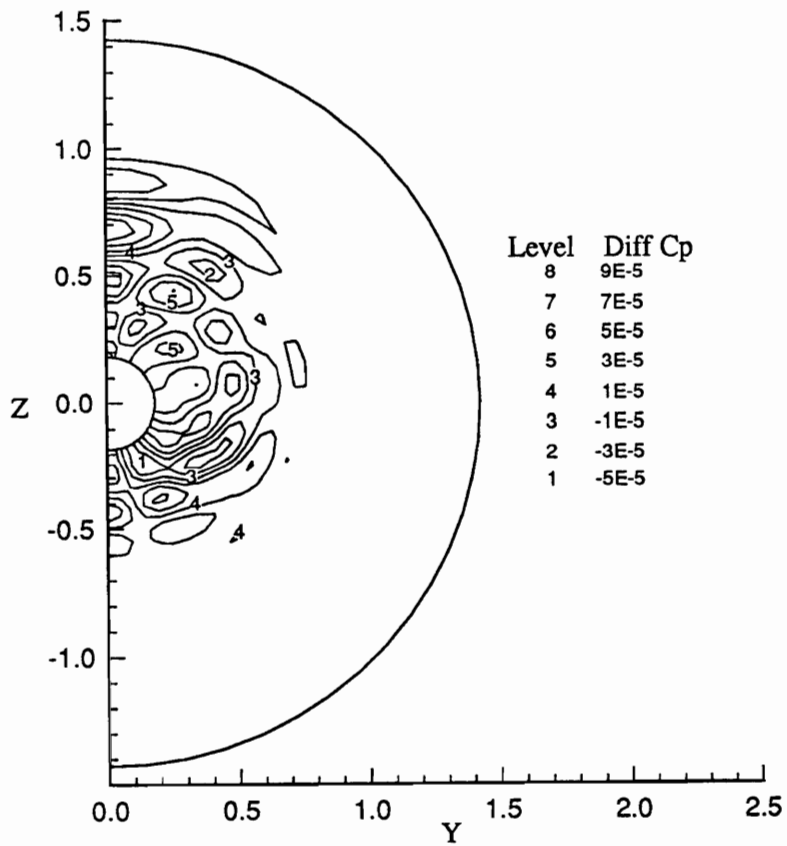
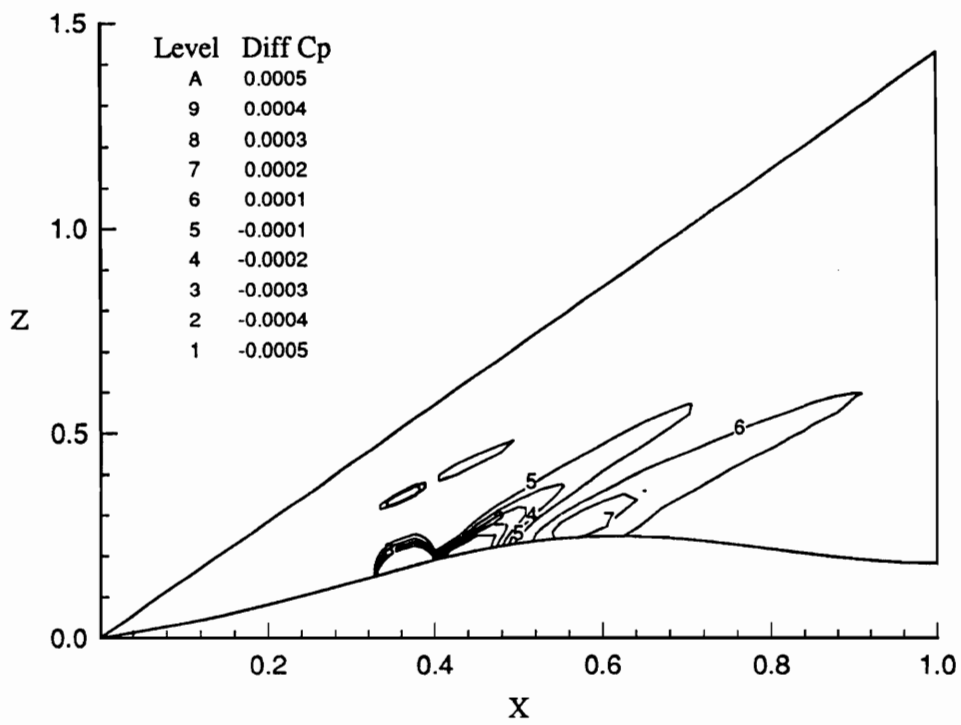


