A STUDY OF TRANSONIC AERODYNAMIC ANALYSIS METHODS FOR USE WITH A HYPERSONIC AIRCRAFT SYNTHESIS CODE

GRANTOR - NASA AMES RESEARCH CENTER

GRANTEE - CAL POLY STATE UNIVERSITY

GRANT NUMBER NCC2-660

6/15/90 - 2/14/92

Principal Investigator

Dr. Doral R. Sandlin

Student Investigator

Paul Christopher Davis

May, 1992

ABSTRACT

A STUDY OF TRANSONIC AERODYNAMIC ANALYSIS METHODS FOR USE WITH A HYPERSONIC AIRCRAFT SYNTHESIS CODE

Paul Christopher Davis

March 1992

The purpose of this study is to develop a means of performing routine transonic lift, drag, and moment analyses on hypersonic all-body and wing-body configurations. The analysis method is to be used in conjunction with the Hypersonic Vehicle Optimization Code (HAVOC) employed by the Systems Analysis Branch at NASA Ames Research Center.

The approach begins with a review of existing techniques, after which three methods, chosen to represent a spectrum of capabilities, are tested and the results are compared with experimental data. The three methods consist of a wave drag code, a full potential code, and a Navier-Stokes code. The wave drag code, representing the empirical approach, has very fast CPU times, but very limited and sporadic results. The full potential code provides results which compare favorably to the wind tunnel data, but with a dramatic increase in computational time. Even more extreme is the Navier-Stokes code, which provides the most favorable and complete results, but with a very large turnaround time. Despite the large CPU times, the full potential code, TRANAIR, is used for additional analyses, because of the superior results it can provide over empirical and semi-

empirical methods, and because of its automated grid generation, which gives it a large advantage over a traditional Euler or Navier-Stokes code.

TRANAIR analyses in this study, include an all-body hypersonic cruise configuration and an oblique flying wing supersonic transport. Although a complete integration of TRANAIR into HAVOC is unrealistic at this point, TRANAIR can be used effectively in the preliminary design process of hypersonic vehicles. To facilitate this, a geometry interface is developed to transfer HAVOC geometry models into TRANAIR geometry input files.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

a	speed of sound
C_D	coefficient of drag
C_L	coefficient of lift
C_M	moment coefficient
C_P	coefficient of pressure, $(P - P_\infty)/q_\infty$
d	x intercept
dx	length of computational box
$D(\theta)$	wave drag
e	internal energy per unit mass
$\mathcal{F}$	nonlinear partial differential operator
F(x)	discrete function representing the boundary value problem
g_1	Neumann boundary condition parameter
g_3	Dirichlet boundary condition parameter
$\mathcal{G}$	continuous Green's function
G	discrete Green's function
h	Δt
J	Bateman functional
J	Jacobian
KN	number of networks in a TRANAIR network file
KT	boundary condition parameter for TRANAIR network
n	normal
l	diameter of box containing element trial function
L	discrete operator for the given boundary value problem
M	Mach number
N	sparse solver preconditioner
NM	number of rows in a TRANAIR network
NN	number of columns in a TRANAIR network

NX	number of equal intervals the x-axis is divided into
Nθ	number of equal intervals the domain of θ is divided into
O	order of magnitude
p	static pressure
Pr	Prandtl number
q	dynamic pressure
q	magnitude of velocity
Q	source strength unknowns
Q	discrete source unknowns
r	ratio of total property to the freestream value
R	distance from boundary
Re	Reynolds number
S	entropy
S(θ,x)	area intersected by cutting plane
t	time
$\mathcal{T}$	continuous far field (Prandtl-Glauret) operator
T	discrete far field operator
u,v,w	velocity components
U,V,W	contravariant velocity components
v	local velocity
V	velocity
W	mass flux
x,y,z	Cartesian coordinates
α	angle of attack
α	average operator across a boundary
β	yaw angle
δ	discrete delta function
δ	variation
δ	central difference finite operator
Δ	difference across a boundary surface
Δ	undivided difference operator
Δ	forward difference operator
ε	small change
ε	smoothing term
γ	ratio of specific heats
λ	step length

Λ	sweep angle
μ	Mach angle
μ	unknown change in Φ across a surface
μ	switching function
μ	coefficient of viscosity
μ_t	eddy viscosity
Ω	domain of integration
ϕ	perturbation potential
ϕ	spanwise angle
Φ	total potential
Ψ	extrapolated boundary potential
ρ	density
Σ	boundary of domain or surface discontinuity
θ	angle between y-axis and normal to Mach plane projected on the yz-plane
ξ,η,ζ	transformed coordinates
∇	gradient
∇	backward difference operator

Subscripts

c	cut off value
LE	leading edge
p	total pressure
p	parent quantity
ps	psuedo-unknown
r	angle of attack rotated about
t	time
T	total temperature
TE	trailing edge
x,y,z	derivatives in the x, y, z coordinate directions
x,y,z	rolling, pitching, yawing moments about the x, y, z axes, respectively
0	condition where $p = p_\infty$
0	zero-lift condition
_	one sided derivative
∞	freestream condition

Superscripts

$\vec{}$	vector
^	unit quantity
^	matrix

Acronyms

ARC3D	Ames Research Center Navier-Stokes code
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
GMRES	Generalized Minimal Residual
HAVOC	Hypersonic Vehicle Optimization Code
LBO	regions of special interest
LHS	Left-Hand-Side
LU-ADI	Lower-Upper factored Alternating -Direction-Implicit
NASP	National Aero-Space Plane
OFW	Oblique Flying Wing
PANAIR	linear potential panel code
PLOT3D	Plotting package developed at NASA Ames
RAID	RA Division Interactive Displayer
RANS3D	Reynolds-Averaged Navier-Stokes code
RHS	Right-Hand-Side
SHADE	graphics program used at NASA Ames
TACONV	TRANAIR Conversion code
TRANAIR	full potential panel code
TSD	Transonic Small Disturbance
TVD	Total Variation Diminishing
WIBCO	Wing-Body Code

CHAPTER 1

Introduction

Objective

The goal of this study is to develop and implement a method, or methods, for predicting transonic aerodynamic characteristics of hypervelocity vehicle configurations. The selected method must be usable in a conceptual design vehicle synthesis computer code. An interface is to be developed to couple the analytically generated vehicle geometries of the synthesis code, with the geometry format used by the selected transonic analysis method.

The approach will include a review of existing methodologies for predicting lift, drag, and pitching moments for all-body shapes in the transonic flow regime. The methods can be empirical, semi-empirical, or numerical. The focus of the analysis will be on the body and wing-body combination only. The method must be able to account for the three-dimensional geometric characteristics of the vehicle. Since the intended use is with a synthesis design code, where typically many iterations are done to achieve convergence, importance will be placed on CPU and storage limits. Therefore, reasonable sacrifices in solution accuracy can be tolerated.

Hypersonic Aircraft Synthesis Code

The Hypersonic Vehicle Optimization Code (HAVOC) [Ref. 1] used at NASA Ames Research Center was initially developed in the 1960's. Since that time it has

undergone significant modifications and improvements. The basic code performs a series of analyses on the vehicle's geometry, including aerodynamics, propulsion, and flight trajectory. As part of the output, HAVOC provides the weight and volume of each major component. HAVOC's role has become increasingly important with the renewed interest in hypersonics resulting from the National Aero-Space Plane (NASP) program. While a vehicle is still in the initial design stages, codes like HAVOC can be used to determine how various changes in specific areas will affect the configuration's aerodynamics, propulsion system, and structure.

The input geometry required for HAVOC is analytically based. Four equations, each with 23 independent parameters, are used to define the body geometry. Each of the equations corresponds to one of four regions of the vehicle, the upper forebody, the lower forebody, the upper aftbody, and the lower aftbody. The shapes of these regions are described using super ellipses. The 23 parameters can be manipulated to provide a very good approximation to a majority of the hypersonic vehicle body shapes. Although the geometry package is unable to model some configuration aspects, it is a very efficient method for storing the geometry and supplying it to HAVOC. Any geometric modeling required for the proposed numerical method must be made compatible with this package.

The use of Computational Fluid Dynamics (CFD) will play an important role in any new hypersonic vehicle design. However, CFD can be costly and time consuming. Therefore, it is necessary to utilize a tool which can eliminate a number of design options, prior to conducting major CFD analyses. The incorporation of rigorous CFD methods with vehicle design optimization codes is rapidly becoming a reality as computational capabilities increase. However, before this can be achieved, a natural progression will take place, which will first incorporate simple numerical techniques and then eventually Navier-Stokes solutions, into the optimization process of the synthesis code. The first steps of this progression have already begun.

Analysis Approach

The search for possible analyses techniques begins with a review of different methods of lift and drag predictions for transonic flow. The initial method examined is a wave drag code based on the method of R.V. Harris [Ref. 2]. After the completion of this basic analysis, a more complete and reliable method was sought with the primary focus being on various CFD methods. The simplest application considered is the Wing-Body Code (WIBCO) which has an improved version capable of handling pod, pylon, and winglet analysis, called WIBCO-PPW [Ref. 3]. This code uses a modified transonic small disturbance (TSD) equation.

The WIBCO-PPW code initially seemed like an ideal candidate for the simple hypersonic vehicle shapes. The modified TSD equation is fairly easy to solve and the code uses a simple multiple nested grid. The grids are embedded on the configuration geometry thus eliminating the difficulty of generating a surface conforming grid. A drawback of the WIBCO-PPW code is that it is limited to cases with subsonic freestream Mach numbers. It also relies upon the small disturbance principle, and any significant deviation from that principle can severely affect the results. The code is quite capable of predicting lift but is less proficient at drag prediction. These limitations, combined with the fact that the code is not readily available, caused it to be passed over. It is still, however, considered a viable option which should be further examined before any final decisions are made.

The next code considered, TRANAIR, is similar to WIBCO-PPW in nature except that it employs the more robust full potential equation rather than the TSD equation, and is capable of handling supersonic freestream Mach numbers. The code is fairly new, although previous versions have been used since the mid 1980's. Also it is already installed and operating on the CRAY Y-MP supercomputer at NASA Ames. Like WIBCO-PPW, TRANAIR has automated grid generation. A major drawback of the TRANAIR code, however, is its the need to run on the CRAY Y-MP. The WIBCO-PPW can perform a wing-body analysis in about 20 minutes on an IBM 370/3081 [Ref. 3], while TRANAIR

could take over an hour for the same configuration on the CRAY Y-MP8/832. The use of TRANAIR generates a need to correlate results with wind tunnel data, and also with results from a more advanced analysis method.

An analysis technique more advanced than the full potential code, has to be either an Euler or a Navier-Stokes method. RANS3D, the chosen code, is a Navier-Stokes method with an Euler mode. It is used because it is available and running on the CRAY and a generated computational grid already exists for the all-body hypersonic configuration, which is the principal test case for this analysis. The Navier-Stokes results are used as a comparison for the full potential results and the wind tunnel data. The Euler mode is not used because the need for a bridge between the full potential results and the Navier-Stokes results is not needed. Since using the Navier-Stokes code for routine preliminary analyses is not a viable option, using an Euler code can also be ruled out for similar reasons. Although an increase in computation time resulting from using an Euler or a Navier-Stokes code is an important factor for their exclusion, the major reason is grid generation. Unlike the WIBCO-PPW and TRANAIR codes, grid generation in RANS3D is not automated. The generation of a computational grid is a very difficult and time consuming process which will greatly subtract from the performance of the code incorporated into a synthesis design code.

CHAPTER 2

Wave Drag Code Theory

Overview

The ability to numerically calculate the zero-lift wave drag of a wing-body aircraft, has been around since the 1960's [Ref. 3]. The basic method relies upon the supersonic area rule and Eminton and Lord's method [Ref. 4]. The version used in this analysis was developed and coded by the University of Georgia and NASA Langley [Ref. 5]. A major advantage of this version over previous ones is its ability to handle more complex geometry, in a less restrictive format. This allows analysis of geometries which have been constructed for use with other numerical (CFD) applications. The advantage of using a wave drag code instead of a CFD code, is the tremendous savings in computational time. The wave drag code does not require the use of a supercomputer. Runs were done on a Silicon Graphics IRIS workstation, with average CPU times of around 40 seconds.

In supersonic flow there are basically three types of drag. The first type is drag due to friction, which is caused by viscosity in the boundary layer. The second is drag due to lift, which is generated by the release of vortices and is called induced drag or sometimes vortex drag. Both frictional drag and induced drag are also present in subsonic flow. The third type of drag is wave drag, and it is only generated in supersonic flow. Wave drag is caused by pressure waves radiating energy away from the vehicle, similar to a fast moving ship generating waves in water [Ref. 6]. The sum of the wave drag and the vortex drag is called the pressure drag.

Supersonic Area Rule

The original application of the area rule, by Whitcomb [Ref. 7], was for transonic flow. In his analysis, Whitcomb considers a wing-body configuration and a series of parallel cutting planes normal to the aircraft axis. The intersection area of each cut is treated as an equivalent area circle. The combined equivalent areas define an equivalent body of revolution. Whitcomb proposed, and experimentally validated, that the wave drag of the wing-body configuration at Mach 1.0 is equal to the wave drag of the equivalent body.

For the supersonic case the area rule theory was modified by Jones [Ref. 8]. Instead of the parallel cuts being normal to the aircraft's axis, they are required to be inclined from the axis at the Mach angle μ. The resulting cut areas are then projected onto a plane normal to the aircraft's axis. There is no longer only one equivalent body of revolution since the cutting planes are not required to be normal to the aircraft's axis. The angle θ is the angle between the cutting plane and the y-axis, as shown in Figure 2.1. For each θ there is an equivalent body of revolution. The wave drag of the configuration is evaluated from the integrated average of the equivalent body wave drags obtained at incremental values of θ from 0° to 360°.

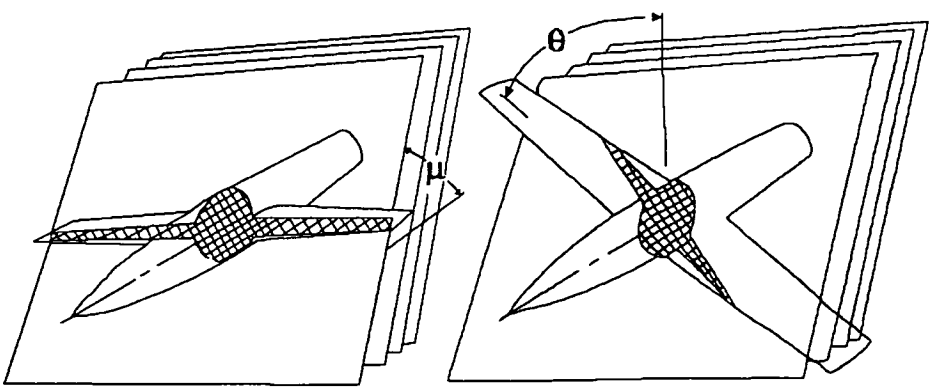


Figure 2.1. Orientation and Intersection of Cutting Mach Planes

The wave drag analysis is very simple compared to most numerical methods. The advantages of simplicity and computational speed are of course diminished when

considering its limitations. The area rule uses the slender body theory, hence blunt bodies or shapes which fall outside the Mach cone should be avoided. Also, the area rule assumes the configuration can be represented by a series of equivalent bodies of revolution, though a typical aircraft shape deviates significantly from a body of revolution. The results are that wave reflections caused by fuselage, wing, or tail interference are not accounted for. This should not be a significant factor for the body alone cases that will be used in this analysis. Another limitation is that the theory only provides the non-lifting wave drag. The wave drag due to lift must be calculated from another method [Ref. 6], as must the induced drag and the frictional drag.

Computational Method

Geometry Input

The input geometry format of this code is much more general than that of its predecessors. The input format requires that non-intersecting contours be used to describe the geometry. The contours are not required to be parallel or perpendicular to the x-axis, or to each other as was previously required. The aircraft can also be non-symmetrical with respect to the x-axis. Therefore axisymmetric configurations such as the oblique wing can be modeled. This is not always the case with many of the advanced numerical codes. The code requires that the vehicle's geometry broken down into components. For each component a separate set of geometry criteria is needed. The criteria includes the number of cross sections, the number of points per cross section, an option to reflect the component about the x-axis, a scale factor, and the origin of the component in relation to the aircraft's origin. Each component is also classified as either a fusiform or nonfusiform component. If the component is specified as fusiform then the cross sections must be orthogonal to the x-axis, but if it is nonfusiform then the cross sections may be rotated at any angle.

After the geometry is entered, the case data is read in. The case data contains the Mach number, the angle of attack, the x_r and z_r values about which the angle of attack is rotated, the number of equal intervals to divide the x-axis into, and the number of equal intervals to divide the domain of θ into. The angle of attack, α, is handled by rotating the entire geometry. The rotated x and z values are defined as

$$\begin{aligned} x' &= (x - x_r)\cos\alpha - (z - z_r)\sin\alpha + x_r \\ z' &= (x - x_r)\sin\alpha - (z - z_r)\cos\alpha + z_r. \end{aligned} \tag{2.1}$$

After the data is read in and rotated for angle of attack, a slope test is performed. In the slope test the program checks each body line segment for a slope which is larger than the slope of the angle of the Mach cone. A warning message is printed for each segment found in violation of this criterion, and it is left up to the user to decide whether or not to accept the results.

Computational Intervals

For each value of θ, an interval of the x-axis is determined which contains all Mach planes that intersect the aircraft. This interval is normally different for each θ. The equation of the Mach plane as a function of θ is given by

$$x - (\sqrt{M^2-1}\,\cos\theta)y - (\sqrt{M^2-1}\,\sin\theta)z = d, \tag{2.2}$$

where d is the x intercept of the Mach plane. The Mach plane is used to evaluate the intercept at both ends of the x interval. For each component, and for the entire aircraft, a minimum and a maximum x are calculated at each value of θ. These extrema are used to eliminate calculations being done outside of these limits.

Intercepted Areas

The next step in the solution process is determining the areas of the aircraft intercepted by the Mach plane for each value of θ. The total number of intervals, Nθ, that

the domain $-\pi/2$ to $3\pi/2$ is divided into, is specified by the user and must be divisible by four. The program begins at $\theta = -\pi/2$ and increments a $\Delta\theta$, n=1,2,3,...Nθ, until $\theta = 3\pi/2$ is reached. If the aircraft is symmetric then θ only proceeds to $\pi/2$. For a given value of θ, the x interval is divided into the user specified, NX number of subintervals starting at the minimum x intercept value d, and proceeding to the maximum d value. For a given angle, θ_i, and a given x interval value, x_j, the equation of the Mach plane, Equation (2.2), can be expressed as

$$x - (\sqrt{M^2-1}\,\cos\theta_i)y - (\sqrt{M^2-1}\,\sin\theta_i)z = x_j. \tag{2.3}$$

The next step is to use the input geometry contours to construct a polygon region from which the intersected area, $S(\theta_i,x_j)$, is calculated. There are two different ways this is done, depending on whether the component is fusiform or nonfusiform. If the intersected component is nonfusiform, then a series of points, falling on the defining line segments, are determined which represents the contour of the intersected area, Figure 2.2. This area, approximated by n points, (x_1,y_1), (x_2,y_2), ..., (x_n,y_n), is evaluated from

$$\begin{aligned} S = .5[&(x_1y_2 + x_2y_3 + \ldots + x_{n-1}y_n + x_ny_1) \\ &- (x_2y_1 + x_3y_2 + \ldots + x_ny_{n-1} + x_1y_n)]. \end{aligned} \tag{2.4}$$

The first point is used twice in order to close the contour.

When the intersected component is a fusiform type, the procedure for finding the intersected area is much easier. Because the cross sections of the fusiform component must be orthogonal to the x-axis, it is only necessary to determine the intersection of the Mach plane with the longitudinal lines. The code assumes that the cross section at the nose of the aircraft is extended upstream towards negative infinity and that the cross section at the base extends downstream towards positive infinity. What this does is allow the most forward point of a longitudinal line to be used if the Mach plane passes in front of it and likewise for the most aft point if the Mach plane passes behind it.

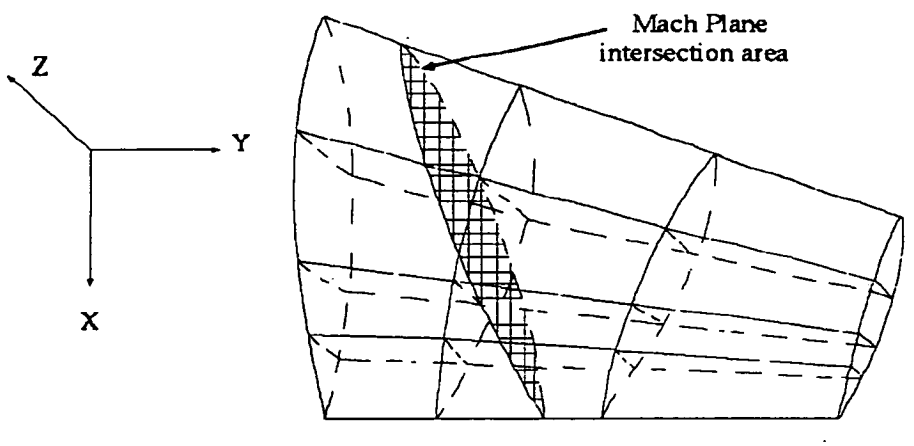


Figure 2.2. Intersection Contour and Area

Wave Drag Calculation

Once all values of $S_{total}(\theta_i,x_j)$ have been found for every x_j, the $D(\theta)/q$ for each θ is computed using the method of Eminton and Lord. After all the incremental values of $D(\theta)/q$ are obtained, where $D(\theta)$ is the wave drag at the angle θ, and q is the dynamic pressure, they are integrated as

$$\frac{D}{q} = \frac{1}{2\pi}\int_{-\pi/2}^{3\pi/2}\frac{D(\theta)}{q}d\theta, \tag{2.5}$$

to obtain the total non-lifting wave drag. This integral is evaluated using the 5 point Newton-Cotes formula.

CHAPTER 3

TRANAIR Computer Code Theory

Overview

TRANAIR [Ref. 9, 10] is a full potential panel code developed by the Boeing Company for NASA Ames. It is capable of providing transonic solutions for complex aircraft configurations at both subsonic and supersonic freestream Mach numbers. TRANAIR combines the ease and flexibility of a linear potential panel code, with the additional accuracy of the nonlinear full potential equation. This equation is especially useful for analyzing transonic flow with its highly nonlinear nature.

The paneling method is based on the linear potential panel code, PANAIR [Ref. 11]. The vehicle configuration is divided into networks of surface panels. The global computational grid is superimposed on the surface geometry by the code, as illustrated in Figure 3.1. This eliminates the time consuming and difficult process of surface fitting a grid. The solution is obtained on a sequence of grids that are adaptively constructed based upon solution errors and user inputs. The final output of desired flow quantities is given at each panel corner point and or panel center point if desired. Integrated values of the aerodynamic forces and moments are also summarized in the output.

TRANAIR is currently capable of handling cases with up to 30,000 surface panels and 450,000 global grid points. The TRANAIR runs were performed on the CRAY Y-MP8/832 at Ames. Average CPU times varied from 40 minutes to over 90 minutes, depending on the configuration, freestream Mach number, angle of attack, and gridding

option. The following theoretical background of TRANAIR is a summarized version of that appearing in the TRANAIR Theory Document [Ref. 9].

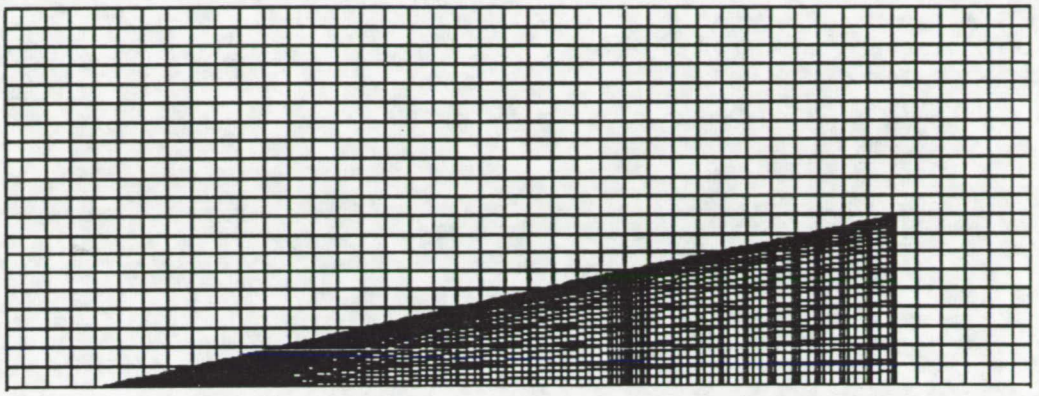


Figure 3.1. Surface Panels Embedded on Global Grid

Boundary Value Problem

Governing Equation

The full potential equation is very useful for solving transonic flow. In the hierarchy of fluid dynamics equations, the full potential equation is a step down in simplification from the Euler equations which are below the Navier-Stokes equations. The full potential equation is given as

$$\vec{\nabla} \cdot \rho \vec{\nabla} \Phi = 0, \tag{3.1}$$

where the isentropic density and pressure are defined as

$$\rho = \rho_\infty \left[1 + \frac{\gamma - 1}{2} M_\infty^2 \left(1 - \frac{q^2}{q_\infty^2}\right) \right]^{\frac{1}{\gamma - 1}}, \tag{3.2}$$

$$p = p_\infty \left[1 + \frac{\gamma - 1}{2} M_\infty^2 \left(1 - \frac{q^2}{q_\infty^2}\right) \right]^{\frac{\gamma}{\gamma - 1}}, \tag{3.3}$$

respectively. In the preceding equations, the total velocity potential is Φ, the local speed is $q = |\vec{\nabla}\Phi|$, the freestream speed is $q_\infty = |\vec{V}_\infty|$, the freestream density is ρ_∞, the freestream pressure is p_∞, the freestream Mach number is M_∞, and the ratio of specific heats is γ. Equation (3.1) is valid for inviscid irrotational compressible flow and it is in conservative form, with mass the quantity being conserved. The full potential equation does not allow for changes in entropy across shocks. This restriction seems to make the full potential equation a poor choice for analyzing transonic and supersonic flow, but it actually does well while the normal Mach number is close to unity [Ref. 12]. This restriction on the normal Mach number refers only to the normal component of the local Mach number and does not necessarily apply to the freestream Mach number.