Figure 36. Analytic Forebody Pressure Coefficient Difference Contours

8.5 Multi-Parameter B-Spline Inverse Design - Turbulent Flow

To demonstrate the ability of the inverse design method to perform viscous calculations, the analytic forebody was analyzed at the same Reynolds number as the experiment, $Re=2.33 \times 10^6$ 1/ft, and turbulent flow was assumed. The parameters for the analysis are presented in Table 12. The changes from the inviscid case were in the boundary condition at the wall (no slip instead of tangency), the inclusion of the viscous flux vector, and the turbulence modeling. The turbulence model chosen was the algebraic model known as the Baldwin-Lomax model. It is a multi-dimensional formulation of the Clauser eddy viscosity model and is well suited for wall bounded flows [2]. In order to generate an appropriate turbulent velocity profile near the wall, the grid from the inviscid case was highly stretched using a hyperbolic tangent law [38].

Table 12. Turbulent Analytic Forebody Input Parameters

idim (transverse direction)	31
jdim (direction normal to body)	31
kdim (streamwise direction)	41
i0 Boundary Condition	x-z plane symmetry
idim Boundary Condition	x-z plane symmetry
j0 Boundary Condition	No slip
jdim Boundary Condition	First order extrapolation
k0 Boundary Condition	Fixed at free stream
kdim Boundary Condition	First order extrapolation
Reynolds Number	2.33×10^6 1/ft
Turbulence Model	Baldwin-Lomax

The stretching function was calculated with the expressions below, in which j was the direction normal to the body, and there were $jdim$ points in that direction. The variable β is a parameter that controls the amount of clustering near the wall. To obtain an appropriate grid for this body, β was set equal to 1.001.

$$b = \log\left(\frac{\beta+1}{\beta-1}\right) \quad (40)$$

$$\eta_y = \beta \frac{e^a - 1}{e^a + 1} \quad (41)$$

where

$$a = b \frac{jdim - j}{jdim - 1} \quad (42)$$

$$\eta_b = jdim - \eta_y(jdim - 1) \quad (43)$$

and finally,

$$r(i, j) = r(i, jdim) \frac{\eta_b - 1}{jdim - 1} + r(i, 1) \frac{jdim - \eta_b}{jdim - 1} \quad (44)$$

This stretched grid was used to calculate the target pressure on the analytic forebody, assuming turbulent flow. The results of the turbulent and inviscid calculations were once again compared to the NASA test case to see if the turbulence modeling improved the solution. The streamwise pressure coefficient distribution was examined at transverse locations of 0° , 90° and 180° around the body (Figure 37). The turbulent flow assumption allowed the data to more closely match the target pressure distribution everywhere on the body, but especially behind the canopy on top of the forebody. The

turbulence modeling more accurately described the physics in the flowfield, and the effective thickening of the body due to the turbulent boundary layer produced the demonstrated change in pressure coefficient.

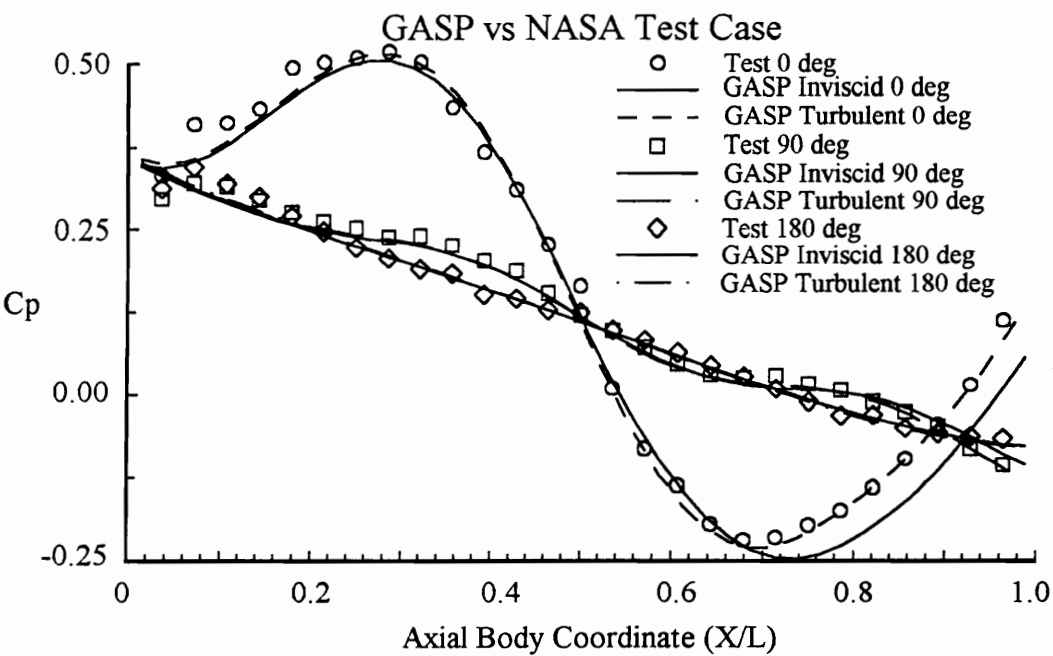


Figure 37. Turbulent GASP Calculations vs NASA Test Case

The target pressure was once again calculated using GASP and the velocity profile near the wall was examined for an appropriate shape (Figure 38). These velocity profiles are on the upper surface of the forebody, near the nose region, and clearly show a turbulent boundary layer profile, approaching the surface of the body with a zero slope, instead of a finite slope, that may be indicative of a laminar velocity profile. The value of

y^+ at the first grid cell near the wall was found to be on the order of 1.0, an acceptable level for accurate turbulence modeling [2].



Figure 38. Near Wall Turbulent Velocity Profiles

GASP supports the solution of the Reynolds Averaged Navier Stokes equations to perform turbulent flow analysis. To continue to take advantage of the computational efficiency of the space marching approach, by solving the Parabolized Navier Stokes equations, the flow should be supersonic everywhere except for a thin region in the boundary layer. A modification to the thin layer Navier Stokes equations known as the Vigneron technique is implemented in GASP to correct the pressure term on the full inviscid flux in the marching direction. This turns the system of equations parabolic instead of elliptic and allows space marching of the body [2]. This assumption does not

permit the solution of circulating or separated flows, and results in the loss of the viscous term in the streamwise direction, since the Thin-Layer derivatives in the normal and transverse directions can only be included in the PNS formulation.

Because the inviscid optimization had already been performed for the analytic forebody, the turbulent optimization was not repeated for the entire body. Several planes were analyzed instead, using the final optimal parameters from the inviscid calculations as the starting point for the turbulent case. Since turbulent calculations normally require more CPU time than inviscid calculations, it was expected that a designer would perform an inviscid analysis first, and then use the results for the turbulent analysis.

Since the inviscid optimization produced a B-spline shape that was very close to the target body shape, it was expected that the turbulent optimization would change the B-spline parameters very little. The pressure coefficient that resulted from using the B-spline parameters from the inviscid calculations on plane 15 and plane 20 is plotted in Figure 39. The agreement here is also within reasonable limits. The CPU time required per iteration (21 function evaluations) was near 100 seconds on the Cray Y-MP C-90.