Boundary Conditions

The proper application of boundary conditions in a CFD problem is critical. In TRANAIR, the far field boundary condition states, the perturbation potential tends to zero as the distance from the vehicle's surface increases. Mathematically, this is

$$\phi = O\left(\frac{1}{R}\right) \tag{3.4}$$

as $x \rightarrow \infty$, where $\phi = \Phi - \Phi_\infty$ is the perturbation potential.

On the vehicle's surface there are several choices for boundary conditions. The normal mass flux is given as

$$\rho \frac{\partial \Phi}{\partial n} = g_1, \tag{3.5}$$

where n represents the direction normal to the surface. For the majority of cases the surface is considered impermeable and g_1 becomes zero. A case where g_1 would not be zero is on an engine inlet.

It is also possible to specify a Dirichlet boundary condition

$$\Phi = g_3, \tag{3.6}$$

where g_3 is a constant. This condition prohibits tangential flow along the boundary surface.

An important category of boundary conditions are wakes. Wakes are extended from the trailing edges of all lifting surfaces. The wakes allow for nonzero circulation in potential flow. The wake cut boundary conditions are

$$\hat{n} \cdot \Delta(\rho \vec{\nabla} \Phi) = 0 \tag{3.7}$$

and

$$\Delta p = 0, \tag{3.8}$$

where p is given in Equation (3.3), $\hat{n}$ is the unit normal vector to the wake cut, and Δ represents the jump across the wake surface. Equation (3.7) represents the conservation of mass across the wake surface and Equation (3.8) insures conservation of normal momentum. Equation (3.8) can be linearized about the freestream pressure, $p = p_\infty$, by assuming a small perturbation velocity $\vec{\nabla}\phi$. This leads to the Dirichlet condition that the flow along the wake is tangential to the wake and is in the freestream direction.

Bateman Variational Principle

The full potential equation can also be derived from the Bateman variational principle [Ref. 13]. The Bateman principle is used to derive the full potential finite element

formulas. The principle states that the integral of pressure over the flow field is stationary. A variation of the integral

$$J = \int_{\Omega} p d\Omega \tag{3.9}$$

is taken and combined with

$$\frac{\partial p}{\partial \vec{V}} = -\vec{W} \equiv -\rho \vec{V}, \tag{3.10}$$

where $\vec{W}$ is the mass flux vector. The resulting equation is

$$\delta J = \int_{\Omega} -\vec{W} \cdot \delta \vec{V} d\Omega. \tag{3.11}$$

Integration by parts yields

$$\delta J = \int_{\Omega} \vec{\nabla} \cdot \vec{W} \delta \Phi d\Omega - \int_{\Sigma} \hat{n} \cdot \vec{W} \delta \Phi d\Sigma, \tag{3.12}$$

where Σ is the boundary of the domain or a surface of discontinuity. If J is stationary with respect to arbitrary variations in Φ, then the first integral in Equation (3.12) becomes

$$\vec{\nabla} \cdot \vec{W} = 0, \tag{3.13}$$

which is the conservation of mass equation and is identical to Equation (3.1). The second integral in Equation (3.12) concerns conservation across surfaces of discontinuity. Across a shock, where mass is continuous, the integral will become

$$\Delta(\hat{n} \cdot \vec{W}) = 0. \tag{3.14}$$

These boundary conditions can be incorporated into the Bateman variational principle. The resulting principle then states that the functional

$$J = \int_{\Omega} p dV + \int_{\partial\Omega_1} g_1 \Phi dS - \int_{\partial\Omega_2} \alpha(\rho \frac{\partial \Phi}{\partial n})(\Delta\Phi - \mu) dS + \int_{\partial\Omega_3} \rho \frac{\partial \Phi}{\partial n}(\Phi - g_3) dS, \quad (3.15)$$

is stationary. In Equation (3.15) g_1 is the mass flux on the domain $\partial\Omega_1$, $\Delta\Phi$ is the jump in Φ across the wake surface $\partial\Omega_2$, μ represents the unknown jump in ϕ on $\partial\Omega_2$, determined from Equation (3.8), α is the average of the upper wake surface and lower wake surface values, g_3 is the potential on $\partial\Omega_3$, and S is entropy.

Variations in Total Properties

It is possible to modify the potential flow simulation to handle regions of differing total temperature and total pressure. The flow in each region is still potential as long as the total temperature and total pressure are constant in that region. In order to model those regions the density is redefined as

$$\rho = \rho_\infty \frac{r_P}{r_T}\left[1 + \frac{\gamma - 1}{2} M_\infty^2 (1 - \frac{q^2}{q_\infty^2 r_T})\right]^{\frac{1}{\gamma - 1}}, \quad (3.16)$$

and the pressure as

$$p = p_\infty r_P\left[1 + \frac{\gamma - 1}{2} M_\infty^2 (1 - \frac{q^2}{q_\infty^2 r_T})\right]^{\frac{\gamma}{\gamma - 1}}, \quad (3.17)$$

where r_P is the ratio of total pressure in the region to the freestream total pressure, and r_T is the ratio of total temperature in the region to the freestream total temperature. The different regions are separated by wake surfaces, across which jumps in the boundary conditions occur. The first boundary condition is static pressure continuity, Equation (3.8). If the

total property difference across the wake is large, then Equation (3.8) cannot be linearized. The second boundary condition is similar to Equation (3.7), but is modified to better suit the areas where total pressure and temperature differences are large. This condition is

$$\hat{n} \cdot \Delta \vec{W}^* = 0, \tag{3.18}$$

where

$$\vec{W}^* = \frac{\rho_\infty q_\infty}{\rho_0 q_0} \rho \vec{\nabla} \Phi. \tag{3.19}$$

In the given region, q_0 is the speed at which $p = p_\infty$ and ρ_0 is the corresponding density. The Bateman principle is modified so that

$$J = \int_\Omega p^* dV, \tag{3.20}$$

where

$$p^* = p \frac{\rho_\infty q_\infty^2}{\rho_0 q_0^2}. \tag{3.21}$$

This modification allows the modeling of engine exhaust assuming that the exhaust is divided into separate regions each with a constant value of entropy.

Discretization

Computational Domain

The computational domain of TRANAIR is a rectangular finite region of space. To show that a finite domain can be used, let the partial differential operator $\mathcal{F}$ be equal to a constant coefficient differential operator $\mathcal{T}$ everywhere outside the finite computational

domain. Then let $\mathcal{G}$ be a Green's function so that $\mathcal{T}(\mathcal{G}*Q) = Q$ for all unknown quantities, Q, and $\Phi = \mathcal{G}*Q + \Phi_\infty$ satisfies the far field boundary condition. The full potential equation, $\mathcal{F}\Phi = 0$ then becomes

$$Q + (\mathcal{F} - \mathcal{T})(\mathcal{G} * Q + \Phi_\infty) = 0. \tag{3.22}$$

By definition, $\mathcal{F} = \mathcal{T}$ outside of the computational domain, thus Equation (3.22) reduces to $Q = 0$, which shows that the unknown quantities are restricted to the computational domain. For the full potential equation, the far field operator $\mathcal{T}$ is

$$\mathcal{T}\Phi = (1 - M_\infty^2)\Phi_{xx} + \Phi_{yy} + \Phi_{zz}, \tag{3.23}$$

which is the Prandtl-Glauret operator. Equation (3.23) is the full potential equation linearized about $\vec{V}_\infty$.

The size of the computational domain can be kept relatively small, because away from the boundary surfaces Q tends to approach zero much faster than Φ approaches the freestream velocity potential, Φ_∞. Thus the computational domain need only enclose the configuration and any areas of nonlinear flow. Trailing wakes are extended to infinity by simply terminating them just beyond the downstream boundary of the computational domain. The downstream edge of the domain is spaced farther away from the surface boundary to allow computations to be done on the trailing wake, and thus account for additional lift.

The continuous operators can be replaced by discrete ones and the same reasoning will hold. The discrete version of $\mathcal{T}$ is T, and it requires that a discrete Green's function G, exist that satisfies, T(G*Q) = Q for all Q, and also satisfies the discrete far field condition. The computational domain need only include nonlinear flow regions and regions where L, the discrete version of $\mathcal{F}$, cannot be suitably approximated by the discrete operator T.

Computational Grid

A central feature of TRANAIR is its ability to automatically generate the three-dimensional computational grid. Also advantageous, is the ability to chose which gridding method to use, grid sequencing or adaptive gridding. The surface grid and an initial global grid are the only input grids required for TRANAIR. The global grid is used as a starting point for both the grid sequencing and the solution adaptive grid methods. The initial global grid is very coarse, and is refined until either the maximum number of grids is used or the maximum number of grid cells is reached. Addition of cells, or refinement, is done by dividing a single cell into eight smaller regions of equal volume.

Grid refinement follows two criteria. The first criterion is based on the size of the surface panels. The global grid boxes neighboring the surface panels are refined according to a panel length weighting scale, which is the panel diameter multiplied by a user specified panel tolerance factor. The second criterion for refinement is determined from the size of the grid box. Restrictions are imposed by specifying a dx_{min} and a dx_{max}, between which the size of the box must fall. No refinement or derefinement may occur which will create a grid cell with a dx length beyond these limiting values.

Special regions can be defined which reset one or both of the criteria to provide either more or less refinement. Thus, regions such as expected shock locations can be emphasized while regions such as wake edges can be de-emphasized. The use of special regions is important since a total number of target grid cells is specified and the best use of these cells is made if the code is given some help deciding where the important and the not so important areas are.

With the grid sequencing method the final grid is generated first and is derefined to give a sequence of grids ranging from coarse to fine. The iterative solution is done on the coarsest grid first, and then proceeds sequentially to the finest grid. In the adaptive method the coarsest grid is used first and the following grids are created based upon estimated local

errors. In both cases the initial values are zero and the starting values are interpolated from the previous grid solution.

Finite Element Method

The finite element method is based on a standard seven point operator for Poisson's equation on a uniform grid. A typical finite element box, with its eight unknown corners, is shown in Figure 3.2.

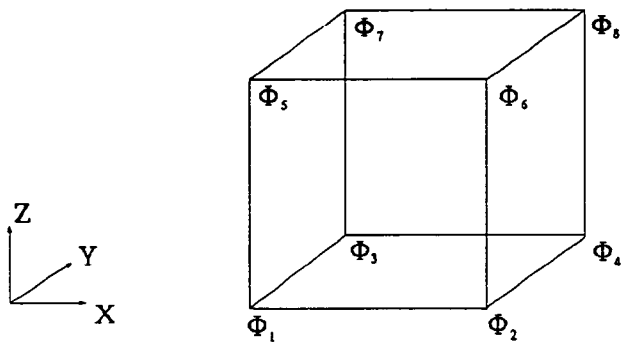


Figure 3.2. Finite Element Box

The seven point operator, represented in Figure 3.3, contains eight finite element boxes. The seven point operator is used because it reduces the size of the stiffness matrices compared to using a nine point operator.

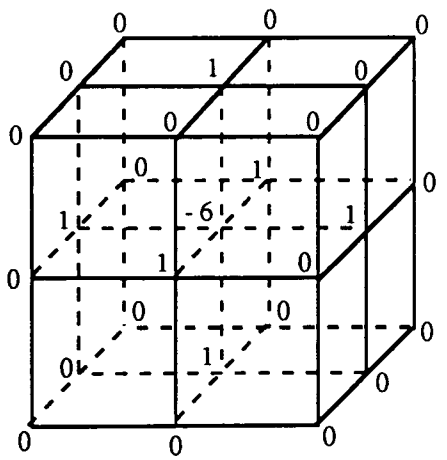


Figure 3.3. Seven Point Laplacian Operator

The stiffness matrices are generated by taking variations of J with respect to each degree of freedom. From Equation (3.11) the variation of J can be presented as

$$\begin{aligned}\delta J &= -\int_{\Omega} \rho \vec{\nabla}\Phi \cdot \vec{\nabla}\delta\Phi dV \\ &= -\sum_i \int_{\Omega_i} \rho \vec{\nabla}\Phi \cdot \vec{\nabla}\delta\Phi dV \\ &\cong -\sum_i \rho_i \int_{\Omega_i} \vec{\nabla}\Phi \cdot \vec{\nabla}\delta\Phi dV ,\end{aligned} \tag{3.24}$$

where ρ_i is the density at the center of the region Ω_i. There are three types of elements into which the computational domain can be divided for discretizing purposes. They are near field boxes, far field boxes, and boundary boxes.

Boxes which are not cut by a boundary surface and where $L \neq T$, are called near field boxes. Equation (3.24) defines the element stiffness matrices by taking the variations of J with respect to the eight corners of the element. For the near field boxes all of the stiffness matrices are identical except for a constant factor that depends upon the level of refinement and the centroid density value, ρ_i. Because of the similarity in stiffness matrices, large amounts of storage space can be saved. Also the discrete velocity formula taken at the center of each element is the same for each near field region resulting in additional savings of storage space.

Far field boxes are defined as those boxes which fall on the computational domain boundaries, where $L \equiv T$. These boxes remain unrefined, and thus are geometrically identical. The density in these boxes is a constant value because the linear flow properties are matched. As with the near field boxes, the discrete operators are identical.

Boxes which are cut by boundary surfaces are labeled boundary boxes. The region in a boundary box which does not lie in the interior of the configuration is called a D-region. D-regions are not defined in the interior of the configuration because the boundary surfaces signify stagnation flow, $\phi = 0$, in the interior. It is possible to have

more than one D-region per boundary box as is illustrated in Figure 3.4. Here a wake dividing an element creates regions D_1 and D_2 in the same boundary box.

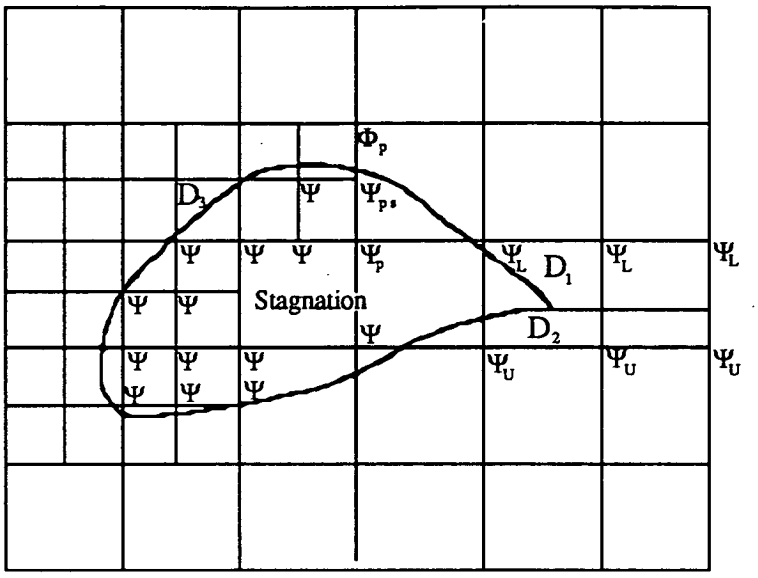


Figure 3.4. Boundary Region

Near boundary surfaces discontinuities can arise, so it is important to have a separate element trial function for each D-region. The element trial function is parameterized by the unknown corner points outside of the interior, and inside the interior by extrapolated values, represented in Figure 3.4 by Ψ. Each D-region also has a unique stiffness matrix and velocity operator. These element stiffness matrices are derived from an expanded form of Equation (3.24).

Grid Interfaces

To insure conservation of mass, the element trial functions must be continuous from box to box. To implement this, psuedo-unknowns are introduced. A psuedo-unknown is an unknown quantity at an element's node which is not a corner. This occurs in interface areas where the levels of grid refinement differ. In Figure 3.4, Ψ_{ps} is a psuedo-unknown, Φ_p and Ψ_p are referred to as its parents. To maintain continuity of the

element trial functions the psuedo-unknown, Φ_1, must be the average of its parents Φ_2 and Φ_3, that is

$$\Phi_1 = \frac{1}{2}(\Phi_2 + \Phi_3). \tag{3.25}$$

This principle is also applied in three dimensions where the psuedo-unknown occurs on an element face and its value is determined from the four corner parents. The governing equations produce residuals at the psuedo-unknown which are then equally distributed to the four parent unknowns.

Stability Considerations

To achieve numerical stability the Bateman variational principle is modified. The last integral in Equation (3.15) enforces a Dirichlet condition. The resulting finite element formation is partially unstable. To remedy this the last integral of Equation (3.15) is replaced by the integral

$$\int_{\partial\Omega_3}\left[\rho\frac{\partial\Phi}{\partial n}(\Phi - g_3) - \frac{\rho}{2\Delta l}(\Phi - g_3)^2\right]dS, \tag{3.26}$$

where Δl is the minimum diameter of the box containing the element trial function.

Another stability problem arises because surfaces are represented by flat panels. The discontinuities in slope from panel to panel can propagate into the flow field. To eliminate this a curved surface is simulated by adding to Equation (3.24) the surface integral

$$\partial J = \partial J + \int_{\partial\Omega_1}\rho\vec{\nabla}\Phi\cdot(\hat{n} - \hat{n}^*)\partial\Phi dS, \tag{3.27}$$

where $\hat{n}^*$ is a polynomial interpolation of $\hat{n}$, and $\partial\Phi$ is the variation of Φ. Intended discontinuities are still allowed because the end points of the interpolation are user specified inputs.

Dissipation

Artificial viscosity is employed when supersonic flow is present, to dissipate the gradients and prevent the code from "blowing up". TRANAIR uses a first order upwind density scheme as an artificial viscosity model. The density in the full potential equation is replaced with

$$\tilde{\rho} = \rho - \mu \hat{V} \cdot \vec{\Delta}_{-}\rho, \tag{3.28}$$

where $\hat{V}$ is the normalized local velocity, $\vec{\Delta}_{-}\rho$ is an upwind undivided difference, and μ is a switching function defined as

$$\mu = max\begin{pmatrix} 1 - M_c^2 / M^2 \\ 0 \end{pmatrix}, \tag{3.29}$$

where M is the local Mach number and M_c is the cut-off Mach number. The cut-off Mach number is used to initiate dissipation just below Mach 1.0. The value chosen is $M_c^2 = 0.95$.

TRANAIR also offers flux biasing as an alternative to density biasing. Instead of upwinding the density ρ, the flux ρq is upwinded, where $q = \| \tilde{\nabla} \|_2$. The form of the flux biasing, similar to that for density, is

$$\tilde{\rho} = \frac{1}{q}\left((\rho q) - \hat{V} \cdot \vec{\Delta}_{-}(\overline{\rho q})\right), \tag{3.30}$$

where

$$\overline{\rho q} = \begin{cases} \rho q & M > 1 \\ \rho^* q^* & M \le 1 \end{cases} \tag{3.31}$$

and $\rho^* q^*$ is the value of ρq at M = 1.0.

Both the density biasing and the flux biasing methods are first order, which sometimes creates reliability problems and decreases the accuracy and efficiency. The problems occur because the biasing methods are first order, and the remainder of the differencing schemes in the code are second order. A second order method is not employed, because historically the performances of such algorithms are lacking, especially for complex geometries.

Algorithms

Linear Algorithm

The intended application of TRANAIR is for nonlinear problems, but it is also capable of performing a completely linear analysis. The general form of the linear potential equation is

$$\vec{\nabla} \cdot (\rho \vec{\nabla} \Phi) = f, \tag{3.32}$$

where the density is assumed known and positive. For Equation (3.32) to satisfy the far field boundary condition, Equation (3.4), source unknowns Q must replace the unknowns Φ on the global grid. On the boundary of the global grid, the value of Q is zero. Extrapolated values of the velocity potential in boundary boxes are denoted by Ψ, and all other velocity potential unknowns are denoted by Φ. On wake surfaces, the doublet parameters are denoted by μ. The finite element operator, L, is defined over the entire global grid except on the boundary. The operator is evaluated by multiplying the element stiffness matrices by the vector of unknowns. The linear system of equations can be represented as

$$L\begin{pmatrix} T^{-1}Q \\ \Phi \\ \Psi \\ \mu \end{pmatrix} = f\,, \tag{3.33}$$

where T is the standard far field operator.

Generally this system of equations will, depending on boundary conditions, be non-symmetric. To solve this system of non-symmetric linear equations, TRANAIR uses the GMRES (Generalized Minimal RESidual) method [Ref. 14]. To achieve greater convergence rates with GMRES, it is necessary to alter the distribution of eigenvalues. The more the eigenvalues are clustered, the faster GMRES will converge. The process of replacing the distribution of eigenvalues with a more favorable one while maintaining the same solution is known as preconditioning.

A simple example of preconditioning uses an approximate inverse to the operator that is in the equation to be solved. Consider the linear operator $\mathcal{L}$ in the matrix equation

$$\mathcal{L}(x) - b = 0. \tag{3.34}$$

If the approximate inverse to $\mathcal{L}$ is $\mathcal{N}^{-1}$, then Equation (3.34) is equivalent to

$$\mathcal{N}^{-1}(\mathcal{L}(x) - b)) = 0. \tag{3.35}$$

In this example $\mathcal{N}$ is the preconditioner for $\mathcal{L}$. Equation (3.34) and Equation (3.35) have the same solution, but GMRES will solve Equation (3.35) faster because the eigenvalues are more clustered.

For the system of equations represented in Equation (3.33), T^{-1} is used as a right preconditioner for the global grid points. The operator T^{-1} is defined over the uniform global grid and can be applied using the Poisson Solver with very rapid results. A left

preconditioner is also required to approximate the problem near the internal boundary. This preconditioner, N, is defined to be a reduced set of global stiffness matrix unknowns. The reduced set contains the unknown corner points of boundary boxes, refined boxes, and boxes containing differences in total pressure or total temperature from the freestream values. To solve the reduced set, it is necessary to use a closure set of unknowns which is part of the stiffness matrix but outside of the reduced set. The boundary condition for the closure unknowns is the far field condition, $\phi = 0$.

The global preconditioner, T^{-1}, and the reduced set preconditioner, N^{-1}, contain overlapping values of the global grid unknowns, Q. This forces the preconditioner T to be applied on the left. The entire preconditioned equation can then be written as

$$T\,N^{-1}(f - LT^{-1}X) = 0, \tag{3.36}$$

where

$$X = \begin{pmatrix} Q^{(1)} \\ Q^{(2)} \\ \Phi \\ \Psi \\ \mu \end{pmatrix}. \tag{3.37}$$

In Equation (3.37), $Q^{(1)}$ is the set of global grid unknowns which are not in the reduced set or in stagnation regions and $Q^{(2)}$ is the set of global grid unknowns which are in the reduced set or in stagnation regions.

Nonlinear Algorithm

The nonlinear algorithm is the backbone of TRANAIR. With this algorithm it is necessary to use the Newton method which solves a linear problem, similar to that for the linear algorithm. The Newton method begins with a system of nonlinear equations

$$F(x) = 0, \tag{3.38}$$

and an initial trial solution, x^0. For n = 0, 1, 2, ... until a sufficiently small residual is obtained, let

$$x^{n+1} = x^n + \lambda(\delta x^{n+1}), \tag{3.39}$$

where δx^{n+1} is the solution of the linear system

$$\bar{F}_{x^n}(\delta x^{n+1}) = -F(x^n), \tag{3.40}$$

and λ is a step length, chosen so that

$$\| F(x^{n+1}) \| < \| F(x^n) \|. \tag{3.41}$$

In Equation (3.40), $\bar{F}_{x^n}$ is the Jacobian of F linearized about x^n and is defined acting on a vector y as

$$\bar{F}_x(y) = \lim_{\varepsilon \to 0} \frac{F(x + \varepsilon y) - F(x)}{\varepsilon}. \tag{3.42}$$

The GMRES algorithm is used to solve Equation (3.40). The preconditioning for this equation is the same as for the linear system and given in Equation (3.36). For the nonlinear case the reduced set, used in the linear case, is expanded to include the elements where upwinding is used.

The nonlinear algorithm relies on Newton's method, which converges rapidly when the initial value is close to the solution, and for problems with weak or no shocks. For other problems which contain strong shocks or where the initial conditions can be far from the final solution, damping must be employed to enhance convergence and prevent divergence. Viscosity damping can be used to improve convergence in the presence of shock waves. During the initial steps of Newton's method the amount of artificial viscosity

decreased. This is repeated until the desired dissipation level is reached. The results obtained using viscosity damping are good but the drawback is an increase in computational time.

Grid Sequencing Method

The grid sequencing method is one of two available options that can be used with the standard Newton damping or viscosity damping. In this method a sequence of coarse to fine grids is created prior to any computations being done. The solution is found on the coarsest grid and then interpolated to the next grid. The solution is then found on that grid and the process is repeated until the final grid is reached. Because the initial grid cell size is large the dissipation is also large. As the cell size is decreased by going to finer grids, the dissipation is automatically reduced. The grid sequencing method is fairly reliable and uses less computer time but results are dependent upon initial grid spacing.