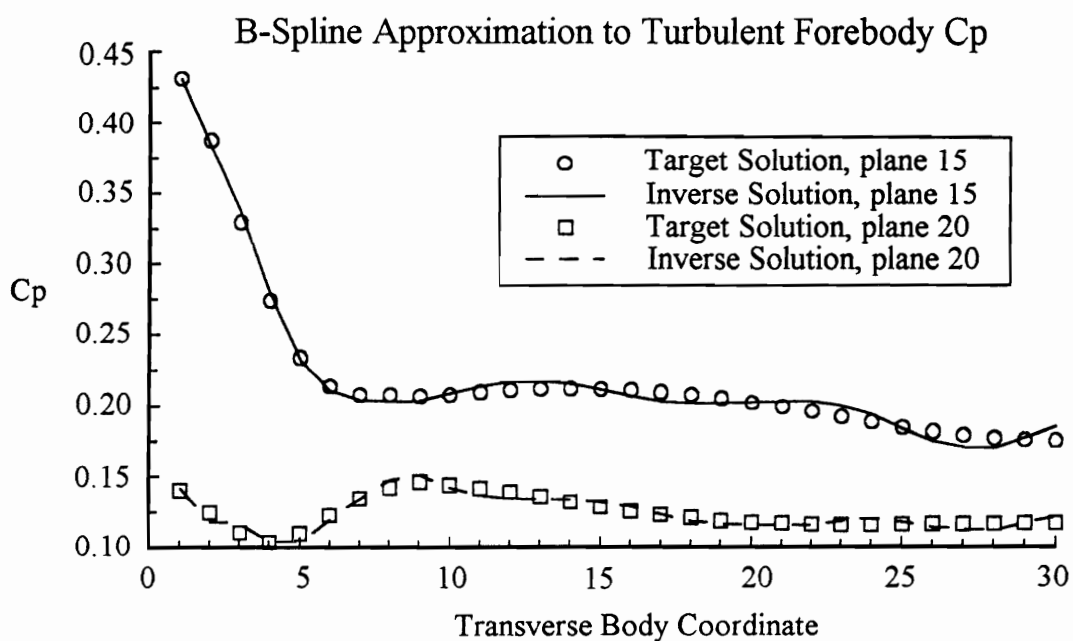


Figure 39. Turbulent Target Cp vs B-Spline Approximations, Planes 15 and 20

9. CONCLUSIONS

9.1 Conclusions About the Current Work

The inverse design procedure presented here has demonstrated the ability of the method to solve a range of problems involving different flow regimes and body shapes. The overall objective, to increase the efficiency and performance of the design engineer, has yet to be demonstrated. Hopefully this procedure will generate other research in the area, and the technology will find its way to the designer's desktop in the near future as a useful design tool.

Several conclusions reached in the process of completing the work are worth reiterating. While space marching a suitable problem, infinitely many solutions to the grid generated on the current computational plane may exist if conditions are not included for determining the 'planeness' of the cell faces. Also, interpolation routines must be included in the inverse design process to ensure that the objective function is comparing the target pressure distribution and the inverse design solution at similar locations.

Specifically, this work has shown that:

- fully 3-D, turbulent inverse design procedures are available today
- subsonic inverse design using this technique is fairly robust
- B-splines are flexible, general tools for surface description that require a low number of parameters to develop complex shapes, and lend themselves well to optimization techniques
- readily available Quasi-Newton optimization techniques can be used to solve these types of unconstrained inverse design problems

9.2 Recommendations for Future Work

A more robust technique for solving supersonic and hypersonic marching problems should be developed, as mentioned previously in the context of the inappropriate iterative guess causing the aerodynamic analysis code to fail. Perhaps techniques involving "trust-region" optimization techniques or intelligent preprocessing of the flow solver to throw out "infeasible" solutions could circumvent this problem and improve convergence rates.

Full use of the optimization techniques available should be investigated, including how constraints, and/or other objective functions such as skin friction coefficients, affect the design process. Progress in sensitivity analysis by automatic differentiation (Appendix) should be monitored as a possible method of forming derivatives and saving large amounts of CPU time.

A graphical interface, perhaps implementing Non-Uniform Rational B-Splines and B-spline surfaces, would greatly improve the capacity of the designer to understand and more easily implement this tool.

Several more test cases should be attempted to determine how a transonic problem, or a problem with shocks and shock interactions is handled by this procedure. A non-smooth body, such as a wing body combination, should also be attempted to determine the effectiveness of the B-spline representation on these bodies. A two part body design could be implemented where one B-spline description handles the wing and the other determines the surface of the body. The optimizer could then be utilized more fully by constraining the surfaces of the wing and body to blend together, and the constrained optimization problem could be investigated.