Solution Adaptive Grid Method

The alternative to grid sequencing is the solution adaptive grid method. Unlike the grid sequencing method which constructs all of the grids before performing any calculations, the adaptive method begins with the coarsest grid and constructs the next grid based upon computed residual errors. The adaptive method not only allows refinement, but also derefinement. When a cell is refined it is divided into eight identical cells. When derefinement occurs, eight identical cells are combined to form one new cell.

The target for the adaptive grid method is a final grid with a specified number of grid cells, N. To obtain as accurate a solution as possible with N elements, five steps are followed for each grid created. The five steps in order are, estimating local error, computing local error predictors, applying *a priori* grid refinement controls, applying grid refinement strategy, and constructing the new grid.

The estimated local error is computed from the variation in the velocity components of each grid element. The error is computed from

The estimated local error is computed from the variation in the velocity components of each grid element. The error is computed from

$$\text{errest} \equiv \max_{\text{region } r} \{\max_{j} \{(\Delta v_1^{r,j})^2 + (\Delta v_2^{r,j})^2 + (\Delta v_3^{r,j})^2\}^{\frac{1}{2}}\}, \tag{3.43}$$

where for the rth solution region contained in the element, Δv_i^{rj} is the difference in velocity across the element's jth face for each directional component. The first maximum is taken over the regions contained in the element. The second maximum is taken over the element faces connected to a region that is not part of a larger element.

Once the local error estimates have been made, the local error predictors are computed by smoothing the error estimates. Grid refinement is implicitly predicted for areas near elements with large detected errors. The refinement covers these regions and one or two additional elements to prevent any refinement regions from being missed.

The ability to control the grid refinement by specifying *a priori* controls is very important to the adaptive grid method. Without refinement controls the code treats all regions with equal estimated errors the same. Common areas where heavy refinement can occur are leading edges, wing tips, wakes, and at irregular shapes in geometry. The user will not always be equally interested in each of these regions. Since only a given number of target cells may be used, areas which are of little interest can draw away a significant number of cells from areas of greater importance. The way refinement is controlled is through user specified hexahedral regions called LBO's. For each LBO region, a minimum and a maximum grid size as well as a weighting factor are specified which override the global parameters.

Grid refinement strategy is the process of determining exactly which elements to refine or derefine. Refinement of eligible cells is based upon the magnitude of the scaled error predictor, with cells over the maximum size limit having highest priority. Similarly, derefinement occurs on cells with the smallest error predictors. To keep refinement from

developing too rapidly, TRANAIR incorporates a simple refinement strategy. The first principle allows refinement and derefinement on coarse and intermediate grids of a fixed percentage of grid elements. The second part of the strategy is to distribute local errors somewhat equally while maintaining about the same number of elements for intermediate grids. The last part is refining only on the last grid.

A new grid is constructed by using the marked elements and a grid legalization constraint. The grid legalization requires neighboring elements to be refined or derefined so that elements sharing the same edge do not differ by more than one level of refinement. If flow near the outer global boundaries is nonlinear, grid refinement can actually enlarge the computational domain to include any significant nonlinear regions.

Supersonic Freestream

When the freestream flow is supersonic the governing equation switches from elliptic to hyperbolic. The method of solving the problem remains unchanged except for how the far field condition is handled and how the discrete equations are solved.

The hyperbolic characteristics of supersonic flow require the upstream boundary to have initial values. The initial perturbation values on the upstream boundary and the side boundaries are taken as zero. Normally because the method marches downstream, boundary conditions are not required at the outflow boundary. But because of the reduced set, some sort of condition is required there, and this is chosen to be $\phi_{xx} = 0$. The current method of handling hyperbolic flow by imposing outer boundary conditions can have the disadvantage of shock waves being reflected back into the flow field. This is especially true at low supersonic Mach numbers where the Mach angles are large.

In supersonic flow calculations the use of the solution adaptive method provides superior results over grid sequencing [Ref. 15]. The main reason for this is the shock capturing ability of the adaptive method. Using the grid sequencing method tends to smear the shock over several grid boxes while the adaptive method can more readily conform to

handle the shock location as well as other important flow effects. It is also known that a more reliable convergence is achieved if viscosity damping is used on each grid in the adaptive method.

CHAPTER 4

RANS3D Computer Code Theory

Overview

RANS3D is a Reynolds-averaged time dependent Navier-Stokes code currently under development at NASA Ames by Gary Cosentino and Scott Thomas [Ref. 16]. It is based upon the widely used ARC3D code also developed at Ames [Ref. 17]. The code uses the thin-layer approximation in the normal, ζ, direction with the Baldwin and Lomax algebraic turbulence closure model [Ref. 18]. The one-equation turbulence model of Baldwin and Barth is also available, but has not been extensively tested with the code. The original diagonal central difference algorithm in ARC3D, is replaced by the Lower-Upper factored Alternating-Direction-Implicit (LU-ADI) upwind scheme of Fujii and Obayashi [Ref. 19]. The right-hand-side (RHS) Euler flux terms can be computed by using either a central differencing scheme or an upwind differencing scheme. RANS3D is also capable of performing inviscid flow, Euler, calculations.

RANS3D is run on the CRAY Y-MP at NASA Ames. The code contains CRAY microtasking directives which allow an efficient use of the multiple processing capability of the CRAY. Utilizations of 98%, using all 8 CPUs, are obtainable.

Governing Equations

The nondimensional thin-layer approximated governing equations are written using the body conforming coordinates, ξ, η, and ζ. The equations are

$$\partial_t \hat{Q} + \partial_\xi \hat{E} + \partial_\eta \hat{F} + \partial_\zeta \hat{G} = Re^{-1} \partial_\zeta \hat{S} \tag{4.1}$$

where

$$\hat{Q} = J^{-1} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \qquad \hat{E} = J^{-1} \begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ U(e+p) - \xi_t p \end{bmatrix},$$

$$\hat{F} = J^{-1} \begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ V(e+p) - \eta_t p \end{bmatrix}, \qquad \hat{G} = J^{-1} \begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ W(e+p) - \zeta_t p \end{bmatrix}, \tag{4.2}$$

$$\hat{S} = J^{-1} \begin{bmatrix} 0 \\ \mu(\zeta_x^2 + \zeta_y^2 + \zeta_z^2) u_\zeta + \frac{\mu}{3}(\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta)\zeta_x \\ \mu(\zeta_x^2 + \zeta_y^2 + \zeta_z^2) v_\zeta + \frac{\mu}{3}(\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta)\zeta_y \\ \mu(\zeta_x^2 + \zeta_y^2 + \zeta_z^2) w_\zeta + \frac{\mu}{3}(\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta)\zeta_z \\ [(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)\{\frac{1}{2}(u^2 + v^2 + w^2)_\zeta + Pr^{-1}(\gamma - 1)^{-1}(a^2)_\zeta\} \\ + \frac{\mu}{3}(\zeta_x u + \zeta_y v + \zeta_z w)(\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta)] \end{bmatrix},$$

and the contravariant velocities U, V, W are

$$\begin{aligned} U &= \xi_t + \xi_x u + \xi_y v + \xi_z w \\ V &= \eta_t + \eta_x u + \eta_y v + \eta_z w \\ W &= \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w \end{aligned} \tag{4.3}$$

and J is the Jacobian. The pressure is related to the density, velocity components, and energy by

$$p = (\gamma - 1)[e - \frac{1}{2}\rho(u^2 + v^2 + w^2)]. \tag{4.4}$$

The metric terms required for Equations (4.1) are

$$\begin{aligned}
\xi_x &= J(y_\eta z_\zeta - y_\zeta z_\eta), & \eta_x &= J(y_\zeta z_\xi - y_\xi z_\zeta) \\
\xi_y &= J(x_\zeta z_\eta - x_\eta z_\zeta), & \eta_y &= J(x_\xi z_\zeta - x_\zeta z_\xi) \\
\xi_z &= J(x_\eta y_\zeta - x_\zeta y_\eta), & \eta_z &= J(x_\zeta y_\xi - x_\xi y_\zeta) \\
\zeta_x &= J(y_\xi z_\eta - y_\eta z_\xi), & \xi_t &= - x_t\xi_x - y_t\xi_y - z_\tau\xi_z \\
\zeta_y &= J(x_\eta z_\xi - x_\xi z_\eta), & \eta_t &= - x_\tau\eta_x - y_\tau\eta_y - z_\tau\eta_z \\
\zeta_z &= J(x_\xi y_\eta - x_\eta y_\xi), & \xi_t &= - x_\tau\zeta_x - y_\tau\zeta_y - z_\tau\zeta_z
\end{aligned} \tag{4.5}$$

and

$$J^{-1} = x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi - x_\eta y_\zeta z_\xi - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi. \tag{4.6}$$

Turbulence Model

The principal turbulent model used in RANS3D, is the Baldwin and Lomax two-layer algebraic eddy viscosity model. The turbulent eddy viscosity, μ_t, is determined by examining the vorticity magnitude of the local flow and then determining a length scale. This length scale will then give the turbulent eddy viscosity. The sum, $\mu + \mu_t$, is substituted for values of μ in Equations (4.1). The other flow modeling options for RANS3D, are laminar flow and the Baldwin and Barth one-equation turbulence model, both of which have not been extensively tested with the code.

Numerical Algorithm

LU-ADI Algorithm

The LU-ADI algorithm developed by Fujii and Obayashi [Ref. 19], is an implicit scheme that simplifies inversion work for the left-hand-side (LHS) operators of the commonly used Beam-Warming method [Ref. 20]. The implicit Beam-Warming scheme applied to Equations (4.1) is given by

$$\begin{aligned}(I+h\delta_\xi\hat{A}^n-\varepsilon_i J^{-1}\nabla_\xi\Delta_\xi J)(I+h\delta_\eta\hat{B}^n-\varepsilon_i J^{-1}\nabla_\eta\Delta_\eta J)\cdot\\ (I+h\delta_\zeta\hat{C}^n-hRe^{-1}\delta_\zeta\hat{M}^n-\varepsilon_i J^{-1}\nabla_\zeta\Delta_\zeta J)(\hat{Q}^{n+1}-\hat{Q}^n)=\\ -\Delta t(\delta_\xi\hat{E}^n+\delta_\eta\hat{F}^n+\delta_\zeta\hat{G}^n-Re^{-1}\delta_\zeta\hat{S}^n)-\varepsilon_e J^{-1}[(\nabla_\xi\Delta_\xi)^2+(\nabla_\eta\Delta_\eta)^2+(\nabla_\zeta\Delta)^2]J\hat{Q}^n\end{aligned} \tag{4.7}$$

where $h = \Delta t$, δ is the central difference finite operator, Δ is the forward difference operator, and ∇ is the backward difference operator. In Equation (4.7), $\hat{A}$, $\hat{B}$, $\hat{C}$, and $\hat{M}$ are Jacobian matrices, I is a unity matrix, and ε_e and ε_i are the coefficients of the explicit and implicit smoothing terms, respectively.

Beam-Warming's ADI operator can be written in diagonal form. For the ξ direction it is expressed as

$$I+h\delta_\xi\hat{A}+J^{-1}\varepsilon_i\delta_\xi^2 J=T_\xi[I+h\delta_\xi\hat{D}_A+J^{-1}\varepsilon_i\delta_\xi^2 J]T_\xi^{-1}, \tag{4.8}$$

where $\hat{A}=T_\xi\hat{D}T_\xi^{-1}$ is a similarity transformation, $\hat{D}$ is a diagonal matrix, and T_ξ contains the eigenvectors. If flux vector splitting is used, the Jacobian matrix $\hat{A}$ can be decomposed as

$$\hat{A}=\hat{A}^+ + \hat{A}^-, \tag{4.9}$$

and the central difference in the RHS of Equation (4.8) can be expressed as two one-sided differences

$$T_\xi[I + \nabla_\xi D_A^+ + \Delta_\xi D_A^-]T_\xi^{-1}, \tag{4.10}$$

where

$$D_A^\pm = \frac{h}{2}(\hat{D}_A \pm |\hat{D}_A|) \pm \underline{J}^{-1}\varepsilon_i J, \tag{4.11}$$

and $\underline{J}^{-1}$ is the Jacobian at the central point in Equation (4.8). Equation (4.10) can also be written as

$$T_\xi[L_A + M_A + N_A]T_\xi^{-1}, \tag{4.12}$$

where

$$\begin{aligned} L_A &= -\frac{8}{6}D^+_{Aj-1} + \frac{1}{6}D^+_{Aj-2} \\ M_A &= I + \frac{7}{6}(D^+_{Aj} - D^-_{Aj}) \\ N_A &= \frac{8}{6}D^-_{Aj+1} - \frac{1}{6}D^-_{Aj+2} \end{aligned} \tag{4.13}$$

for three-point upwinding. The LU factorization of the ADI operator is expressed as

$$I + h\delta_\xi \hat{A} + J^{-1}\varepsilon_i\delta_\xi^2 J = T_\xi(L_A + M_A)M_A^{-1}(M_A + N_A)T_\xi^{-1}. \tag{4.14}$$

The entire scheme, for all of the operators, is given by

$$\begin{aligned} &T_\xi(L_A + M_A)M_A^{-1}(M_A + N_A)(T_\xi^{-1}T_\eta)(L_B + M_B)M_B^{-1} \\ &(M_B + N_B)(T_\eta^{-1}T_\zeta)(L_C + M_C)M_C^{-1}(M_C + N_C)T_\zeta^{-1}\Delta\hat{Q}^n = \\ &-\Delta t(\delta_\xi \hat{E}^n + \delta_\eta \hat{F}^n + \delta_\zeta \hat{G}^n - Re^{-1}\delta_\zeta \hat{S}^n) - \varepsilon_e J^{-1}[(\nabla_\xi\Delta_\xi)^2 + (\nabla_\eta\Delta_\eta)^2 + (\nabla_\zeta\Delta)^2]J\hat{Q}^n, \end{aligned} \tag{4.15}$$

where the RHS is the same as the RHS of Equation (4.7).

Explicit Euler Flux Terms

RANS3D provides a number of options which may be used to solve the explicit Euler flux terms on the RHS. The standard option, is to use either a second or fourth order accurate central difference formula with added numerical dissipation. Other options consist of Total Variation Diminishing (TVD) upwind differencing schemes. The TVD flux limiters can be evaluated by Roe's method [Ref. 21] or Goorjian and Obayashi's streamwise upwind algorithm [Ref. 22]. The inviscid Euler calculations also require that an inviscid computational grid be used, which is a grid that has fewer grid lines concentrated in the boundary layer than the viscous computational grid.

CHAPTER 5

Results

All-Body Hypersonic Aircraft

Background

The all-body configuration is a product of the hypersonic programs of the 1960's and early 1970's. Subsequent wind tunnel tests [Ref. 23], generated a wealth of experimental data for subsonic, transonic, supersonic, and hypersonic Mach numbers. The complete all-body configuration is an elliptical delta wing with control surfaces, shown in Figure 5.1. Computational analyses are performed for the body alone configuration. Although a full-scale working vehicle is not likely to be constructed, the all-body is an ideal configuration to analyze because its shape and design are similar to preliminary NASP designs.

Wind Tunnel Model

The leading edge of the all-body is swept back at a 75° angle. The forebody is an elliptical cone and the aftbody has elliptical cross sections with a sharp trailing edge. The point of transition from the forebody to the aftbody occurs at 2/3 the body length. This also corresponds to the location of maximum thickness. The major to minor axis ratio of the forebody elliptical cross sections is 4.0. The canards, the vertical tail, and the horizontal control surfaces shown in Figure 5.1, are removable on the wind tunnel model. The body tips are also removable, and are taken off when the horizontal tails are in place.

Wind tunnel test were conducted, beginning-with the body only, and then proceeding through a build-up of components until the final configuration was obtained. The only wind tunnel results needed in this analysis are for the body alone.

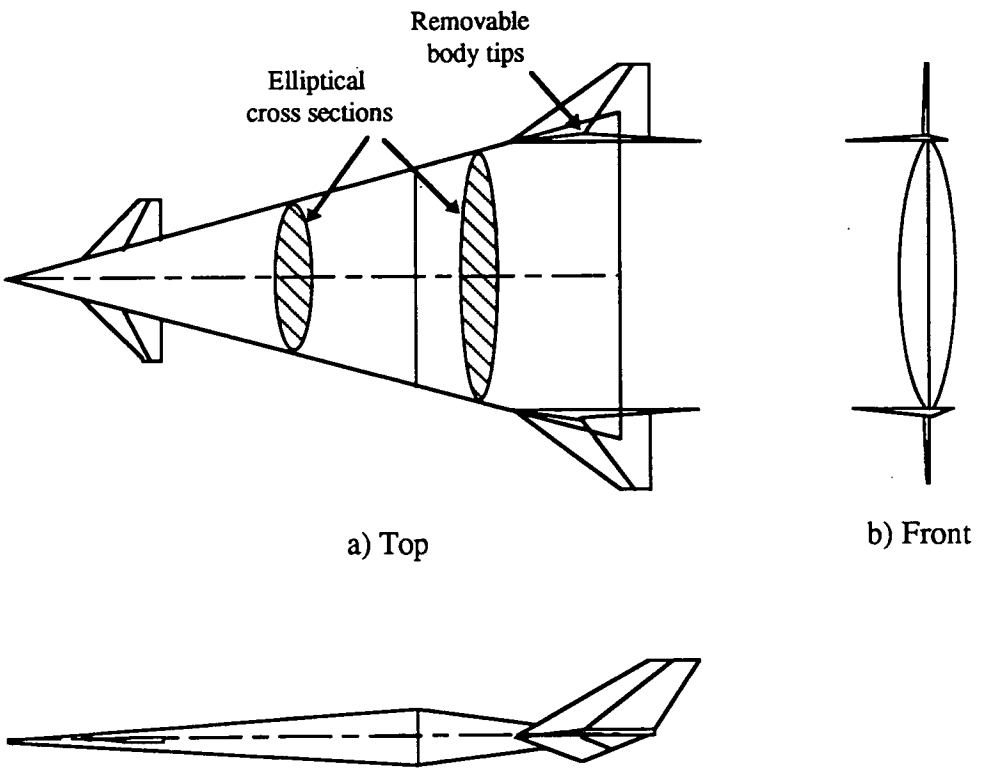


Figure 5.1. The All-Body Hypersonic Aircraft

Wind Tunnel Tests

Wind tunnel tests were conducted using two all-body models. Initially, a 19 inch model was used to obtain force and moment data followed later by a 3 foot model used to collect pressure data . The 19 inch model was tested in the Ames 6 ft. by 6 ft. Supersonic Wind Tunnel at a Reynolds number of 2.5×10^6. The 3 foot model was tested in the Ames 11 ft. Supersonic Wind Tunnel at a Reynolds number of 5.0×10^6. Both models were sting mounted on the aft portion of the upper surface to provide an undisturbed lower body

surface for hypersonic testing. In tests with the 19 inch model, angles of attack ranged from -2° to +15°. For the 3 foot pressure model, angles of attack ranged from -10° to +10°. The large negative angles of attack for the 3 foot model were necessary to obtain undisturbed pressure data on the upper surface. Since the sting is on the upper surface and because the model has xy as well as xz symmetry, upper surface pressure data for a positive α can be obtained from the lower surface at the negative value of the desired α.

The sting, mounted on the upper aft surface, tended to produce a higher pressure area on the upper aft surface. The result is a negative shift of C_{L_0} and a positive increase in $C_{m_{y_0}}$ at the lower speeds. These effects disappear at hypersonic speeds. Unfortunately hypersonics, and not transonics is the primary interest of the original study, so the shift in data is considered acceptable. Intuitively one knows a body with upper and lower surface symmetry should produce no lift at a zero angle of attack. This is not the case of the wind tunnel results. The same is true for the pitching moment, C_{m_y},which should be zero at a zero angle of attack, but is not.

Wave Drag Model

The wave drag method was the initial computational analysis applied to the all-body. The required geometric model is simple, because the shape of the all-body is very basic. Only three cross sections, with 37 points each, are used to accurately define the body geometry. The first cross section is the nose tip. The second is the maximum thickness location which divides the geometry into fore and aft sections, and the third cross section is the trailing edge. These three cross sections are sufficient for the wave drag analysis because the code linearly interpolates the body geometry between cross sections, and the actual body shape is linear in these regions. More complex configurations, however, would require additional cross sections.

The all-body is a good configuration to use with the wave drag code because it produces no lift at a zero angle of attack, which means the pressure drag is composed

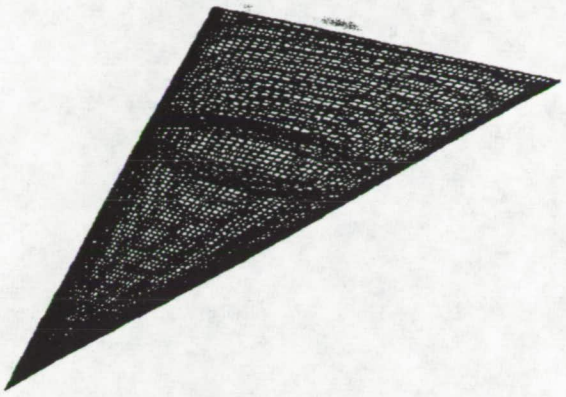
primarily of the wave drag since there is no induced drag. Also the all-body has a sharp nose and high sweep angle which makes it possible for the body to be entirely in the Mach cone. Results obtained when the body does extend beyond the Mach cone must be considered with skepticism because the theory does not hold for those cases.

Wave drag tests were performed for $\alpha = 0°$ at freestream Mach numbers of 1.1, 1.2, 1.3, 1.6, and 2.0. Runs were done on either a Silicon Graphics IRIS 4D/20 or 4D/340, with run times taking from 0.3 to 1.5 minutes, depending upon the Mach number, the number of cutting planes, and the number of longitudinal section cuts.

TRANAIR Model

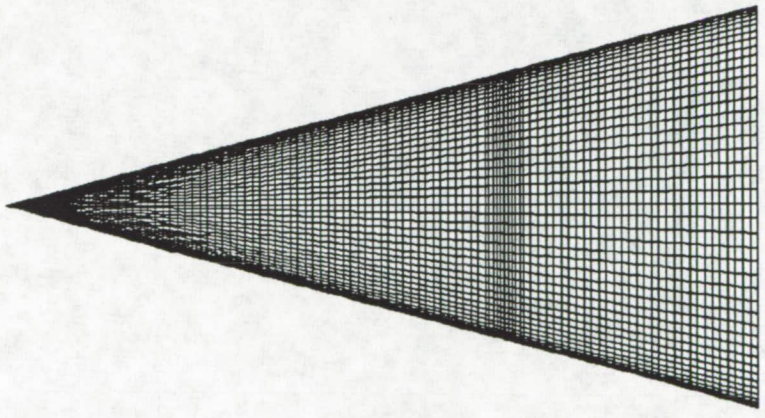
The all-body was the initial case analyzed with TRANAIR. The paneling scheme for the all-body is shown in Figure 5.2. The actual computational model used consists of only half the vehicle, since TRANAIR is capable of handling symmetry. The configuration is divided into four networks. Divisions are made between upper and lower surfaces and fore and aft sections of the body. The paneling is very dense over the entire surface. Panels are clustered near the nose, the leading edge, and the vehicle's maximum thickness point. The surface geometry contains 4,796 panels. This paneling scheme is finer than absolutely necessary, but was used partly out of inexperience and partly because the computational speed of TRANAIR is not directly related to the number of surface panels.

Runs were made for angles of attack of 0°, 2°, and 4° at freestream Mach numbers of 0.7, 0.9, 1.1, and 1.3. Run times on the CRAY Y-MP8/832 varied from 0.6 to 1.1 CPU hours depending upon the angle of attack and freestream Mach number. The minimum target residual for each case was 10^{-7}. In all runs the grid sequencing option was used. The results are generally good, but there are problem areas apparently occurring because the grid is not fine enough. The more accurate but time consuming adaptive gridding method is used for one case as a comparison to the grid sequencing method.



a) Isometric View

b) Side View



c) Top View

d) Front View

Figure 5.2. All-Body Paneling Scheme

RANS3D Model

The all-body computational grid used in the RANS3D analysis was supplied, along with the code, by Gary Cosentino. It is a viscous grid which implies a dense clustering of grid lines in the boundary layer, near the body surface. The grid has the dimensions 80x51x30 in the ξ-η-ζ transformed coordinate system. The outer boundaries of the grid resemble a cylinder, Figure 5.3, and the inner boundaries conform to the surface of the configuration, Figure 5.4 and Figure 5.5. In Figure 5.3 the small grid section in the center, shown in red, corresponds to the surface of the all-body defined between $\xi_{LE} = 15$ and $\xi_{TE} = 63$. The yellow grid lines represent the inner boundary wall, the blue represents the outer boundary, and the green grids are inlet and outlet boundaries.

Figure 5.4 and Figure 5.5 are close up views showing the grid fitted around the all-body surface. Figure 5.4 shows a front view of an η–ζ grid plane taken at the maximum thickness location. The dense green area near the surface shows the high clustering of grid points in the boundary layer. If the Euler mode of RANS3D was chosen then this grid would need to be coarser in this region. The grid is shown missing the first ($\eta = 1$) and last ($\eta = 51$) grid lines. These grid lines are used to transfer information between grids for cases involving sideslip, which require two grids to be used. Figure 5.5 shows the side view of the computational grid along the centerline ($\eta = 2$ and $\eta = 50$) of the vehicle. This view also shows the heavy clustering of grid lines in the boundary layer region and also near the nose, the body transition point, and the trailing edge.

Runs were made at freestream Mach numbers of 0.7, 0.9, 1.1, 1.3, and 1.6 with angles of attack of 0°, 2°, and 4°. At Mach 0.9, 1.1, and 1.3, additional angles of 6° and 10° were also examined. Run times on the CRAY Y-MP8/832 varied from 0.7 to 1.7 CPU hours depending upon angle of attack and freestream Mach number. Convergence was considered achieved when a residual of 10^{-5} for C_L was obtained. This typically occurred between 1,500 and 3,000 iterations depending upon angle of attack and freestream Mach

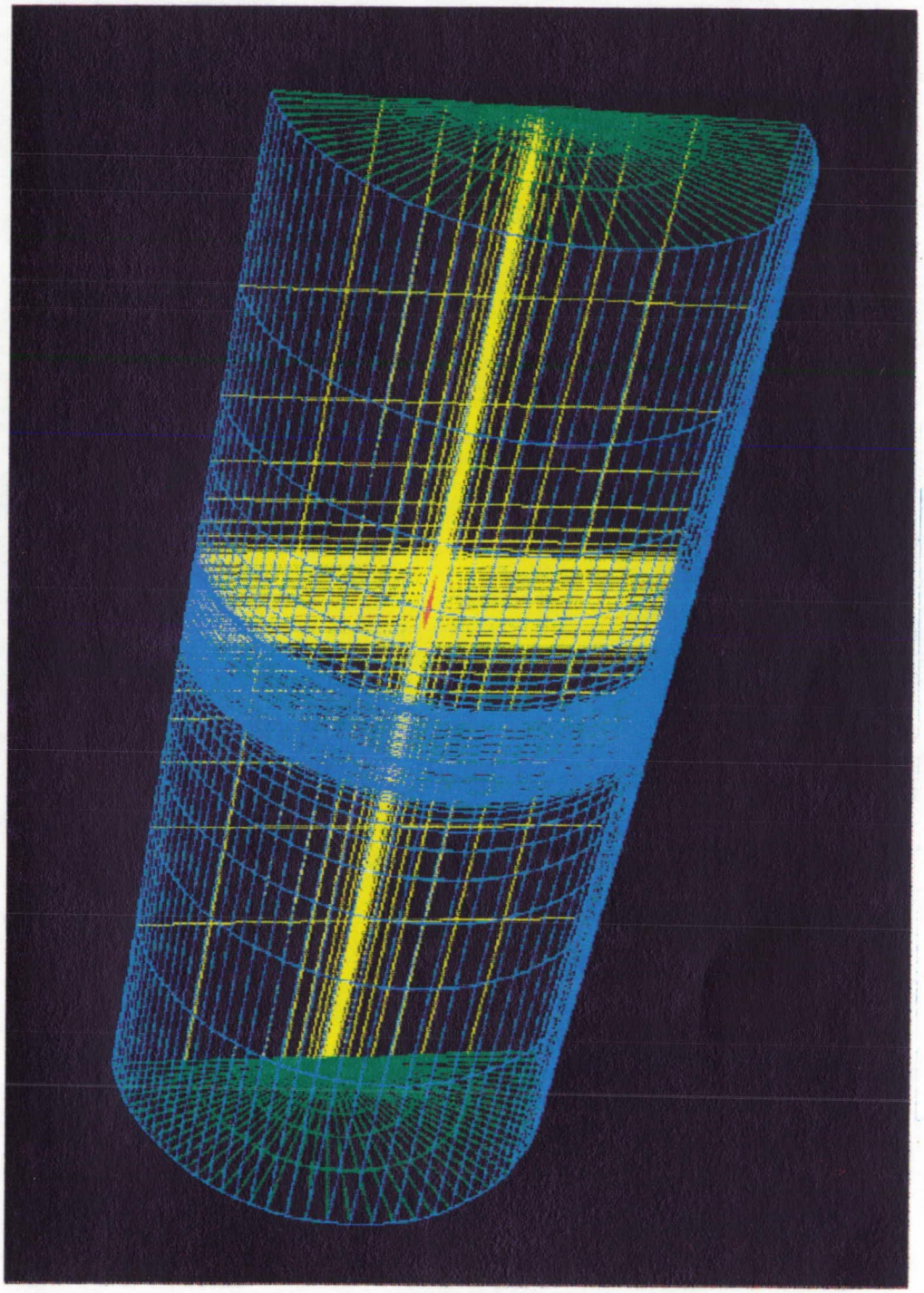


Figure 5.3. All-Body 80x51x30 Viscous Computational Grid

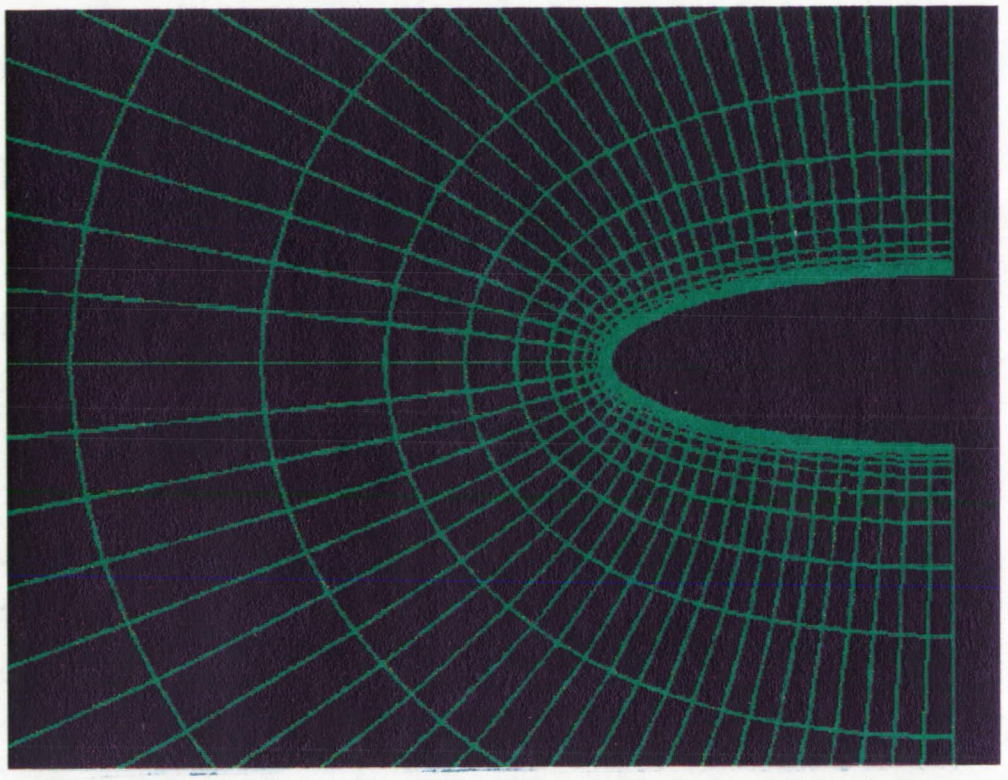


Figure 5.4. Front View of All-Body Viscous Grid, $\xi = 45$

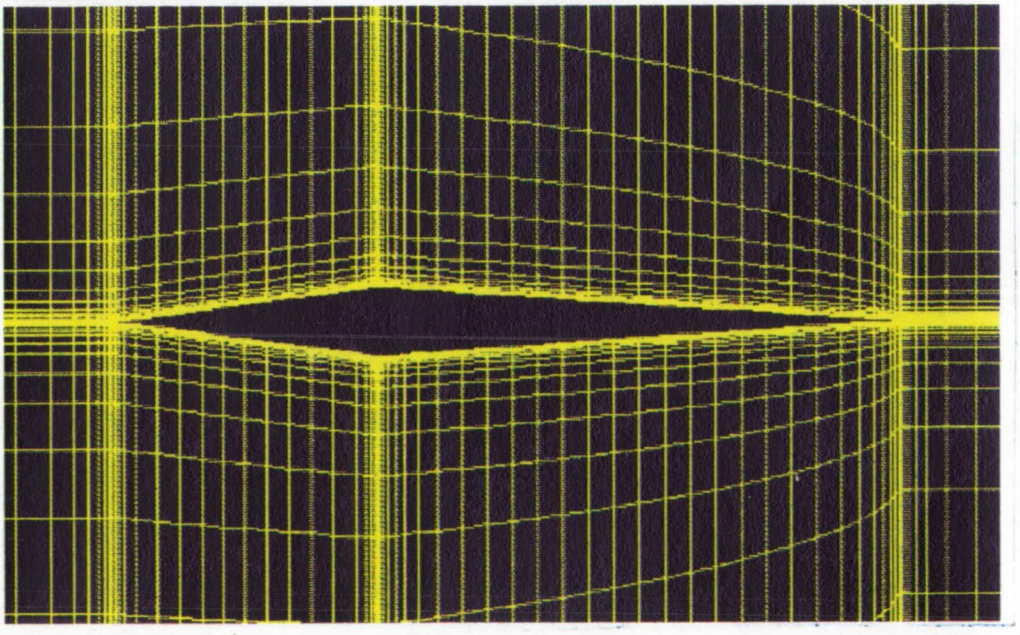


Figure 5.5. Side View of All-Body Viscous Grid, $\eta = 2$ and $\eta = 50$

number. Figure 5.6 shows the convergence history of the all-body lift coefficient for the $M_\infty = 0.7$ and $\alpha = 2°$ run. For this case, with a subsonic freestream Mach number and low angle of attack, the solution converges fairly rapidly. As the Mach number approaches supersonic values, the iterations required for convergence go up. Also as the angle of attack increases so do the number of iterations.

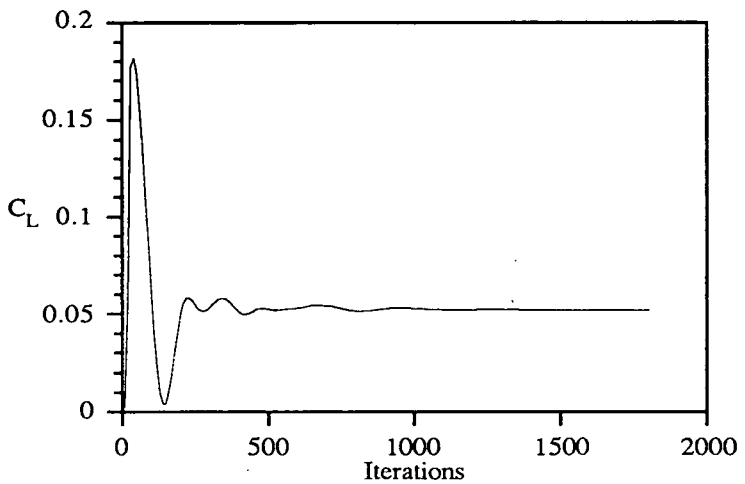


Figure 5.6. RANS3D C_L Convergence History, All-Body, $M_\infty = 0.7$, $\alpha = 2°$

Comparison of Pressure Distributions

Both TRANAIR and RANS3D are capable of providing pressure data on the configuration surface. Comparisons between TRANAIR, RANS3D, and the wind tunnel pressure coefficients (C_P's) are done at spanwise angles of 0°, 62°, 82°, 90°, 98°, 118°, and 180° for Mach numbers of 0.7, 0.9, 1.1, and 1.3, at angles of attack of 0° and 4°. The spanwise angle, ϕ, is taken with respect to the positive z-axis in the yz-plane as illustrated in Figure 5.7. The value of ϕ is 0° on the centerline of the upper surface, 90° on the leading edge, and 180° on the centerline of the lower surface.

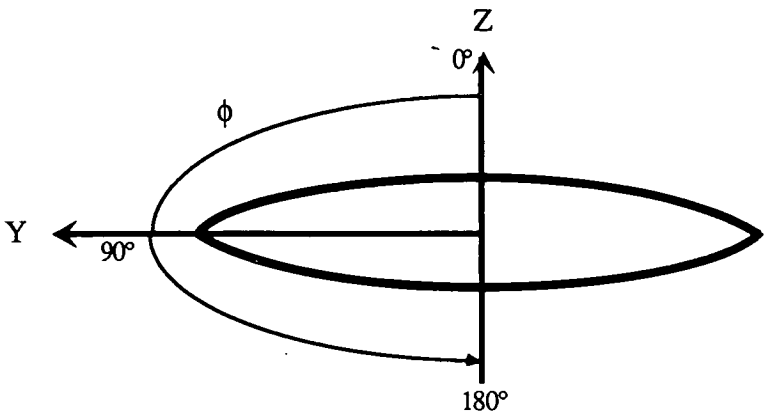


Figure 5.7. Definition of Spanwise Angle, ϕ, for an All-Body Cross Section

Freestream Mach number of 0.7. At the subsonic freestream Mach number of 0.7 and angle of attack of 0°, both TRANAIR and RANS3D results compare favorably with the wind tunnel data, as shown in Figure 5.8. For spanwise angles of 0° and 62°, TRANAIR predicts a larger change in pressure at the maximum thickness location, than RANS3D. It is probable that the RANS3D results at this point are closer to the physical case, but it is difficult to absolutely determine, since there is no experimental data exactly at this critical location. On either side of the break point, both codes are able to closely match the wind tunnel data. At the 82° location both TRANAIR and RANS3D agree well with the experimental data. They also both predict the same pressure loss at this location, although it is smaller than for the 0° and 62° locations.

Along the leading edge, at $\phi = 90°$, RANS3D fails to completely match the wind tunnel data. The code gives results which show a very high increase in pressure near the trailing edge. At the trailing edge, however, the pressure does drop back down. This behavior is only exhibited on the leading edge. The last grid point on the leading edge which is also on the trailing edge wake, is a singularity point. The most probable reason for the pressure increase is due to the method that the code employs to handle the singularity point. At the wake-sheet tip on the trailing edge of the body, the boundary conditions are obtained by extrapolating values in the η direction. RANS3D's default

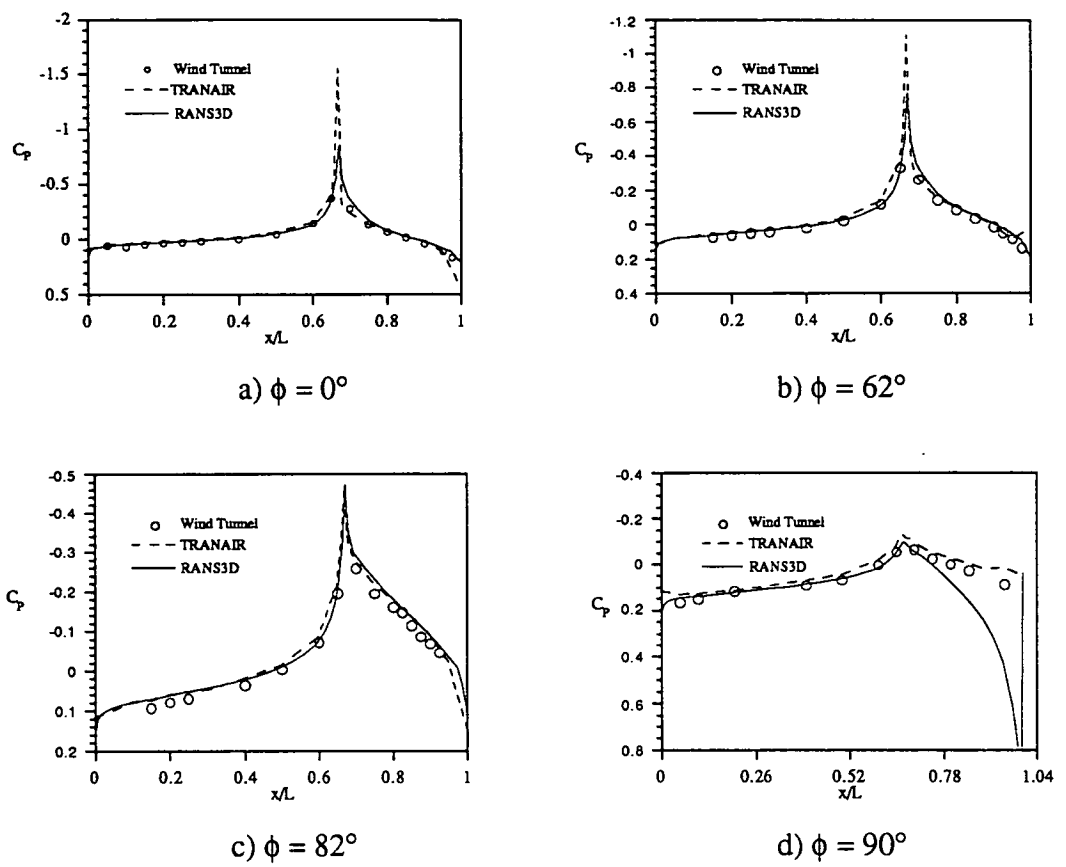
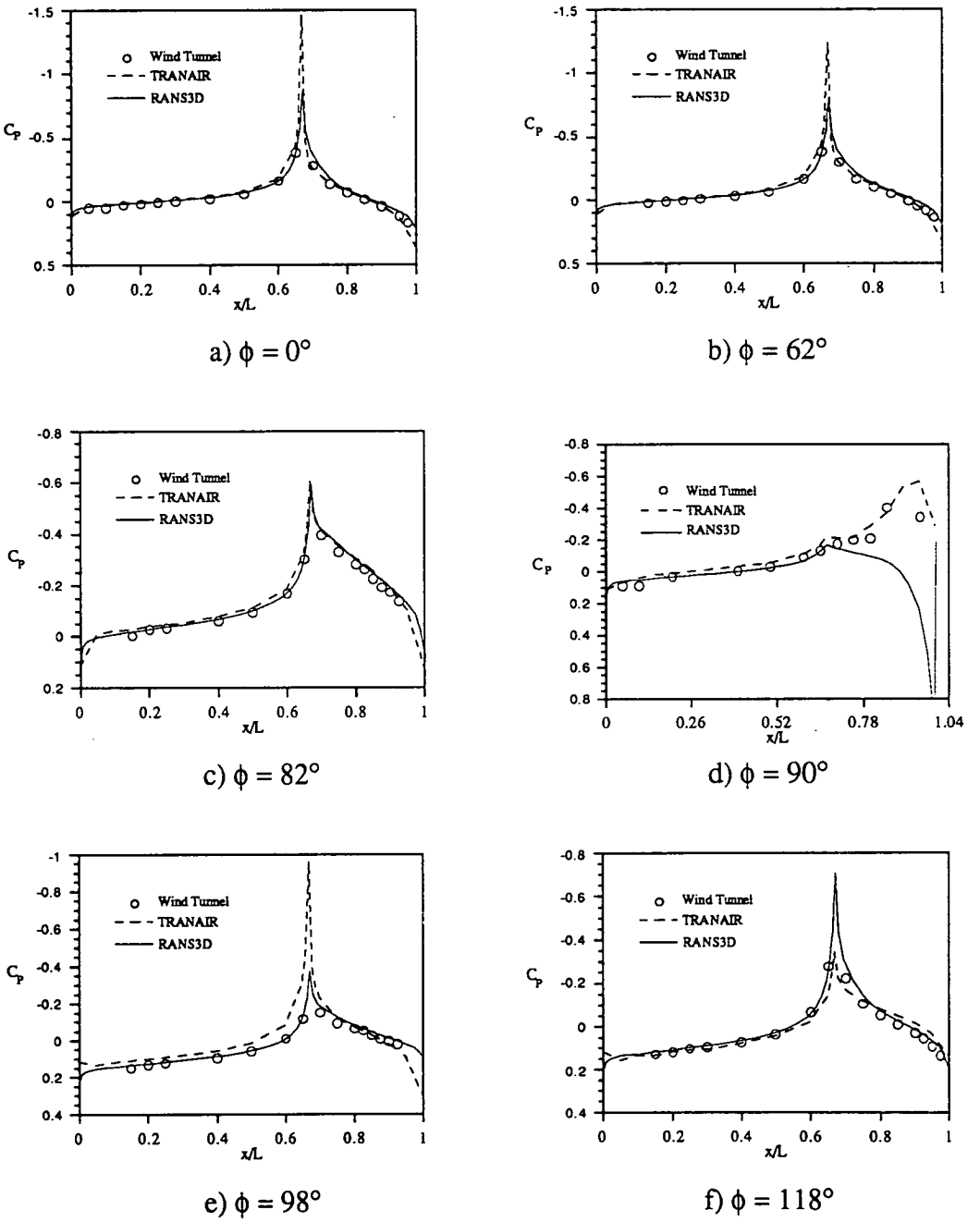


Figure 5.8. All-Body C_P Distribution, $M_\infty = 0.7$, $\alpha = 0°$

option is a third order technique. This, however, is too high for this particular case and results in an overshoot of the pressure. It should be possible to modify the code to handle the tip boundary condition so that this result subsides [Ref. 24]. The high pressure region should have little effect on the overall lift since it acts perpendicular to the leading edge. However, it could result in slightly higher predicted pressure drags, which do occur and are illustrated in the "Comparison of Forces and Moments" section of this chapter.

For $\alpha = 4°$, the results are similar to those for $\alpha = 0°$ as shown in Figure 5.9. Since the flow on the upper surface is no longer symmetric to the flow on the lower surface, additional spanwise angles of 98°, 118°, and 180° are used to display the pressure distribution for the lower surface. The upper surface trends for both codes remain the same as for the previous case. On the lower surface, the RANS3D results follow the same trend

as on the upper surface, but the TRANAIR results differ. At $\phi = 98°$, TRANAIR slightly under predicts the pressure along the forebody and then greatly over predicts the pressure loss at the maximum thickness location. At a spanwise angle of 118°, TRANAIR under

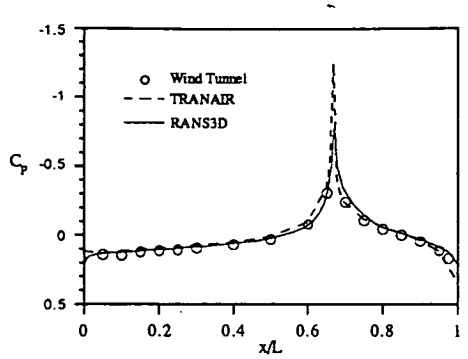


a) $\phi = 0°$ b) $\phi = 62°$

c) $\phi = 82°$ d) $\phi = 90°$

e) $\phi = 98°$ f) $\phi = 118°$

Figure 5.9 (part i). All-Body C_P Distribution, $M_\infty = 0.7$, $\alpha = 4°$



g) $\phi = 180°$

Figure 5.9 (part ii). All-Body C_P Distribution, $M_\infty = 0.7$, $\alpha = 4°$.

predicts the pressure drop at this same location, but is closer to the experimental data than for the 98° location. The 180° results are very similar to the upper surface results at 0°.

In all cases, both TRANAIR and RANS3D are able to predict the loss in pressure resulting from the expansion flow. RANS3D seems better at predicting the magnitude of the pressure loss, but does exhibit a boundary condition problem which causes unrealistic high pressure on the leading edge. For this Mach number, the full potential code, TRANAIR, is able to compete with the Navier-Stokes code, RANS3D. TRANAIR's success occurs partly because the freestream Mach number is sufficiently below 1.0, the vehicle shape is simple, and the angles of attack are low, thus preventing areas of major separation.

Freestream Mach number of 0.9. At $M_\infty = 0.9$ the flow is transonic with both subsonic and supersonic regions. At $\alpha = 0°$, Figure 5.10, many of the same characteristics exhibited at $M_\infty = 0.7$ are also present. For $\phi = 0°$ and $\phi = 62°$, however, RANS3D under predicts the magnitude of the expansion flow pressure loss, while TRANAIR handles it slightly better. At $\phi = 82°$, the TRANAIR pressure curve is slightly jagged. This is probably the result of an insufficient amount of grid cells in the area, caused by using the grid sequencing method. At this spanwise location both codes show an increase in the low

pressure region occurring at the maximum thickness location. These predictions are validated by the experimental results.