10. BIBLIOGRAPHY

1. Anderson, D.A., Tannehill, J.C. and Pletcher, R.H., Computational Fluid Mechanics and Heat Transfer, Hemisphere Publishing Co., 1984, pp. 181-260.
2. Walters, R.W., et. al, "GASP Version 2.2," Aerosoft, Inc., 1993.
3. Lighthill, M.J., "A New Method of Two-Dimensional Aerodynamic Design," Aeronautical Research Council, London, R&M 2112, 1945.
4. Henne, P., ed. Applied Computational Aerodynamics, Progress in Astronautics and Aeronautics, Volume 125, AIAA, 1990.
5. Volpe, G. and Melnik, R., "The Role of Constraints in the Inverse Design Problem for Transonic Airfoils," AIAA Paper 81-1233, June 1981.
6. Lee, J. and Mason, W.H., "A Three-Dimensional Inverse Method for Supersonic and Hypersonic Body Design", AIAA Paper 91-3325.
7. Dulikravich, G.S., et.al., "Aerodynamic Shape Optimization of Hypersonic Missiles", AIAA Paper 87-0272.
8. Bock, K.W., "Aerodynamic Design By Optimization", AGARD CP 463, No. 20, May 1989.

9. Bowcutt, K.G., Anderson, Jr., J.D., and Capriotti, D., "Viscous Optimized Hypersonic Waveriders", AIAA Paper 87-0272.
10. Lee, J., "Efficient Inverse Methods for Supersonic and Hypersonic Body Design, with Low Wave Drag Analysis," Virginia Polytechnic Institute and State University, Dept. of Aerospace and Ocean Engineering, May, 1991.
11. Burgreen, G. and Baysal, O., "Three-Dimensional Aerodynamic Shape Optimization of Wings using Sensitivity Analysis," AIAA Paper 94-0094, January 1994.
12. Vanderplaats, G.N., Numerical Optimization Techniques for Engineering Design, McGraw-Hill, 1984, pp.283-291.
13. Davis, W., "TRO-2D: A Code for Rational Transonic Aero Optimization," AIAA Paper 85-0425, January 1985.
14. Ratcliff R.R. and Carlson, L.A., "A Direct-Inverse Transonic Wing-Design Method in Curvilinear Coordinates Including Viscous Interaction", AIAA Paper 89-2204.
15. Birckelbaw, L., "Inverse Airfoil Design Using the Navier-Stokes Equations", AIAA Paper 89-2202.

16. Selig, M.S. and Maughmer, M.D., "Generalized Multi-Point Inverse Airfoil Design", AIAA Paper 91-3333.
17. Aidala, P.V., Davis, W.H., Jr., and Mason, W.H., "Smart Aerodynamic Optimization," AIAA Paper 83-1863, July 1983.
18. Xia, Z.X., "A Computational Method for Inverse Design of Transonic Airfoil and Wing", AIAA Paper 93-3482.
19. Campbell, R.L. and Smith, L.A., "A Hybrid Algorithm for Transonic Airfoil and Wing Design", AIAA Paper 87-2552.
20. Campbell, R.L. and Smith, L.A., "A Method for the Design of Transonic Flexible Wings", NASA Technical Paper 3045, 1990.
21. Campbell, R.L. and Smith, L.A., "New Developments in CFD Design Using the Direct/Iterative Approach", NASA CFD Conference, NASA Ames Research Center, March, 1991.
22. Smith, L.A. and Campbell, R.L., "Applications of a Direct/Iterative Design Method to Complex Transonic Configurations", NASA Technical Paper 3234, 1992.

23. Jayaram, S. and Myklebust, A., "ACSYNT-A Standards Based System for Parametric Computer Aided Conceptual Design of Aircraft", AIAA Paper 92-1268.
24. Coe, D.H., "Skinning Engineering Models with Non-Uniform, Hierarchical B-Spline Surfaces," Virginia Polytechnic Institute and State University, Dept. of Mechanical Engineering, July 1993.
25. Mason, W.H. and Lee, J., "Aerodynamically Blunt and Sharp Bodies", 30th Aerospace Science Meeting, January, 1992.
26. Anderson, J.D., Jr., Modern Compressible Flow, McGraw-Hill, 1982, pp. 316-317.
27. Hayes, W.D. and Probstein, R.F., Hypersonic Flow Theory, Academic Press, 1966.
28. Schetz, J.A., Boundary Layer Analysis, Prentice-Hall, 1993, p. 198.
29. Haftka, R.T., Elements of Structural Optimization, Dordrecht Publishing, 1992, pp. 115-158.
30. Garbow, B.S., Matrix Eigensystem Routines: EISPACK guide extension, Springer-Verlag, 1977, pp. 30-32, 72-74.