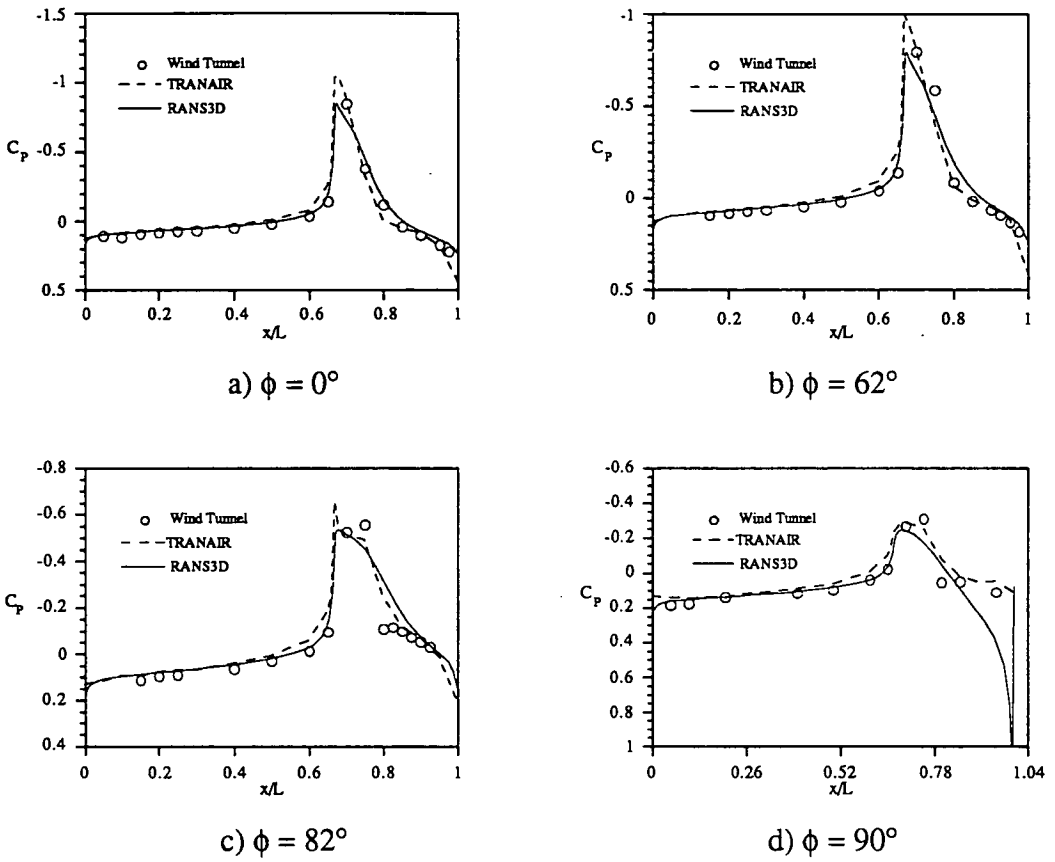


Figure 5.10. All-Body C_P Distribution, $M_\infty = 0.9$, $\alpha = 0°$

At $\alpha = 4°$, Figure 5.11, the trends of the RANS3D and the TRANAIR pressure distributions are about the same as they were for $\alpha = 0°$. Again at $\phi = 82°$ in the expansion region the pressure curve is slightly smeared, but both codes do fairly good at matching the experimental data. The leading edge case of 90° is similar to before with the erratic high pressure of the RANS3D code being the most noticeable deviation. Although TRANAIR does better in the area where RANS3D has a problem, it still misses some of the wind tunnel points. It is very likely that in this region there is a fair amount of circulation because of the leading edge vortices which TRANAIR is unable to model. On the lower surface at the 98° and 118° spanwise angles, the TRANAIR predicted pressure loss is much

smaller in magnitude than the wind tunnel loss. The RANS3D predicted pressure loss for both of these cases is good. One small shortcoming which both codes exhibit at these two spanwise locations, is a slower pressure recovery after the pressure drop compared to the

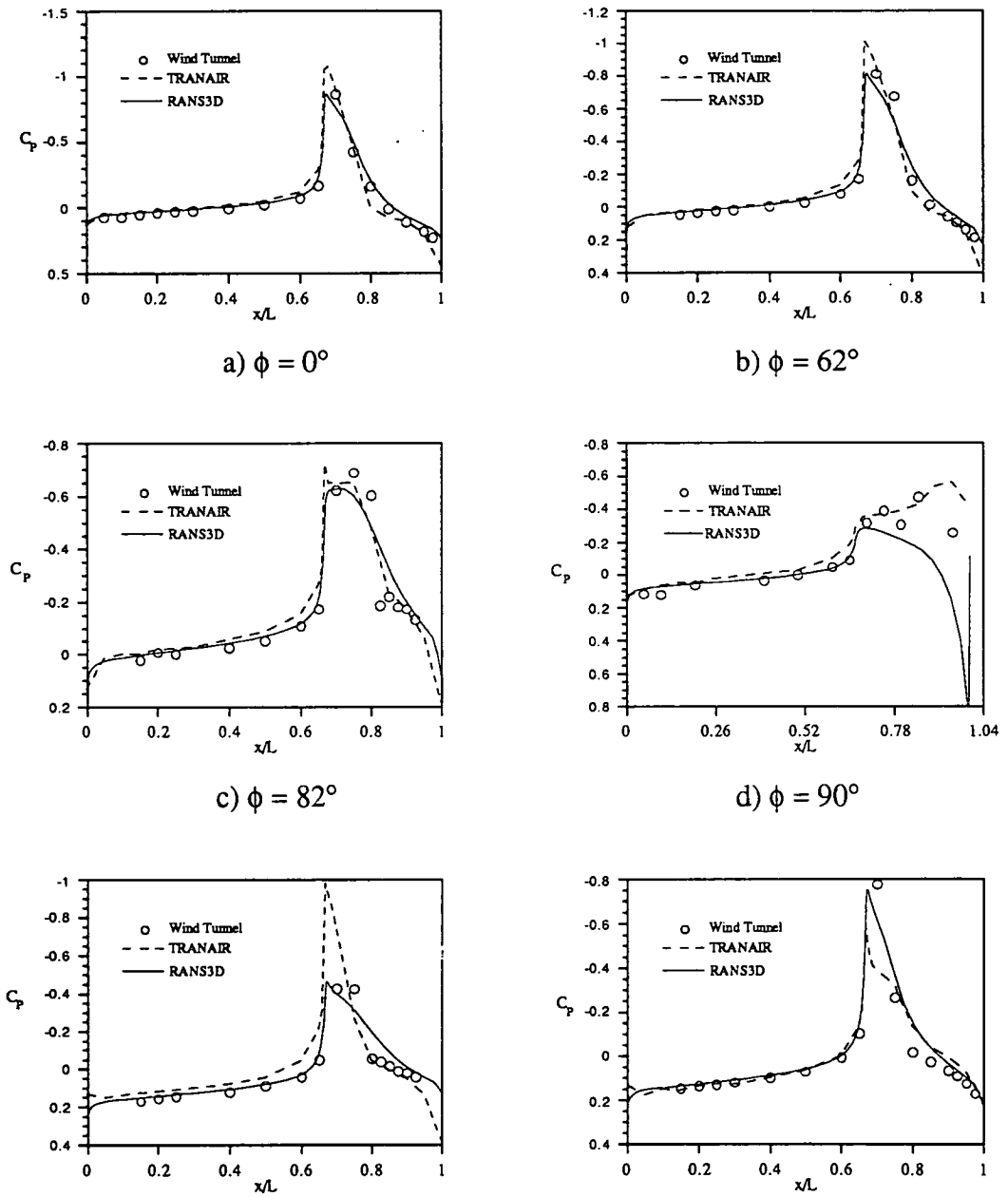
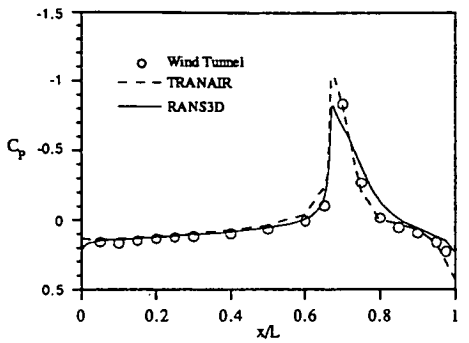


Figure 5.11 (part i). All-Body C_P Distribution, $M_\infty = 0.9$, $\alpha = 4°$



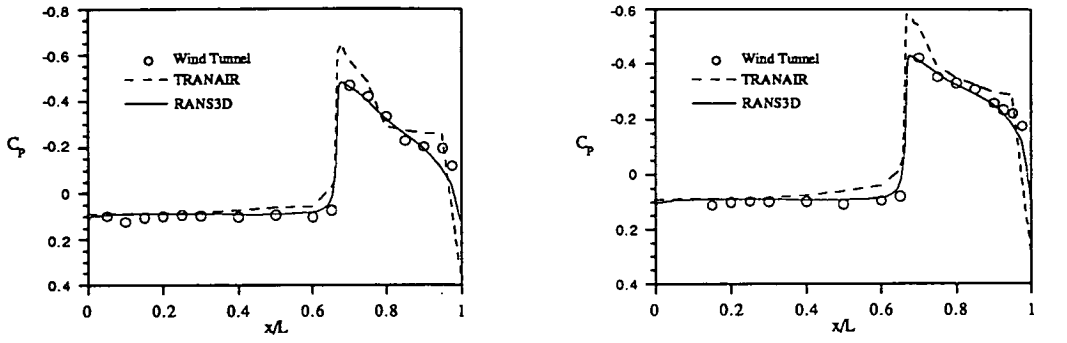
g) $\phi = 180°$

Figure 5.11 (part ii). All-Body C_P Distribution, $M_\infty = 0.9$, $\alpha = 4°$

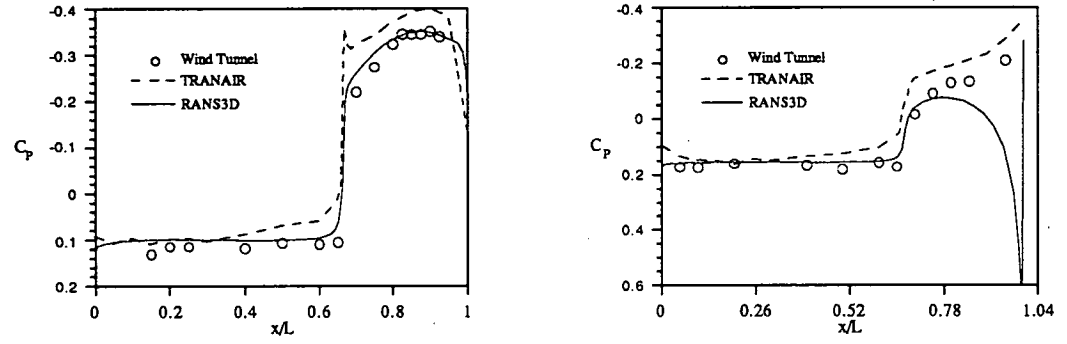
wind tunnel results. The 180° case, in Figure 5.11, shows just a small discrepancy in the pressure loss magnitude between the two codes. Otherwise, the results are good.

At Mach 0.9, the numerical analysis becomes more difficult because the freestream Mach number is approaching unity. In general, the results are poorer than the Mach 0.7 cases, but they are still good. The major area of discrepancy between the codes is again in the magnitude prediction of the expansion region with RANS3D generally doing better than TRANAIR. Also RANS3D tends to slightly under predict the pressure along the aftbody.

Freestream Mach number of 1.1. The next case is for a supersonic freestream Mach number of 1.1. At an angle of attack of zero, Figure 5.12, the RANS3D results, except for the leading edge, are very good. Overall, the pressure distributions predicted by RANS3D successfully match the experimental data. The TRANAIR results, however, experience some deviations. TRANAIR shows spikes in the 0°, 62°, and 82° spanwise location pressure curves. These spikes most likely occur from using the grid sequencing option. In retrospect, the adaptive gridding option should have been chosen, especially since the freestream is supersonic. The 90° location also shows poor TRANAIR results, compared to the subsonic freestream cases. RANS3D also can still not adequately handle the trailing edge boundary condition at $\phi = 90°$.



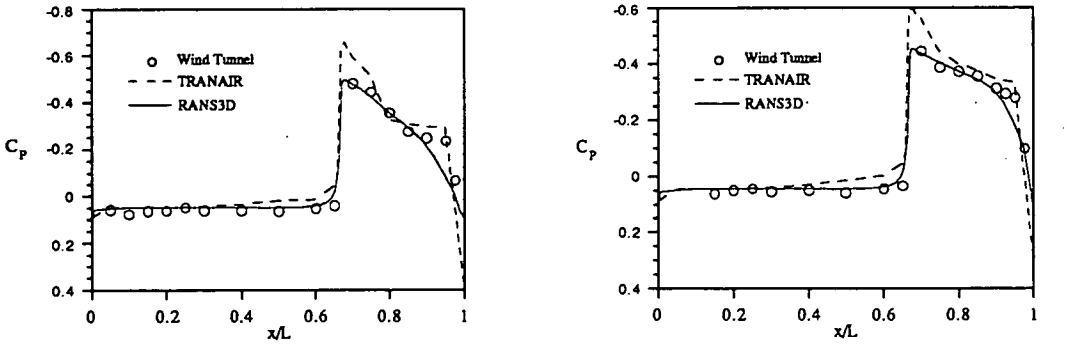
a) $\phi = 0°$ b) $\phi = 62°$



c) $\phi = 82°$ d) $\phi = 90°$

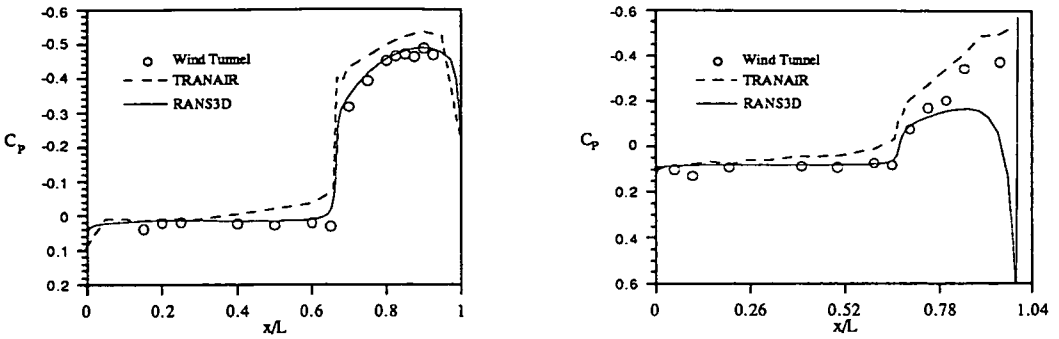
Figure 5.12. All-Body C_P Distribution, $M_\infty = 1.1$, $\alpha = 0°$

For an angle of attack of 4°, Figure 5.13, the RANS3D results are still very good. There are a few places where the pressure distribution deviates slightly from the



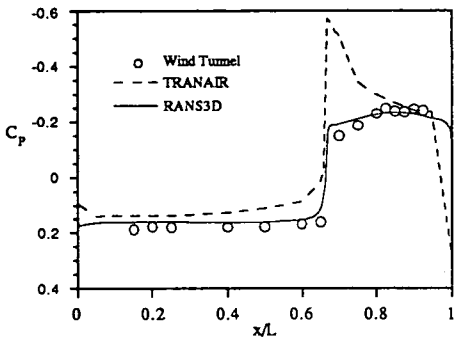
a) $\phi = 0°$ b) $\phi = 62°$

Figure 5.13 (part i). All-Body C_P Distribution, $M_\infty = 1.1$, $\alpha = 4°$

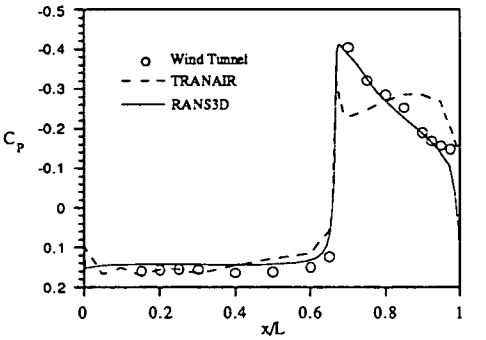


c) $\phi = 82°$

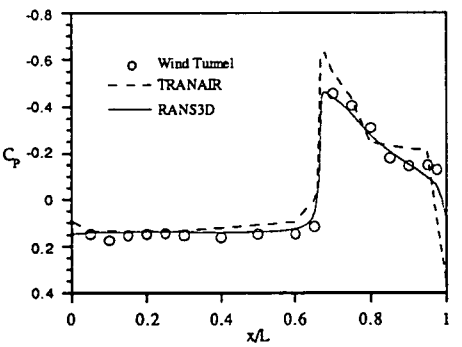
d) $\phi = 90°$



e) $\phi = 98°$



f) $\phi = 118°$



g) $\phi = 180°$

Figure 5.13 (part ii). All-Body C_P Distribution, $M_\infty = 1.1$, $\alpha = 4°$

experimental values, but for transonic flow, the results are impressive. The TRANAIR results still suffer once the expansion location is reached for every spanwise location. On the lower surface at $\phi = 98°$ and $\phi = 118°$, the pressure curve predicted by TRANAIR is

especially poor. At the 98° spanwise location the pressure drop is significantly over predicted. The pressure drop at 118° is under predicted and then, after some pressure recovery, an additional loss in pressure occurs which does not correspond to the wind tunnel results.

At $M_\infty = 1.1$, there is mixed subsonic and supersonic flow. The TRANAIR pressure distribution curves are somewhat jagged and falter in several locations, especially in the expansion region. The RANS3D results are in general very good. A Navier-Stokes code is better suited to handle supersonic freestream Mach numbers than a full potential code. Nevertheless, TRANAIR's limitations are not entirely responsible for the deviations in the supersonic results. As stated previously, the use of adaptive gridding over grid sequencing would be a better choice for supersonic freestream values.

Freestream Mach number of 1.3. At $M_\infty = 1.3$ and $\alpha = 0°$, Figure 5.14, the results

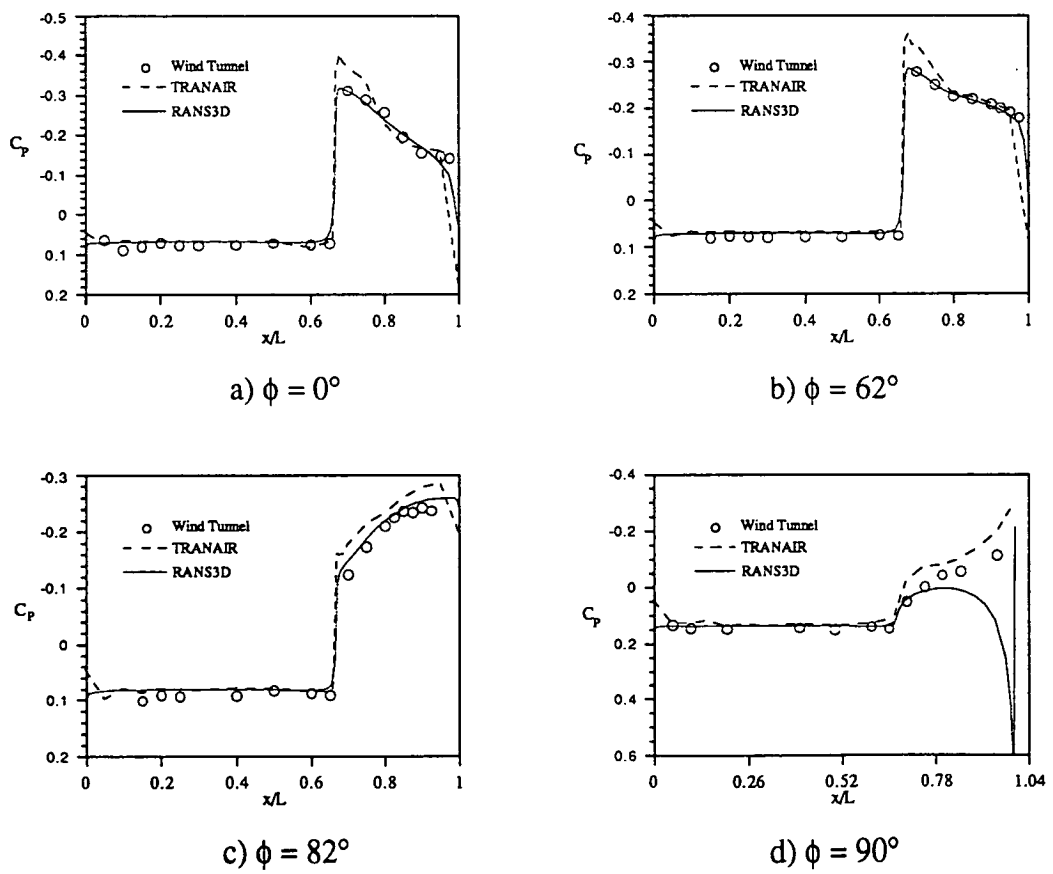


Figure 5.14. All-Body C_P Distribution, $M_\infty = 1.3$, $\alpha = 0°$

are very similar to the those at $M_\infty = 1.1$ and $\alpha = 0°$. Both numerically predicted pressure distributions do well along the forebody, with only the TRANAIR distributions experiencing some moderate deviations in locations on the aftbody.

At the 4° angle of attack, Figure 5.15, the results are again similar to those for Mach 1.1. At spanwise angles of 0°, 62°, and 180°, the predicted magnitudes of pressure loss are slightly greater than the experimental magnitudes. At $\phi = 98°$ and 118°, both the magnitude and shape of the expansion flow are poorly predicted by TRANAIR. Also at these lower surface locations, TRANAIR has problems predicting the pressure distribution along the forebody. The RANS3D results are again very good.

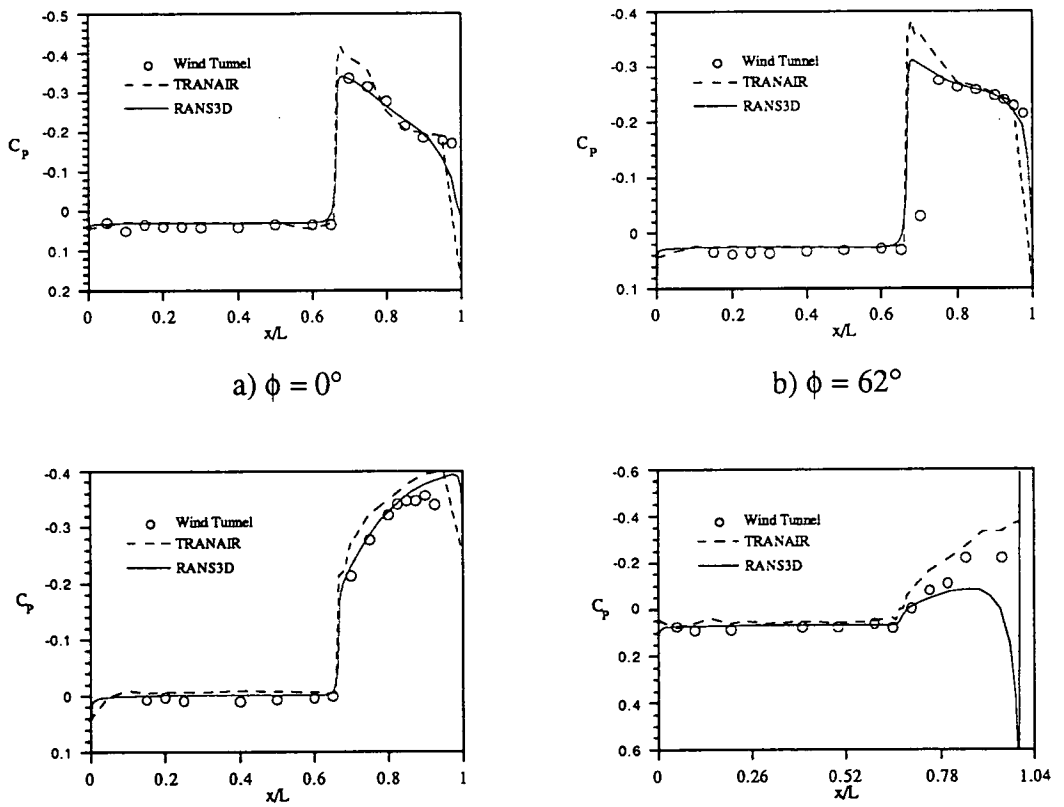
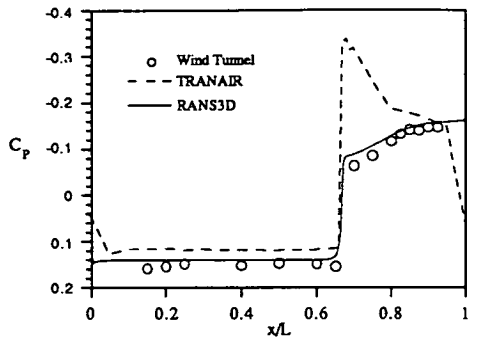
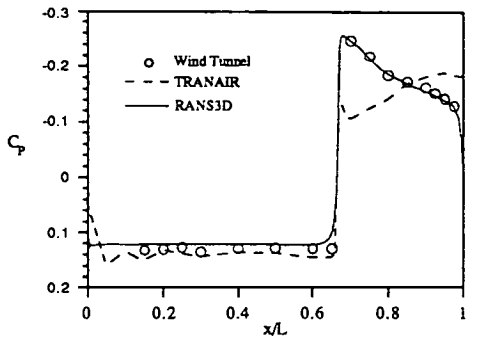


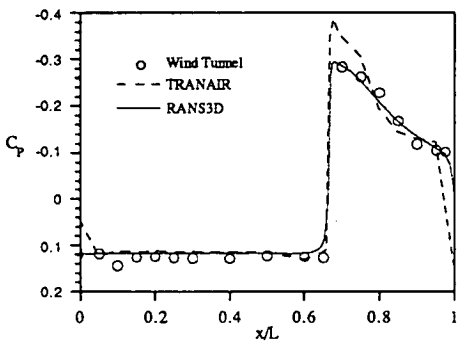
Figure 5.15 (part i). All-Body C_P Distribution, $M_\infty = 1.3$, $\alpha = 4°$



e) $\phi = 98°$



f) $\phi = 118°$



g) $\phi = 180°$

Figure 5.15 (part ii). All-Body C_P Distribution, $M_\infty = 1.3$, $\alpha = 4°$

The results for $M_\infty = 1.3$ are basically the same as for $M_\infty = 1.1$, and for the same reasons. The use of TRANAIR at higher Mach numbers is possible but near $M_\infty = 2.0$, the code's accuracy diminishes severely.

Comparison of TRANAIR Gridding Methods

To determine what effect the grid sequencing option has on the computed solution, a comparison is made with the solution adaptive grid technique. The test is conducted on the all-body configuration for $M_\infty = 0.7$ and $\alpha = 2°$. The previously examined TRANAIR grid sequencing pressure distributions for $M_\infty = 0.7$ at $\alpha = 0°$ and $\alpha = 4°$, were relatively good except for a few minor deviations. The results of the comparison are given in Figure

5.16. On the upper surface there is no difference between the two methods, and both methods agree very well with the wind tunnel results.

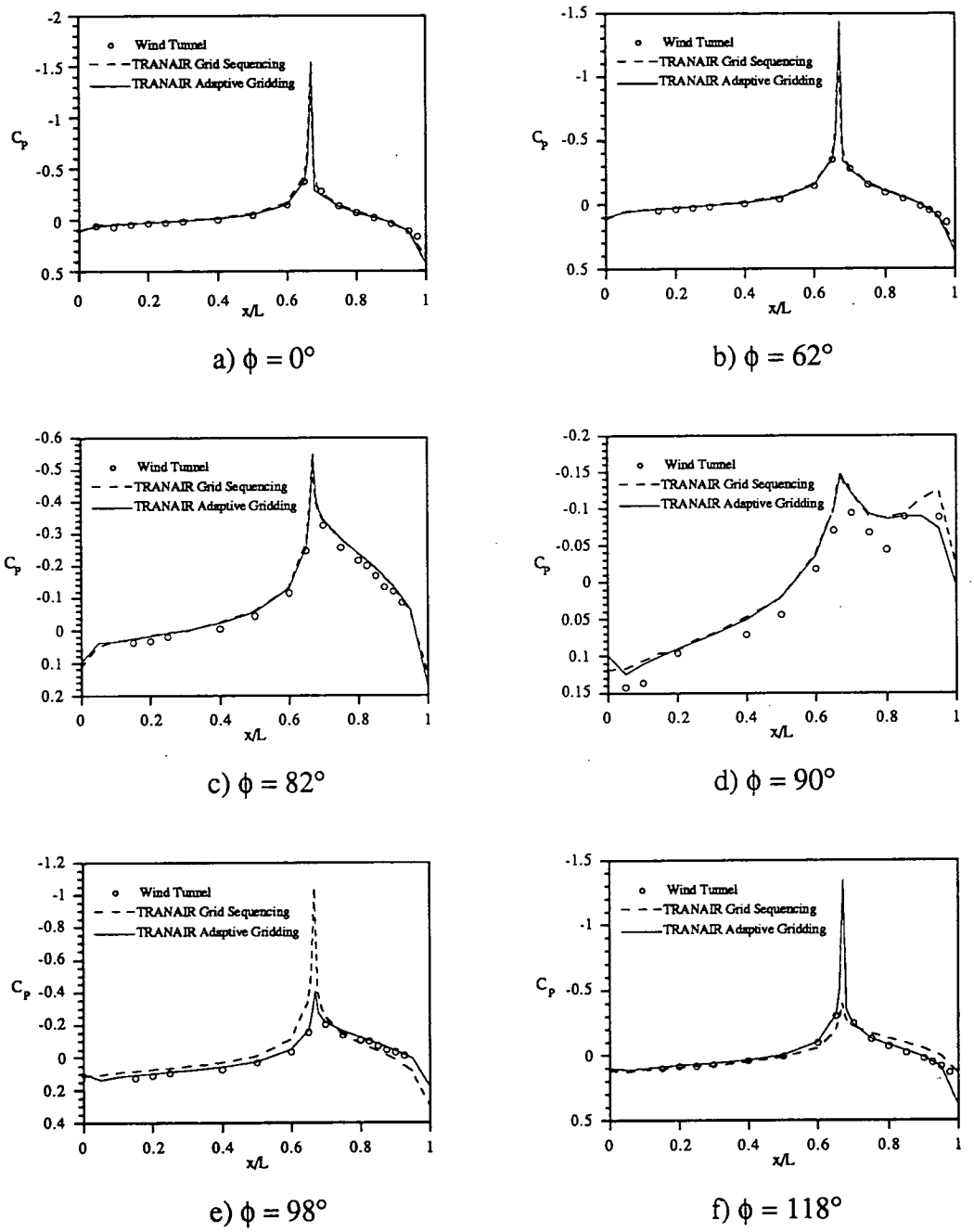
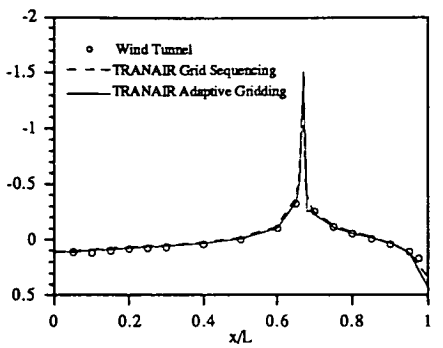


Figure 5.16 (part i). Comparison of Grid Sequencing and Adaptive Gridding Methods, All-Body C_P Distribution, $M_\infty = 0.7$, $\alpha = 2°$



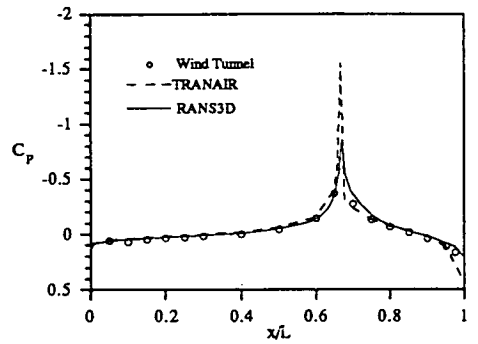
g) $\phi = 180°$

Figure 5.16 (part ii). Comparison of Grid Sequencing and Adaptive Gridding Methods, All-Body C_P Distribution, $M_\infty = 0.7$, $\alpha = 2°$

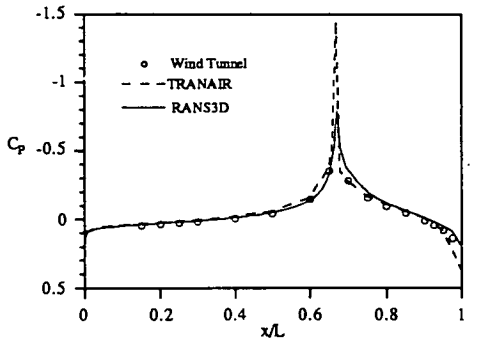
On the leading edge, $\phi = 90°$, there is the first sign of a difference between the two methods. The adaptive gridding method seems to do better near the trailing edge, although the overall results for the leading edge are still only fair. This is only a full potential code and along the leading edge there is probably a vortex which TRANAIR is incapable of predicting. On the lower surface is where the largest improvement in C_P prediction occurs. At the $\phi = 98°$ location, the adaptive gridding results completely match the wind tunnel data, while the grid sequencing method under predicts the forebody pressure, over predicts the pressure loss, and slightly over predicts the aftbody pressure. At the $\phi = 118°$ location the adaptive gridding method shows an improvement in matching the pressure distribution along the aftbody. However, the magnitude of the pressure drop seems to be over predicted although all of the wind tunnel data points are hit by the adaptive gridding curve. As with the upper surface, both methods agree with each other and the experimental data at the 180° location.

The next logical step is to compare the adaptive gridding results, the RANS3D results, and the wind tunnel data. These three pressure distributions, shown in Figure 5.17, match very well. The main discrepancy seems to be the magnitude of the pressure drop. Significant differences occur at $\phi = 0°$, 62°, 118°, and 180°. In all four locations,

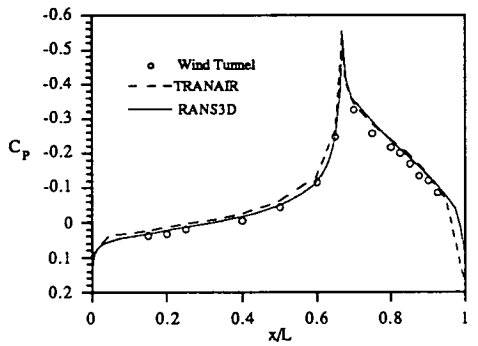
TRANAIR predicts a larger drop in pressure than RANS3D. Although it is difficult to tell which method predicts the pressure loss better, TRANAIR does match the experimental



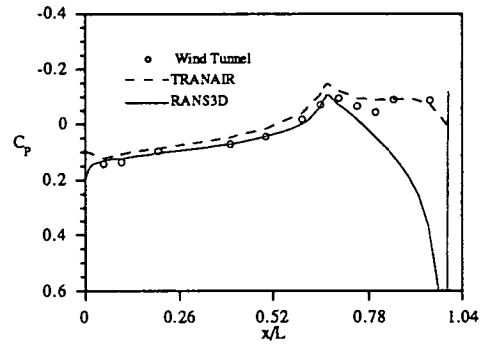
a) $\phi = 0°$



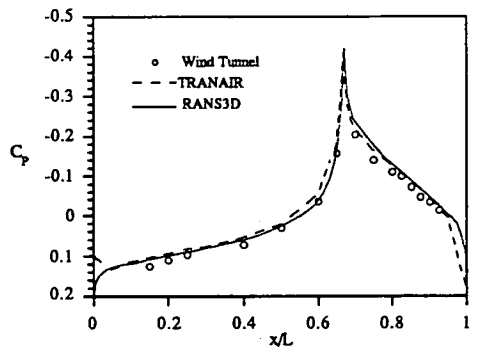
b) $\phi = 62°$



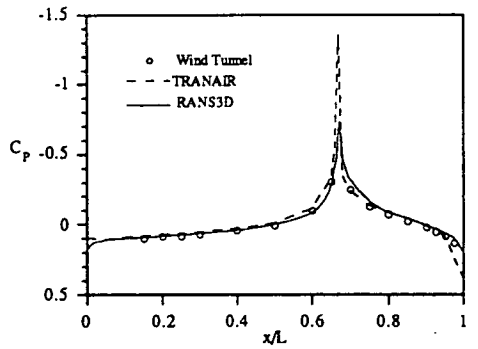
c) $\phi = 82°$



c) $\phi = 90°$

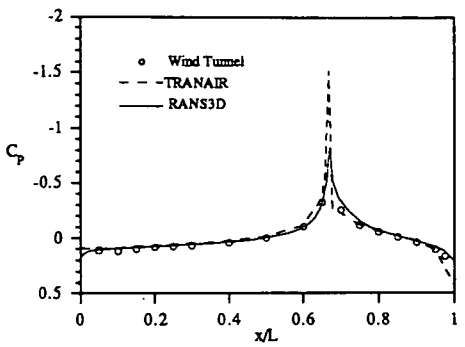


d) $\phi = 98°$



e) $\phi = 118°$

Figure 5.17 (part i). Comparison of TRANAIR Adaptive Gridding with RANS3D, All-Body C_P Distribution, $M_\infty = 0.7$, $\alpha = 2°$



g) $\phi = 180°$

Figure 5.17 (part ii). Comparison of TRANAIR Adaptive Gridding with RANS3D, All-Body C_P Distribution, $M_\infty = 0.7$, $\alpha = 2°$

pressure immediately after the expansion region better, while RANS3D predicts the pressure immediately before the expansion location slightly better than TRANAIR.

It is evident that the adaptive gridding method improves the TRANAIR results for this case. It would probably also do so for other cases, especially those with supersonic freestreams. To actually determine how much it would improve, it is necessary to perform the analysis. Unfortunately this could not be done, because at the time of this writing, TRANAIR had been inoperable at NASA Ames for 4 months because of a change in the CRAY operating system.

Another comparison of RANS3D and TRANAIR pressure distributions is shown in Figure 5.18. The color legend correlates the highest pressure to magenta and the lowest pressure to blue with a spectrum of colors and corresponding pressures in between. The RANS3D plot was obtained using PLOT3D [Ref. 25] and the TRANAIR plot was obtained using NASA Ames RA Division Interactive Displayer (RAID) [Ref. 26]. Both plots were done using the same color scale to insure compatibility between the colors. The quality of the TRANAIR plot appears to be superior only because there are a larger number of panels defining the surface, so the resolution is better.

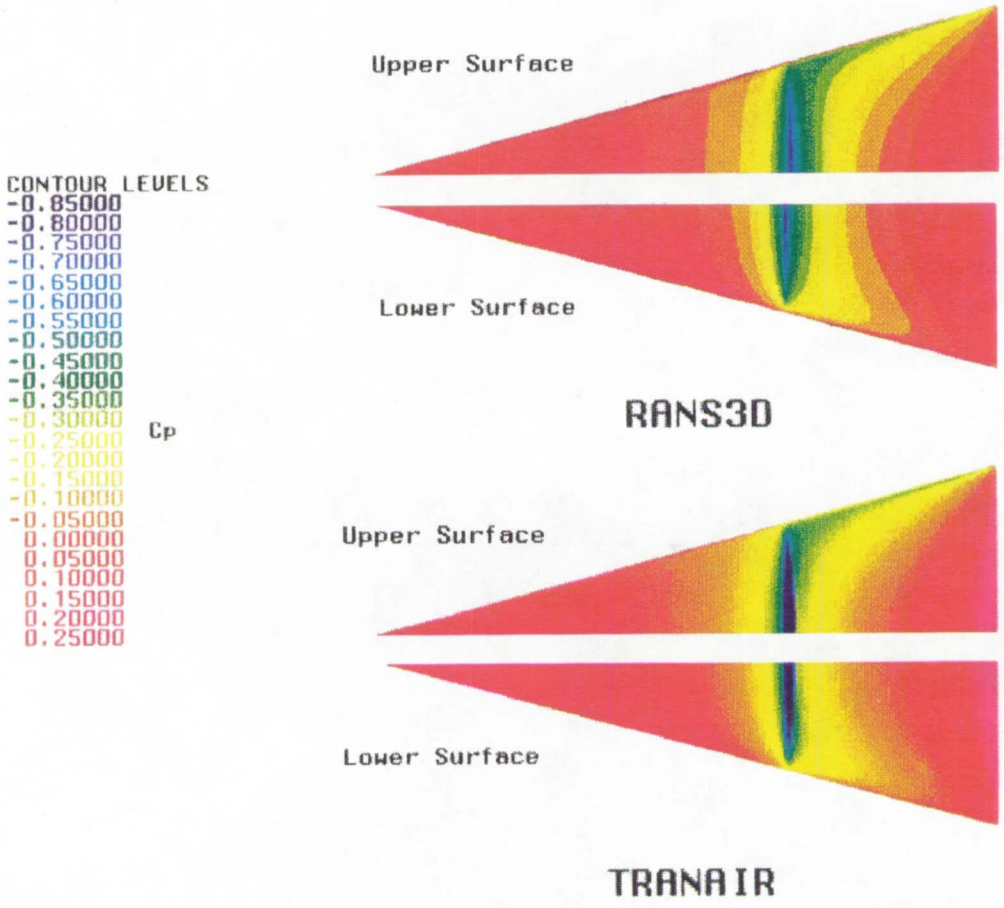


Figure 5.18. Comparison of RANS3D and TRANAIR All-Body Color C_P Distribution, $M_\infty = 0.7$, $\alpha = 2°$

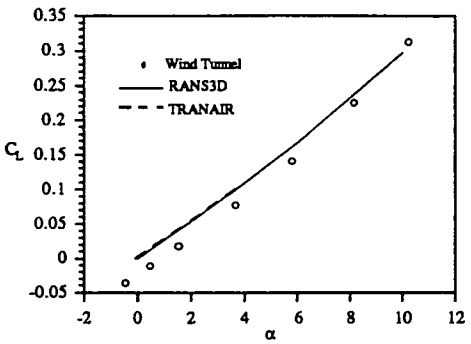
Each plot shows half of the lower surface juxtaposed to half of the upper surface. The general trends between TRANAIR and RANS3D seem to be very close as was shown in the sectional C_P comparisons. The one area of significant difference is, as before, the magnitude of the pressure loss, shown in dark blue. At this location the TRANAIR plot shows lower C_P values, corresponding to a larger change in pressure. It is also interesting to compare the upper surface pressure distributions to the lower surface pressure distributions. The angle of attack is only 2° so there should not be a large difference. One noticeable trend is that on the lower surface there is a larger area of high pressure on the forebody, while on the upper surface the transition to lower pressures begins sooner on the body. The higher pressure on the lower surface translates into the generation of lift. Another characteristic evident in both plots is a slightly lower pressure area near the leading edge on the upper surface that begins near the maximum thickness location and extends towards the trailing edge.

Comparison of Forces and Moments

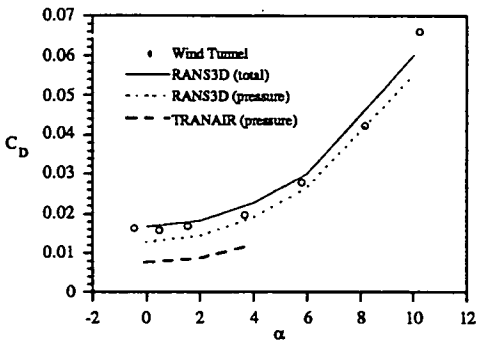
The wind tunnel force and moment data obtained from the 19 inch model was available for comparison with the numerical results at the Mach numbers of 0.9, 1.1, and 1.3. The principle quantities examined are lift, drag, and pitching moment. The drag is divided into two types. Since TRANAIR is an inviscid code it can only provide the pressure drag. RANS3D is a viscous code so it can also account for skin friction drag. The total experimental drag is compared to the total drag from RANS3D and to the pressure drags from TRANAIR and RANS3D.

The coefficients of lift, drag, and pitching moment versus angle of attack for the freestream Mach number of 0.9 are shown in Figure 5.19. The C_L curves for TRANAIR and RANS3D overlap each other. Both curves are shifted up from the experimental data, but as previously mentioned the wind tunnel results were affected by the sting mount. Both sets of numerical results pass through zero C_L at zero α, which is where the

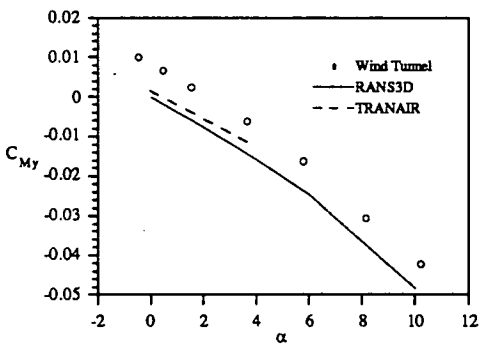
unaffected wind tunnel data would also be expected to pass. The remainder of both numerical plots follow the experimental one fairly well considering the shifted data.



a) C_L vs α



b) C_D vs α



c) C_{My} vs α

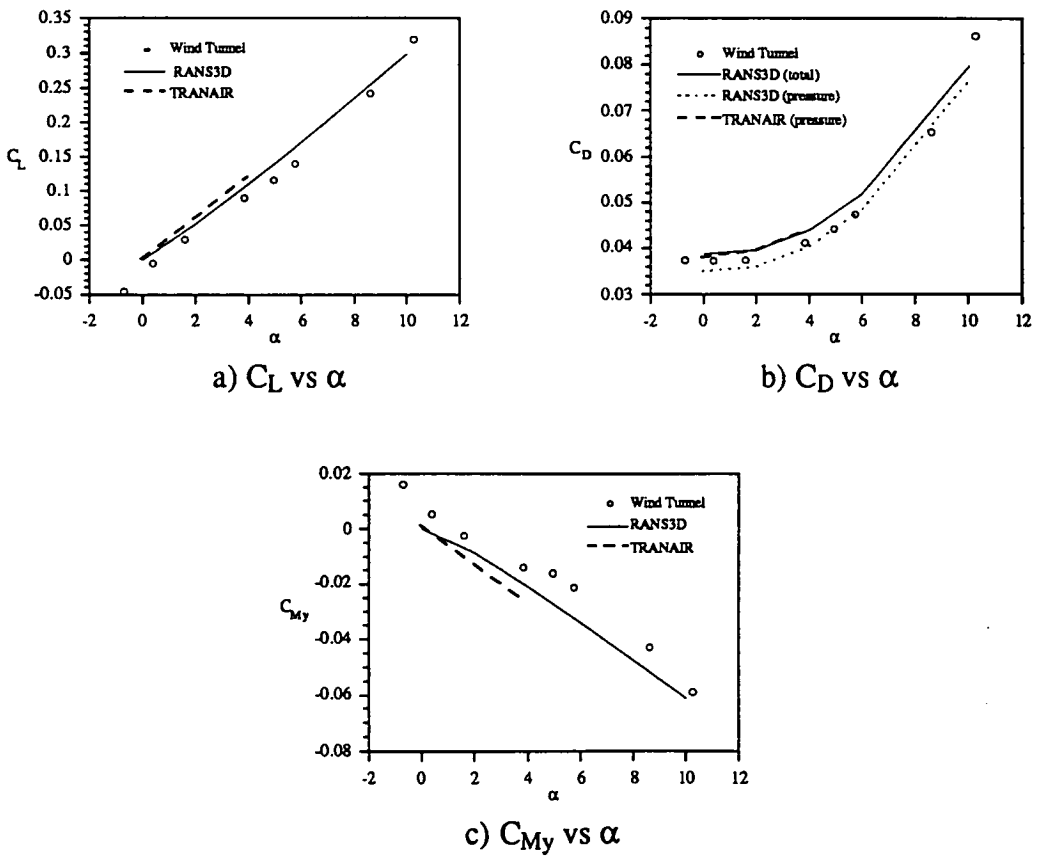
Figure 5.19. All-Body Forces and Moment, $M_\infty = 0.9$

The drag curves also compare favorably to the experimental data. The only available numerical total drag is from RANS3D, and it is slightly higher than the wind tunnel total drag. As seen in the pressure comparisons, there is an erroneous high pressure area acting upon the leading edge of the RANS3D results. This causes an increase in the RANS3D pressure drag. The pressure drag is the total drag less the friction drag. Therefore the total RANS3D drag will be higher than it would be if the problem with the boundary condition was fixed. The TRANAIR pressure drag is considerably less than the RANS3D pressure drag. Even if frictional drag is added to the TRANAIR pressure drag, the results are still somewhat lower than the experimental values. Drag is a quantity that is

very difficult for CFD to predict. The full potential code is not as well suited for drag prediction as the Navier-Stokes code.

The moments are taken about the aerodynamic center of the all-body which occurs in the x direction at 55% of the body length. Because of symmetry, the y and z locations of the aerodynamic center are both at the origin. The experimental pitching moments are also affected by the high pressure on the upper surface caused by sting interference. Both codes successfully model the trend of the pitching moment. Correct values are difficult to determine, but ideally the curve should have a zero C_{My} at a zero α.

At Mach 1.1 the results, shown in Figure 5.20, are somewhat different than at Mach 0.9. For this case, the C_L vs α slopes of the TRANAIR and RANS3D results differ slightly, although both pass through the origin. For the drag curve, the RANS3D results



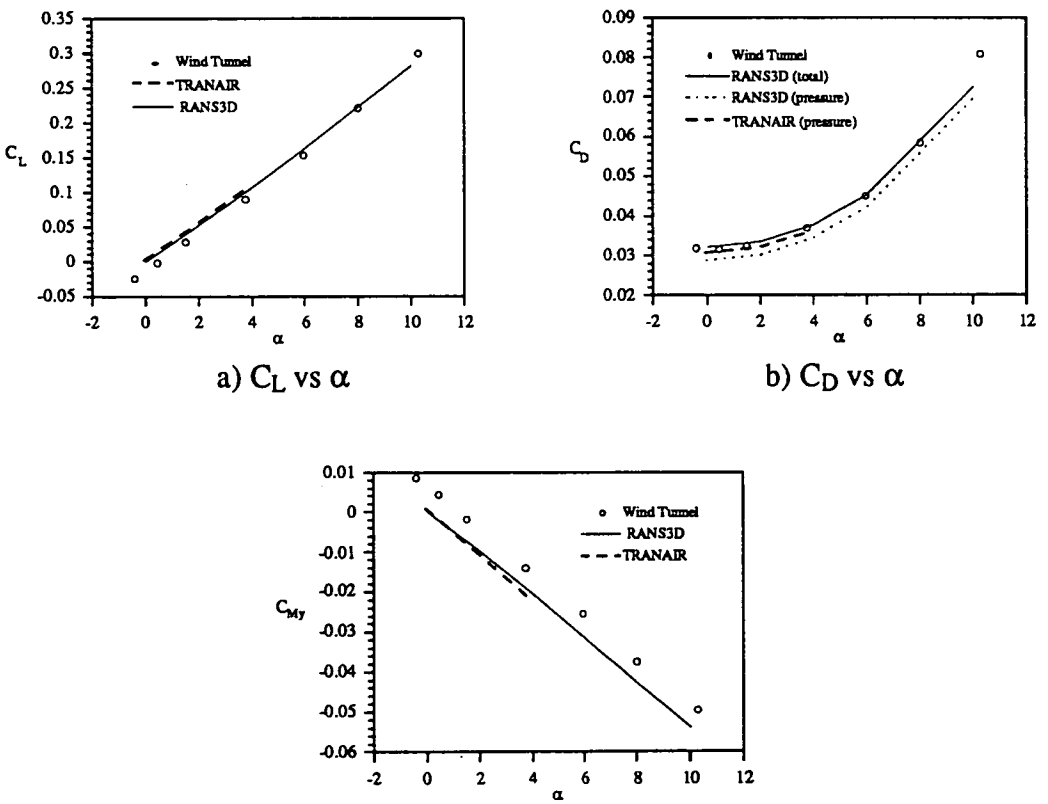
a) C_L vs α

b) C_D vs α

c) C_{My} vs α

Figure 5.20. All-Body Forces and Moment, $M_\infty = 1.1$

are about the same as before, with slightly higher pressure and total drags occurring because of the high pressure predicted by RANS3D on the leading edge. TRANAIR actually predicts the pressure drag to be larger than the total experimental drag. Previously the TRANAIR pressure drag was considerably less than the wind tunnel total drag. This high predicted drag could be partially explained by the high predicted lift which directly affects pressure drag, but this did not happen previously. This is just another deviation in drag prediction which is to be expected from using the full potential code. The numerical pitching moment curves follow the experimental trend and both pass through the origin. The RANS3D curve does show a smaller moment than TRANAIR and is closer to the wind tunnel pitching moment although both are still offset from the experimental because of sting interference.



a) C_L vs α

b) C_D vs α

c) C_{My} vs α

Figure 5.21. All-Body Forces and Moment, M_∞ = 1.3

At a Mach number of 1.3, both sets of computational results, Figure 5.21, seem very good. The C_L vs α curves are identical and compare well with the experimental one which is still shifted down, but not as much as before. The drag curves are also very good. The RANS3D total drag curve passes right through the experimental data. The TRANAIR pressure drag is still higher than the RANS3D pressure drag, but overall seems reasonable. It is unclear why the RANS3D drag prediction is better at this Mach number. It could be because at this Mach number the transonic effects are less severe than for Mach 0.9 and 1.1. The RANS3D code would be expected to perform better in the supersonic region than in the transonic region. The TRANAIR pressure drag probably just happens to fall close to the RANS3D pressure drag. The pitching moment comparison looks good for both TRANAIR and RANS3D except for the differences in magnitude.