31. Anderson, E., Lapack User's Guide, Society for Industrial and Applied Mathematics, 1992.
32. Luenberger, D.G., Linear and Nonlinear Programming, Addison-Wesley Publishing Co., 1984, p. 206.
33. Schittkowski, K., "NLPQL, A Fortran Subroutine Solving Constrained Nonlinear Programming Problems", Annals of Operations Research, Volume 5, 1985-1986, pp. 485-500.
34. Gill, P.E., Murray, W. and Wright, M.H., Practical Optimization, Academic Press, 1981, pp. 88-92.
35. Dennis, J.E. and Schnabel, R.B., Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall, 1983.
36. Townsend, J.C., et. al., "Surface Pressure Data on a Series of Analytic Forebodies at Mach Numbers from 1.70 to 4.50 and Combined Angles of Attack and Sideslip," NASA Technical Memorandum 80062, June 1979.
37. Mortenson, M.E., Geometric Modeling, John Wiley and Sons, 1985, p. 125-134.
38. Walters, R.W., et. al., "A User's Guide to GASP," Virginia Polytechnic Institute and State University, Dept. of Aerospace and Ocean Engineering, NASA Langley Research Center, November 1990.

39. Dadone, A. and Grossman, B., "Surface Boundary Conditions for the Numerical Solution of the Euler Equations", ICAM Report 92-10-04.
40. Godfrey, A.G., "Topics on Spatially High-Order Accurate Methods and Preconditioning for the Navier-Stokes Equations with Finite-Rate Chemistry", Virginia Polytechnic Institute and State University, Dept. of Aerospace and Ocean Engineering, December, 1992.
41. Bischof, C., Corliss, G., Green, L., Griewank, A., Haigler, K. and Newman, P., "Automatic Differentiation of Advanced CFD Codes for Multidisciplinary Design", presented at the Symposium on High Performance Computing for Flight Vehicles, December 1992.
42. Applebaum, M.P., et. al., "Application of Thermo-Chemical Models on Unstructured Meshes," AIAA Paper 93-0895.
43. Arora, J.S., Introduction to Optimum Design, McGraw-Hill, 1989.
44. Bellman, R.E. and Roth, R.S., Methods in Approximation, D. Reidel Publishing Co., 1986, pp. 40-59.
45. Borggaard, J., et.al., "Sensitivity Calculations for a 2D, Inviscid, Supersonic Forebody Problem", ICASE Report No. 93-13, NASA Langley Research Center, Hampton, Virginia, 1993.

46. Burden, R.L. and Faires, J.D., Numerical Analysis, PWS-KENT Publishing Co., 1989, pp. 527-558.
47. Curtis, A.R., Powell, M.J.D. and Reid, J.K., "On the Estimation of Sparse Jacobian Matrices", *Journal of the Institute of Mathematics and its Applications*, Volume 13, 1974, pp. 117-119.
48. Dulikravich, G.S. and Sheffer, S.G., "Aerodynamic Shape Optimization of Hypersonic Configurations Including Viscous Effects", AIAA Paper 92-2635.
49. Farin, G., Curves and Surfaces for Computer Aided Geometric Design, Academic Press, Inc., 1993, p. 157-186.
50. Fletcher, R., Practical Methods of Optimization, John Wiley and Sons, 1980.
51. Hirsch, C., Numerical Computation of Internal and External Flows, Volume 2, John Wiley and Sons, 1990.
52. Hutchison, M.G., "Multi-Disciplinary Optimization of High-Speed Civil Transport Configurations using Variable-Complexity Modeling", Virginia Polytechnic Institute and State University, Dept. of Aerospace and Ocean Engineering, March, 1993.