To utilize the zero-lift wave drag results, a comparison of wave drag versus Mach number is made for the three analysis methods. The zero-lift wave drag is equivalent to the zero-lift pressure drag. For the all-body, the zero-lift condition occurs at $\alpha = 0°$. The experimental zero-lift pressure drag is determined by estimating the friction drag and subtracting it from the total drag. The results are shown in Figure 5.22. The wave drag

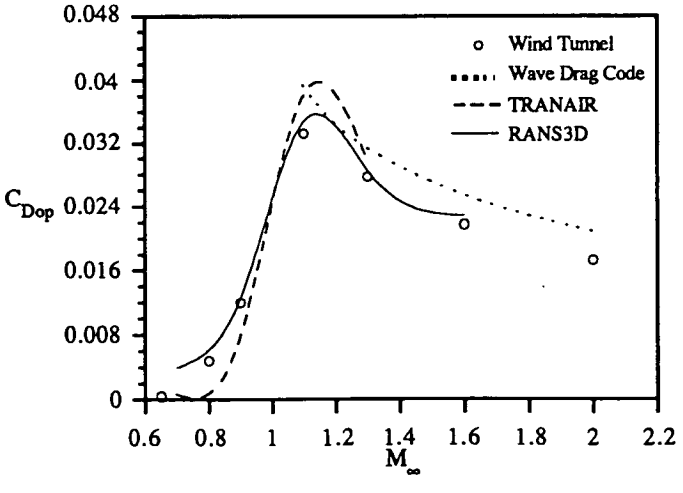


Figure 5.22. All-Body Zero-Lift Pressure Drag Comparison

code results are only for supersonic freestream Mach numbers. The magnitude predicted by the wave drag code is significantly higher than the experimental magnitude, but the trend of the data is very good. TRANAIR predicts the basic shape of the drag curve but as also seen in the force analysis, it under predicts the subsonic drag and over predicts the supersonic drag. RANS3D provides the best results for the zero-lift pressure drag.

Oblique Wing Supersonic Transport

Background

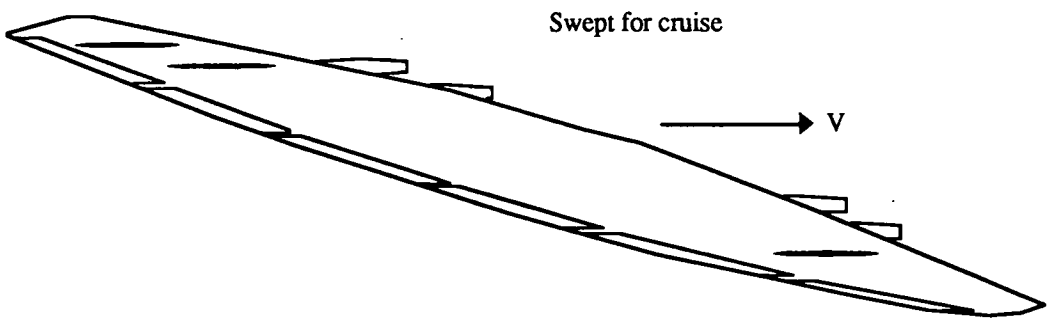
The concept of an oblique wing was originally proposed by R.T. Jones in the late 1950's. Recently, the oblique wing concept has been revived, which has resulted in a number of new proposals and studies. One proposal is the low supersonic transport Oblique Flying Wing (OFW) [Ref. 27]. This configuration combines the advantages of an oblique wing with a flying wing. The main advantage of using an asymmetrically swept wing over conventional configurations, is a reduction in subsonic and transonic drag. An extension of the OFW design is the Space Wing, an oblique flying wing capable of hypersonic flight, also being studied at NASA Ames.

Unlike the all-body configuration, the OFW does not have the support of extensive wind tunnel tests. Nevertheless, it is a good test case because of the unique requirements it presents. The oblique wing cannot be modeled using a plane of symmetry. This restriction excludes many codes from being used. Also the geometry modeling may change, because of the sweep angle, as the freestream Mach number changes. Therefore, the geometry and any computational grids, must be easily generated and modified.

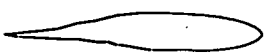
Configuration

The numerical model is based on the preliminary design of the Oblique Flying Wing shown in Figure 5.23. The wing has a near elliptic planform which is swept at 35° for takeoff and 70° in cruise. The engines and vertical tails can be swiveled to correspond to

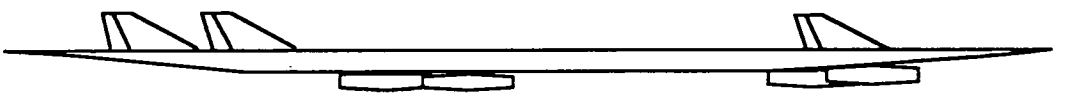
the wing sweep. The basic airfoil is designed for Mach numbers between 0.5 and 0.7. This is the range of Mach numbers normal to the leading edge, at which the OFW would operate. The sweep of the wing is determined so that the normal Mach number is kept at a constant value within this range.



a) Top View



b) Mid-Span Airfoil



c) Side View

Figure 5.23. The Oblique Flying Wing

TRANAIR Model

The oblique wing presented a great challenge to model with the TRANAIR code. The geometry used differs slightly from the proposed OFW. The restrictions of the arbitrary geometry modeling system for the synthesis code, require that the planform be entirely elliptic. The airfoil thickness is 14%. Two sweep angles, $\Lambda = 37°$ and $\Lambda = 66°$,

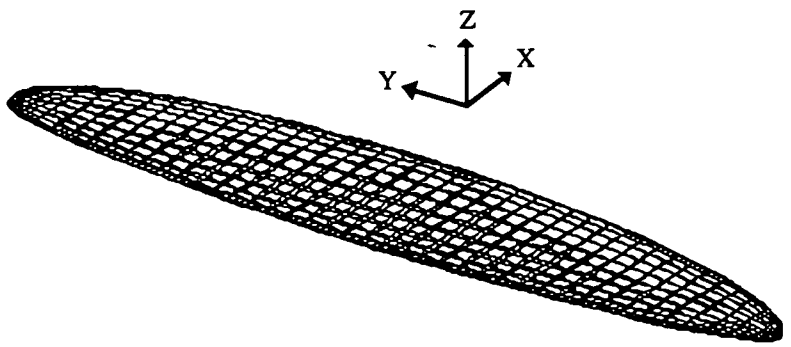
corresponding to freestream Mach numbers of 0.8 and 1.6 respectively, were run at angles of attack of 0°, 2°, and 4°.

The initial paneling scheme consisted of conventional rectangular panels with the trailing wake oriented at the sweep angle, which corresponds to the TRANAIR yaw angle. This method only required the trailing wake to be altered for different sweep angles. Unfortunately this paneling scheme was not able to give a converged solution. The problem appeared to be that the angle between the freestream flow and the leading edge of the panels, equivalent to the sweep angle, was too great and disrupted the solution procedure.

The paneling scheme finally chosen, has the panels rotated according to the sweep angle so that they are in the streamwise direction. This means that for each sweep angle a new paneling scheme is required to maintain the alignment of the panels. The paneling scheme for $\Lambda = 37°$ is shown in Figure 5.24. In the top view, the flow would be coming from the left and the wake, not pictured, would extend directly back from the trailing edge with both boundaries parallel to the freestream flow. The configuration is divided into two networks of panels, the upper surface and the lower surface. Clustering of panels is highest at the leading edge, the trailing edge, and the wing tips. The $\Lambda = 37°$ model contains 1,104 panels.

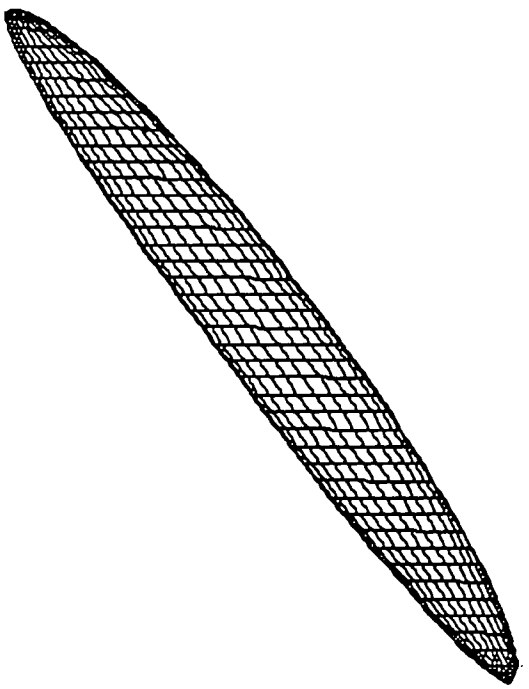
The paneling for the $\Lambda = 66°$ case was more difficult because of the higher sweep. The larger sweep angle caused panels with a large fineness ratio (length/width) to occur near the tips. This proved difficult for TRANAIR to handle. The solution to the problem is to remove part of the tips so that the tip edges are parallel to freestream flow. Doing this requires that end-plate networks be used to connect the upper and lower networks at the tips. The total number of panels for this model is 1,176.

The target computational residual was again 10^{-7}. Run times varied from 1.1 to 1.7 CPU hours on the CRAY Y-MP. CPU times increased as the angle of attack was increased and also as the freestream Mach number was increased from 0.8 to 1.6.



a) Isometric View

b) Side View



c) Top View

d) Front View

Figure 5.24. TRANAIR Surface Paneling of Oblique Flying Wing, $\Lambda = 37°$

Pressure Distributions

Unlike the all-body, there is no experimental pressure data to compare the numerical results with. The pressure distributions, however, can be examined graphically to analyze trends and overall pressure effects. Figure 5.25 shows the upper and lower surfaces for the $M_\infty = 0.8$, $\Lambda = 37°$, and $\alpha = 0°$ case. This C_P distribution shows a trend of higher pressure towards the forward swept end of both the upper and lower surfaces. The highest pressure appears on a thin section of the leading edge.

At $M_\infty = 0.8$ and $\alpha = 4°$, Figure 5.26, there is again a high pressure area on the forward swept trailing edge of the upper surface. On the lower surface the trend is reversed, with the highest pressure occurring on the leading edge, especially on the aftward swept half. In these results significant pitching and rolling moments are shown, which are supported by the moment output. Other than these two high pressure areas, the pressure distribution on the upper and lower surface seems to indicate an elliptic type lift distribution with the maximum lift being produced near mid-span.

The pressure distribution for $M_\infty = 1.6$, $\Lambda = 66°$, and $\alpha = 0°$, is shown in Figure 5.27. At this Mach number and sweep angle, the pressure distribution is much different from the subsonic case. In Figure 5.27, the low pressure regions appear to be shifted towards the forward swept part of the wing. It is interesting to note the two high pressure area which occur on the upper surface of both tips. These areas also correspond to areas where TRANAIR had difficulty analyzing. To prevent the solution from diverging, it was necessary to limit grid refinement near the tips. Therefore, although the solution for this case converges, the high pressure areas at the tips should be viewed with some skepticism.

Figure 5.28 shows the $M_\infty = 1.6$, $\alpha = 4°$ case, which did not fully converge. The problem with this case is also the tip regions. The very high pressure area near the forward swept tip on the upper surface and the aft sweep tip on the lower surface seem very irregular. To handle regions such as this, requires a certain amount of manipulating LBO

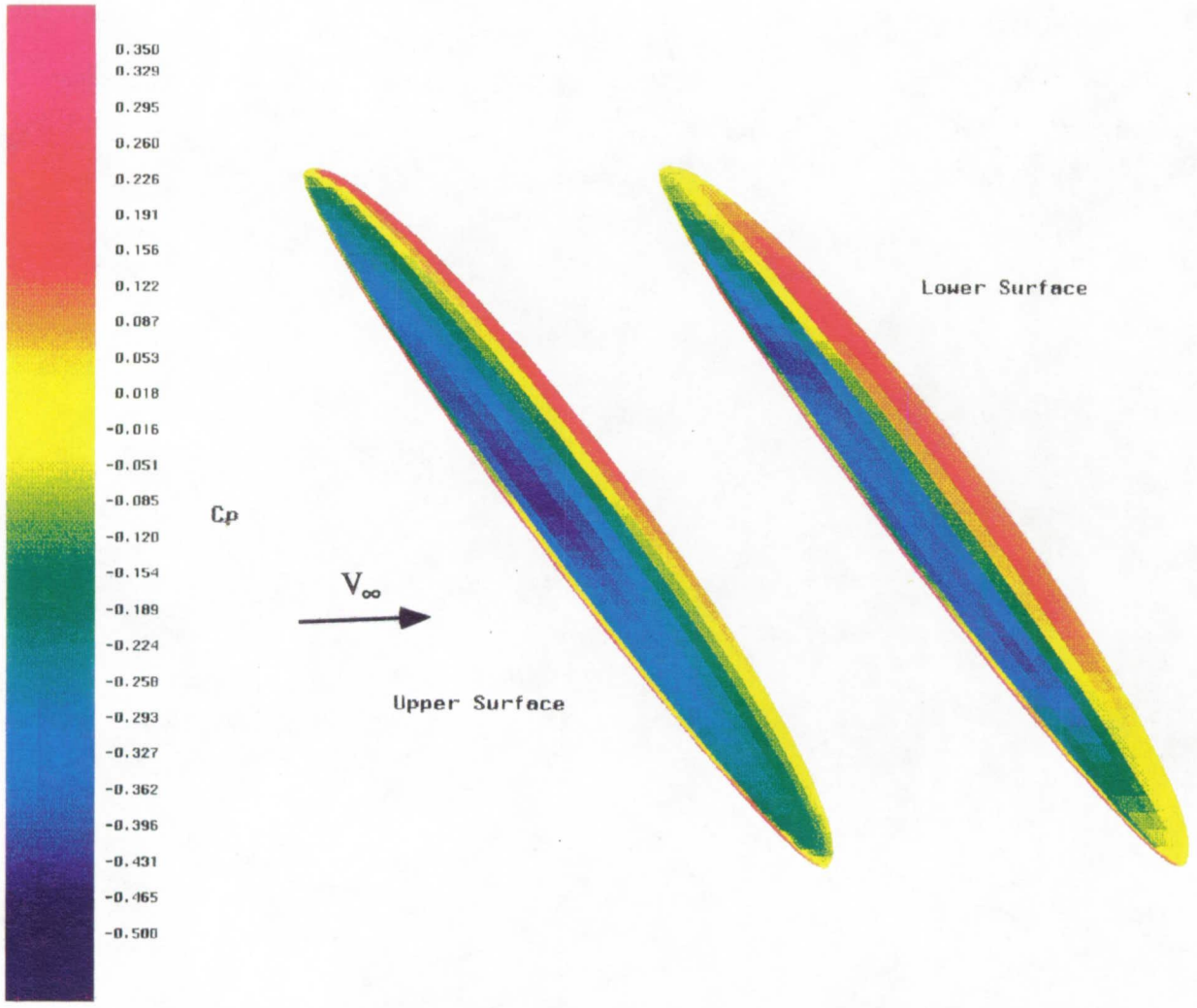


Figure 5.25. TRANAIR Oblique Flying Wing Color C_P Distribution Plot, $M_\infty = 0.8$, $\Lambda = 37°$, $\alpha = 0°$

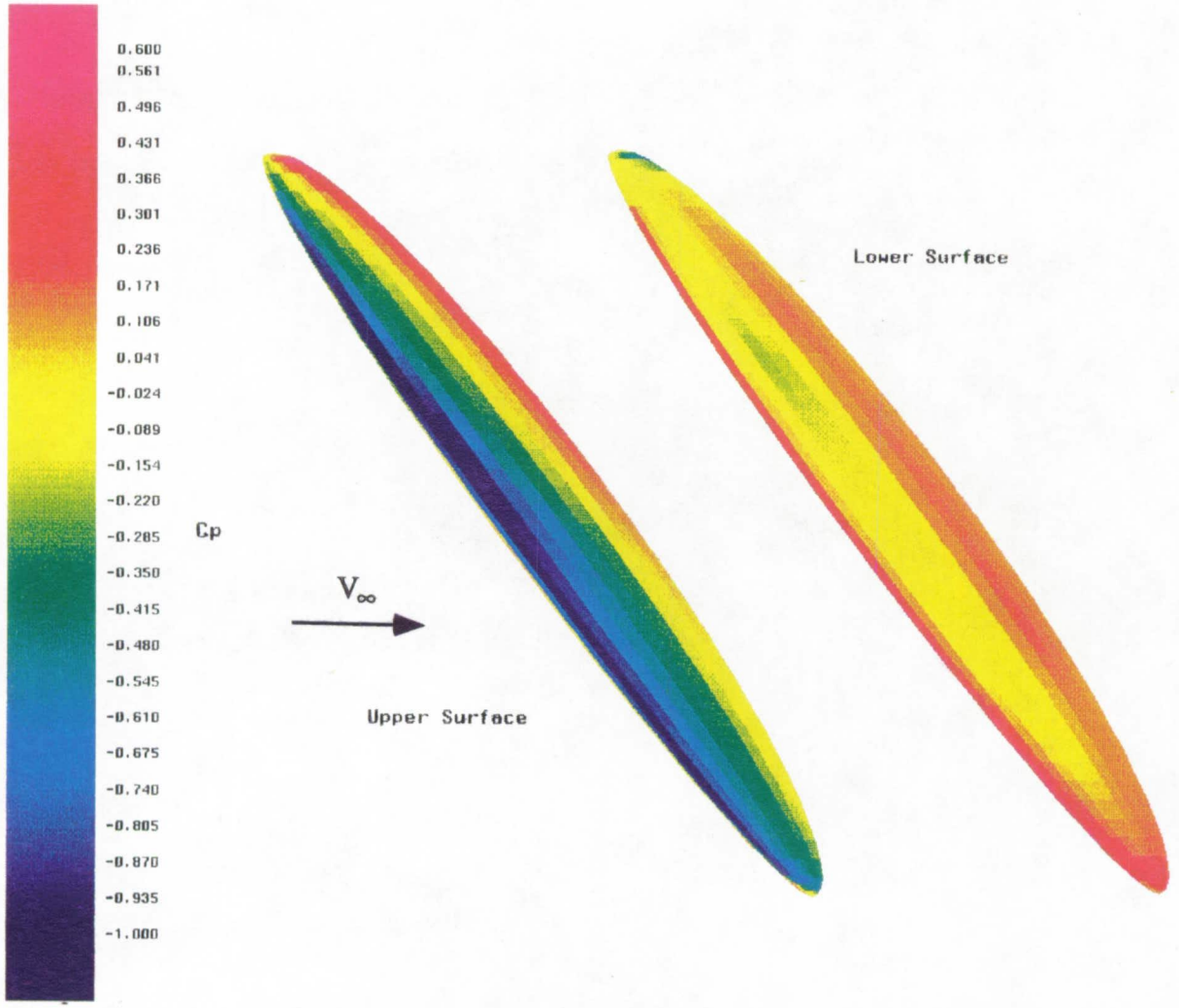


Figure 5.26. TRANAIR Oblique Flying Wing Color C_P Distribution Plot, $M_\infty = 0.8$, $\Lambda = 37°$, $\alpha = 4°$

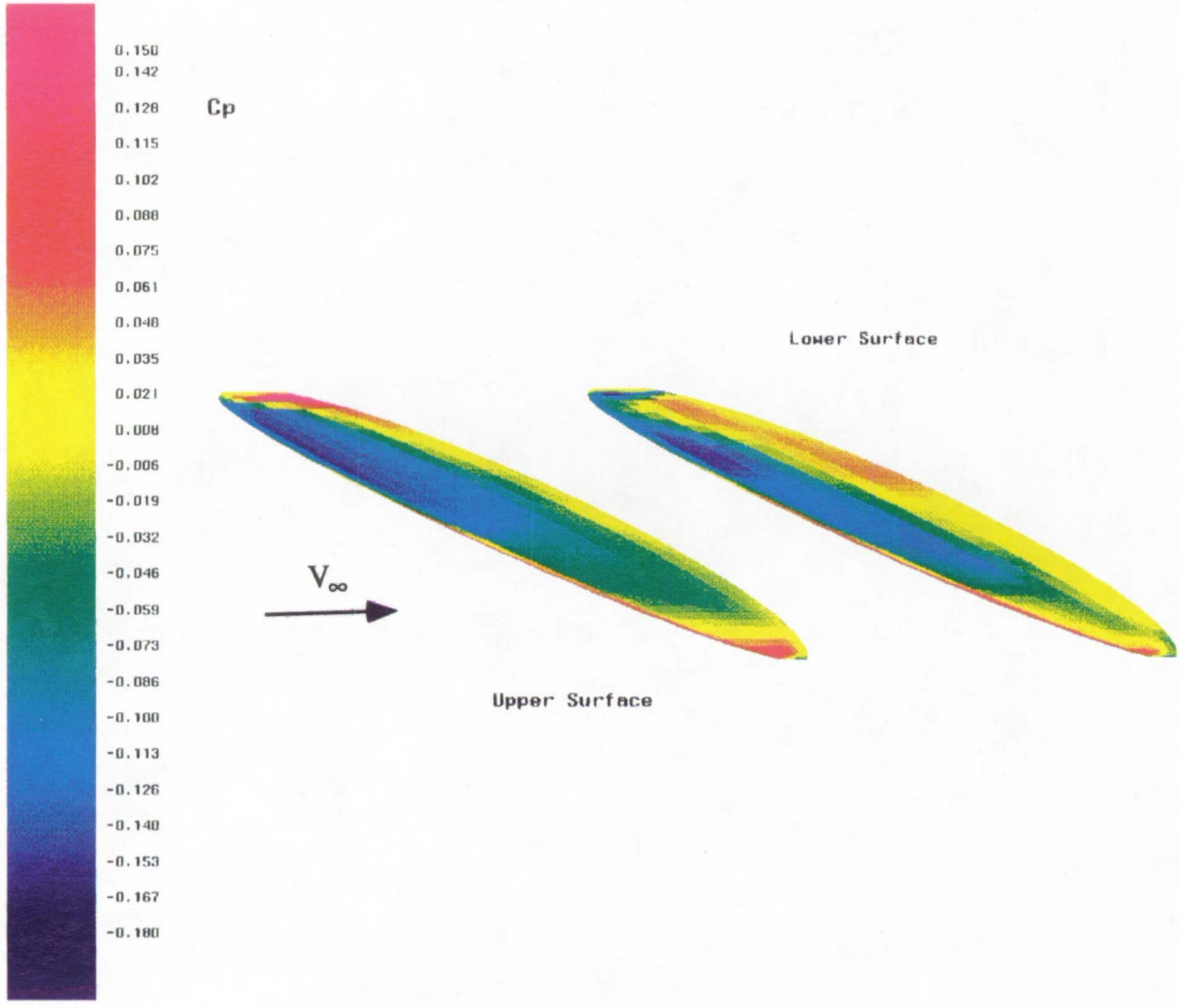


Figure 5.27. TRANAIR Oblique Flying Wing Color C_P Distribution Plot, $M_\infty = 1.6$, $\Lambda = 66°$, $\alpha = 0°$

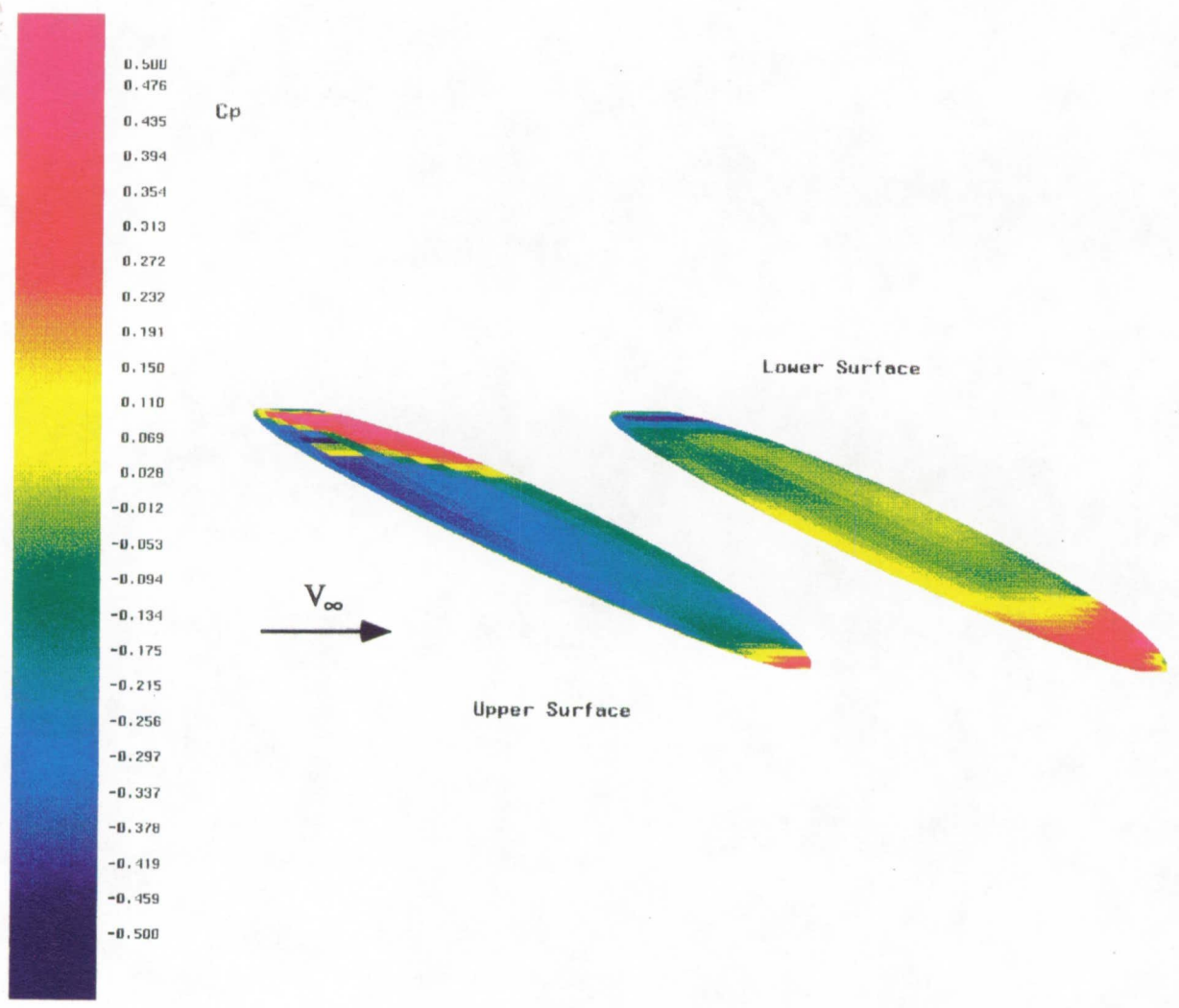
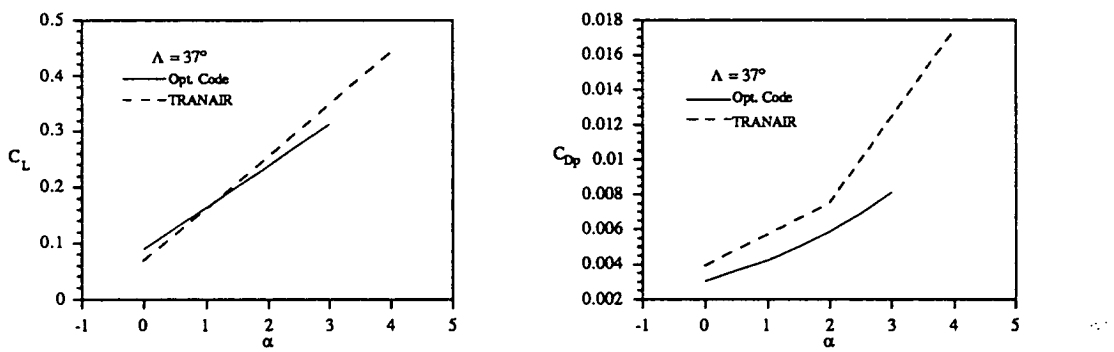


Figure 5.28. TRANAIR Oblique Flying Wing Color C_P Distribution Plot, $M_\infty = 1.6$, $\Lambda = 66°$, $\alpha = 4°$

regions and other gridding parameters. Because of the presence of these high pressure regions near the tips, the moment results, especially C_{Mx}, should be taken cautiously.