53. Lee, J. and Mason, W.H., "Inverse Solution Uniqueness and Domain of Existence for Supersonic Space Marching Applications", AIAA Paper 92-0028.
54. Lee, J. and Mason, W.H., "Development of an Efficient Inverse Method for Supersonic and Hypersonic Body Design", AIAA Paper 91-0395.
55. Myklebust, A., "Improving Aircraft Conceptual Design Tools - New Enhancements to ACSYNT", AIAA Paper 93-3970.
56. Press, W.H., et. al., Numerical Recipes in C, Cambridge University Press, 1989, pp. 60-71.
57. Ralston, A. and Rabinowitz, P., A First Course in Numerical Analysis, McGraw-Hill, 1978, pp. 338-344.
58. Rogers, D.F., Mathematical Elements for Computer Graphics, McGraw Hill, 1976, pp. 144-155.
59. Schetz, J.A., Foundations of Boundary Layer Theory, Prentice-Hall, 1984, p. 8.
60. Schramm, U. and Pilkey, W.D., "Structural Shape Optimization for the Torsion Problem Using Direct Integration and B-Splines," *Computer Methods in Applied Mechanics and Engineering*, Vol. 107, August 1992.

61. Schramm, U., "The Use of Rational B-Splines for Shape Optimization," Presented at Virginia Polytechnic Institute and State University, January 25, 1994.
62. Schubert, L.K., "Modification of a Quasi-Newton Method for Nonlinear Equations with a Sparse Jacobian", *Mathematics of Computation*, Volume 24, Number 109, 1970.
63. Smith, B.T., et. al., Matrix Eigensystem Routines, EISPACK guide, Springer-Verlag, 1976, p. 30.
64. Thwaites, B. ed., Incompressible Aerodynamics, Oxford at the Clarendon Press, 1960.
65. Wait, R., The Numerical Solution of Algebraic Equations, John Wiley, 1979, pp. 118-131.

11. APPENDIX

11.1 Sensitivity Analysis by Automatic Differentiation

The time honored technique for performing sensitivity analysis of a multidisciplinary design optimization problem is to obtain the sensitivity derivatives by divided differences. That is, derivatives of interest can be obtained by running a CFD code several times and obtaining numerical derivatives by manipulation of the results.

A new technique called automatic differentiation has been developed which is worthy of examination. "Automatic differentiation is a chain-rule-based technique for evaluating the derivatives of functions defined by computer programs with respect to their input variables" [41]. For a specified Fortran subroutine, in which the independent and dependent variables are defined, the ADIFOR code will produce a new Fortran 77 code that will compute the partial derivatives of the result with respect to the dependent variables.

In Bischof, Griewank, et. al. [41], the code was applied to a 2-D transonic, inviscid flow over a NACA 1406 airfoil and a 3-D TLNS code for the ONERA M-6 wing including turbulence and multi-grid calculations. Both sample problems produced good results for derivatives such as C_L and C_D vs Mach number, angle of attack, and Reynolds number. The ADIFOR technique attempts to take advantage of the vectorization of the original CFD code, and produces derivatives much faster than the divided difference technique.

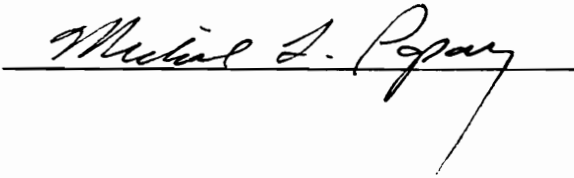
The major drawback of the procedure at this time is that the geometric sensitivity derivatives, of primary interest to the designer, are still under scrutiny and have not been applied. Therefore, the mesh generation/design process has not been fully implemented

yet. For the present inverse design analysis, derivatives relating to the changes in design variables with changes in geometry are of primary importance, and automatic differentiation has been relegated to a topic of future work.

VITA

The author was born in Zanesville, Ohio on July 11, 1964. He completed his B.S. in Aerospace and Ocean Engineering at Virginia Polytechnic Institute and State University in June 1986. In July 1986, he started work as a Member of the Technical Staff at the Ballistic Missiles Division of TRW, Inc. in San Bernardino, California. After four years working on various projects relating to preliminary design and systems engineering of ballistic missile systems, he returned to VPI&SU to pursue his Ph.D. in Aerospace Engineering.

The author is an active member in the AIAA, having served on one national committee and twice acting as the Western Region Student Conference Co-Chairman. His interests include Computational Fluid Dynamics, Aerodynamic Design, and golf, not necessarily in that order.

A handwritten signature in cursive script, reading "Michael L. Pappas", is written over a horizontal line. The signature is fluid and extends slightly below the line.