Forces and Moments

As with the pressure distributions, there is no experimental data to compare the numerical force and moment results with. There is, however, other predicted data available for comparison. These results are obtained from an aerodynamic optimization code written by Kroo and based upon the work of Smith [Ref. 28]. The optimization code and TRANAIR results are plotted for comparison. Figure 5.29 shows the results and general trends for the lift and the pressure drag versus angle of attack for a 37° sweep angle. For the $M_\infty = 0.8$, $\Lambda = 37°$ case, the results match very well. The slope of the C_L vs α curves are different but fairly close. The pressure drag curves also have similar shapes although the values differ considerably. TRANAIR predicts a higher C_{Dp} than the optimization code.

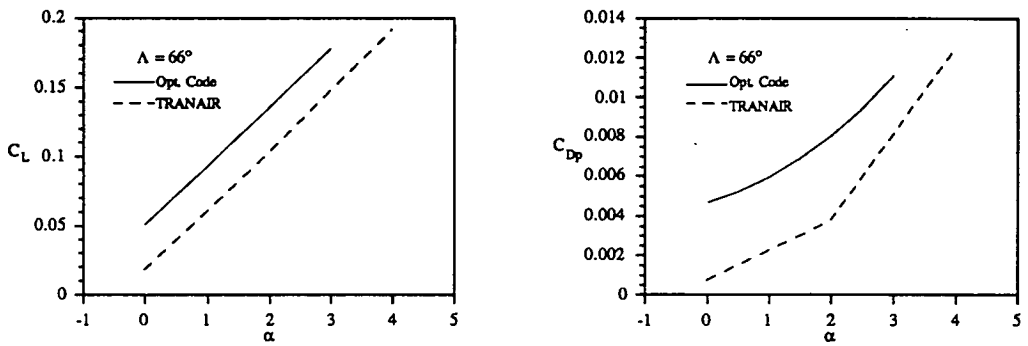


a) C_L vs α b) C_{Dp} vs α

Figure 5.29. TRANAIR Force Results for the Oblique Flying Wing, $\Lambda = 37°$, $M_\infty = 0.8$

For $M_\infty = 1.6$ and $\Lambda = 66°$, Figure 5.30 the C_L vs α curves have the same slope but the optimization code predicts higher values of C_L. For the pressure drag, the two methods differ greatly. TRANAIR predicts a much lower pressure drag curve. The shapes of the

curves are similar but not identical. It is difficult to interpolate these results since there is no experimental data for comparison. It can be expected that TRANAIR should provide more accurate results because it is a more refined method. The similarity of the two lift predictions is an encouraging sign. The drag predictions, however, can cause reasonable doubts in both methods until actual experimental data can be used to further support one or neither.



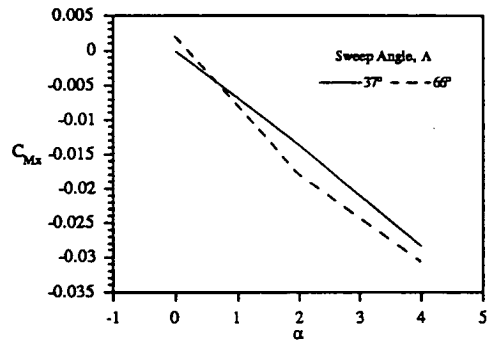
a) C_L vs α

b) C_{Dp} vs α

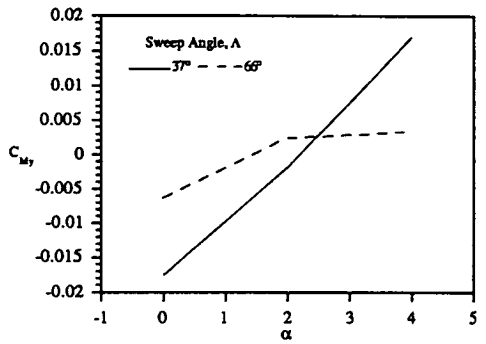
Figure 5.30. TRANAIR Force Results for the Oblique Flying Wing, $\Lambda = 66°$, $M_\infty = 1.6$

The moments are taken with respect to a stationary body axis system. The x-axis is always taken as the freestream flow direction when the sweep angle is zero. The y-axis is taken along the span of the wing which is measured from tip to tip regardless of the sweep angle. The z-axis is the standard vertical axis. The aerodynamic optimization code does not provide moment data. The TRANAIR moments alone are shown in Figure 5.31. The rolling moment, C_{Mx}, becomes negative as the angle of attack is increased. At both sweep angles the rolling moments are about the same. The pitching moment, C_{My}, seems to behave reasonably. The exception being at $\Lambda = 66°$, where it begins to level off between 2° and 4°. This is most likely due to the high sweep angle which stabilizes the increasing pitching moment. The yawing moment, C_{Mz}, for $\Lambda = 37°$ increases only slightly as α is increased. For the sweep of $\Lambda = 66°$, the yawing moment changes much more rapidly as α

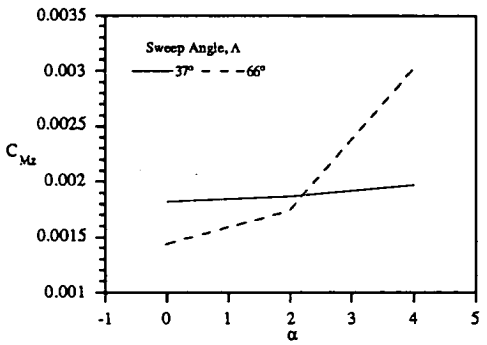
is increased. This is because at this sweep angle the tendency is for the moment to turn the wing so that the span is facing directly into the flow.



a) C_{Mx} vs α



b) C_{My} vs α



c) C_{Mz} vs α

Figure 5.31. TRANAIR Moment Results for the Oblique Flying Wing, $\Lambda = 37°$ and $\Lambda = 66°$

TRANAIR Solution Adaptive Grid

Running TRANAIR on the all-body showed that the adaptive gridding option is a better choice than the grid sequencing option. This is especially true for a complex case such as the OFW. Figure 5.32 shows the z = 0 plane of the fifth and final adaptive grid for the $M_\infty = 0.8$, $\Lambda = 37°$, $\alpha = 4°$ case. The surface paneling of the configuration and trailing wake are shown in white and the global grid is in red. The LBO's, regions of special interest, are shown outlined in yellow. This is a view of the xy-plane with the x-axis

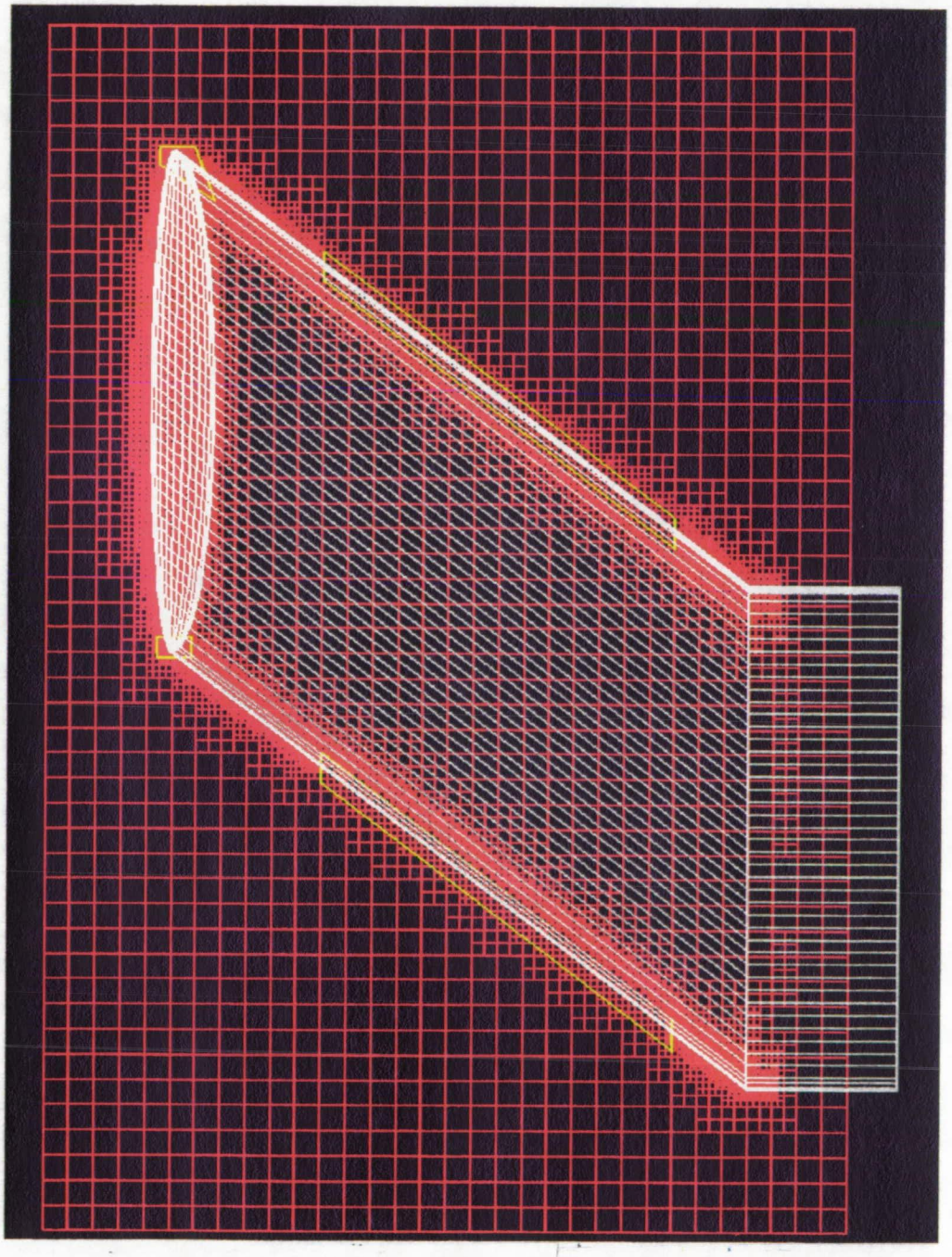
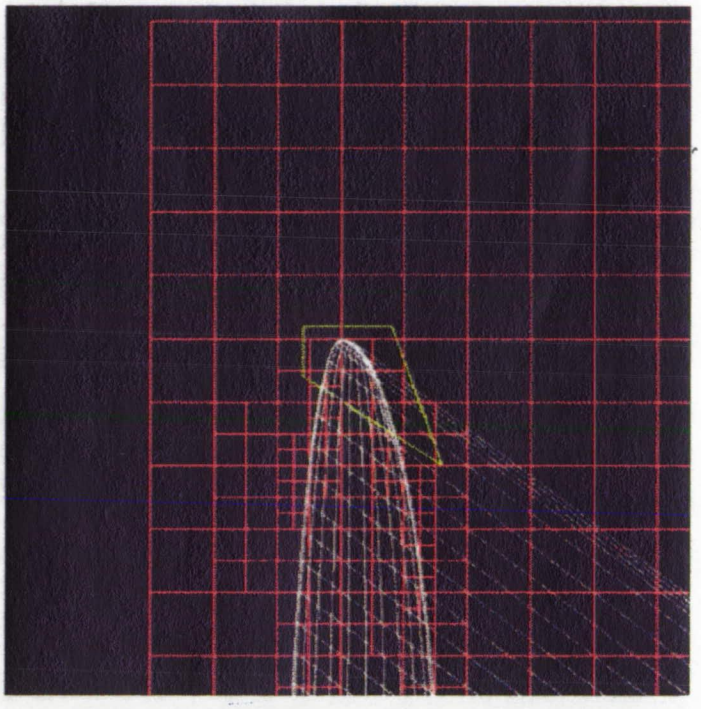


Figure 5.32. Grid Cut at z = 0, of Fifth and Final Adaptive Grid for the Oblique Flying Wing, $M_\infty = 0.8$, $\Lambda = 37°$, $\alpha = 4°$

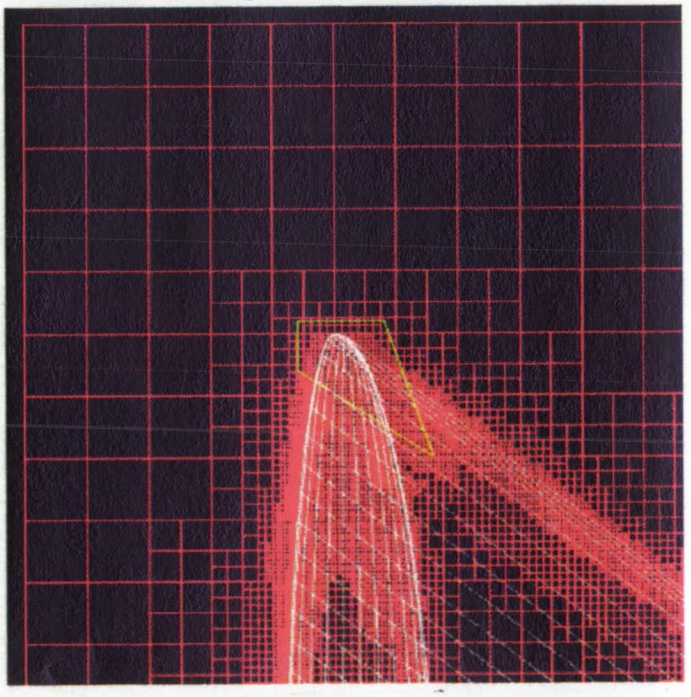
horizontal on the page and the y-axis vertical. The yaw angle, β, is -37° which corresponds to the sweep direction of the trailing wake. TRANAIR alters the original wake near the downstream boundary. The original wake's end points are out of the computational domain, but along the sweep angle of the wake. TRANAIR inserts a break in the trailing wake inside of the computational domain and then extends the wake straight back in the x direction. This feature of the code does not significantly affect the solution.

Four LBO regions are used in this case to limit grid refinement. TRANAIR normally tends to cluster a large number of grid boxes along the edges of the trailing wakes. A large amount of refinement is not needed here and will only result in a large number of wasted grid boxes if left unchecked. The other two areas where refinement is limited by LBO's, is near each of the tips. These areas exhibit a tendency to become heavily refined and then begin to diverge from the surrounding solution. By limiting these areas to only four levels of refinement, as opposed to six globally, the solution is generated nicely without many problems. Each sweep angle as well as angle of attack, requires slightly different changes in grid control and LBO specification to achieve a converged solution.

Some effects of the solution adaptive gridding on the global grid are illustrated in Figure 5.33. This figure shows the forward swept tip of the OFW and surrounding grid for the first and the last computational grids. In the first grid there is only a very little amount of refinement. By the fifth grid, heavy refinement is evident. The large refinement near the leading edge causes an enlargement of the global domain. Figure 5.33a shows the minimum x boundary on the left hand side of the illustration as extending a little over two global grid boxes in front of the configuration. By the fifth grid, Figure 5.33b, two additional rows of global grid boxes have been added to the computational domain because of refinement emanating from the vehicle's surface.



a) Grid 1



b) Grid 5

Figure 5.33. Grid Cuts at z =0, Comparing First and Final Adaptive Grids Around the Forward Swept Tip of the Oblique Flying Wing, $M_\infty = 0.8$, $\Lambda = 37°$, $\alpha = 4°$

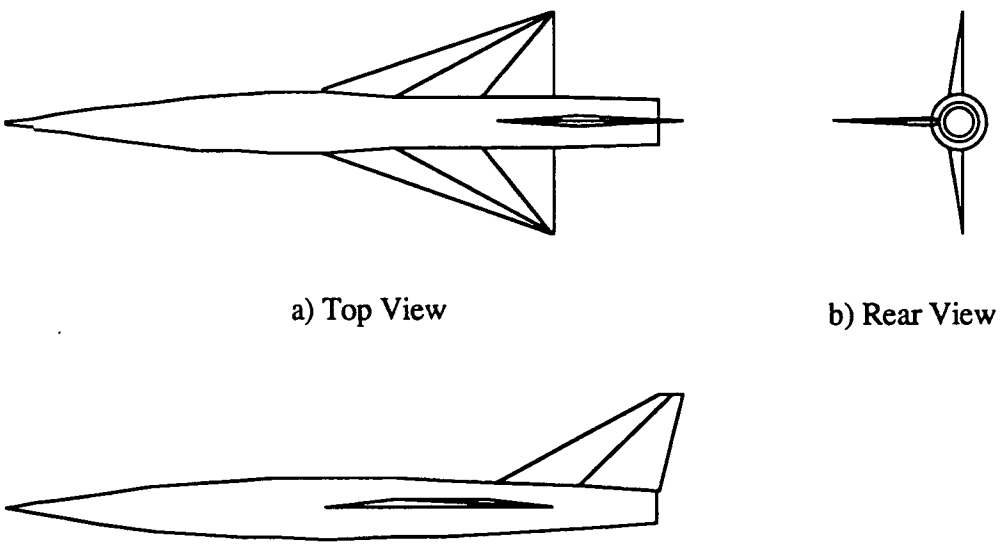
C-2

Wave Drag Code Results

Because the wave drag code is simple and fast to run, other hypersonic configurations were tested to better evaluate code performance. The results of two configurations, a Sears-Haack wing-body and a waverider are presented here. The waverider does not have experimental data to compare the wave drag results with, but the general trends are still valuable.

Sears-Haack Wing-Body

The Sears-Haack wing-body configuration, Figure 5.34, is the hypersonic reference model from the wind tunnel study of Reference 29. The forebody has circular cross sections with a Sears-Haack profile. The aftbody is a cone frustum. The delta wing has a flat lower surface and a leading edge sweep of 70°. Wind tunnel tests were conducted for the body alone, the wing-body, and the wing-body-tail combinations.



a) Top View

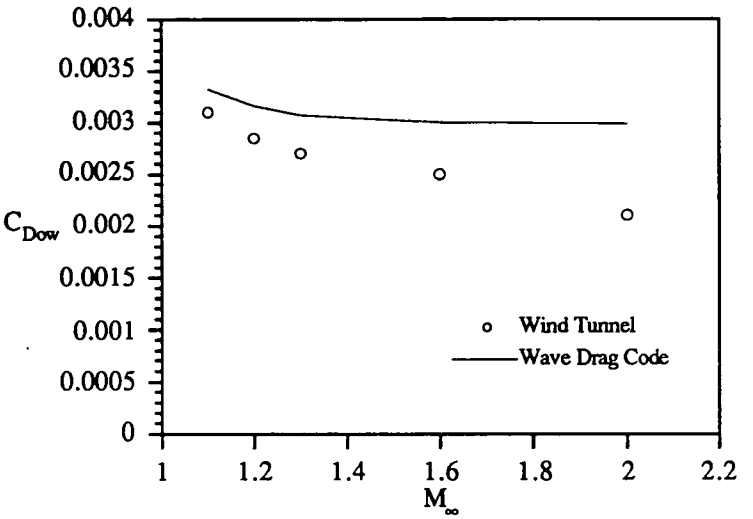
b) Rear View

c) Side View

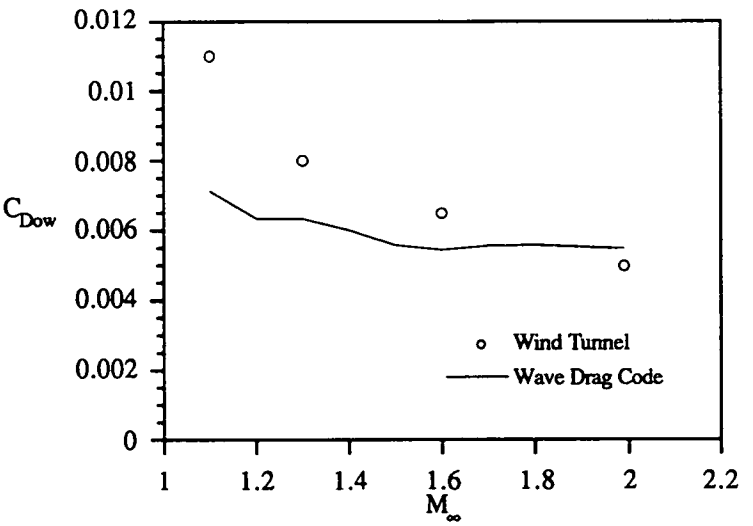
Figure 5.34. The Sears-Haack Wing-Body

The wave drag analysis is performed for the body alone and the wing-body case. The angle of attack is held constant at zero. The number of longitudinal cuts, NX, is 80

and the the number of cutting planes, Nθ, is 12. Supersonic freestream Mach numbers ranging from 1.05 to 2.0 are examined. In Figure 5.35, the zero-lift wave drag for both the body alone and the wing-body are compared to the experimental values. For the body



a) Body Alone



b) Wing-Body

Figure 5.35. Sears-Haack Wing-Body Zero-Lift Wave Drag vs Mach Number for Body Alone and Wing-Body, NX = 80, Nθ = 12, α = 0°

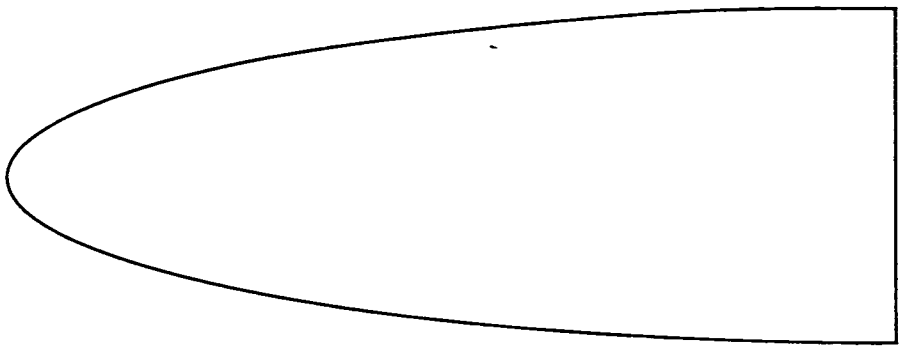
alone case, the wave drag code predicts a curve which begins slightly higher than the experimental results, but diverges as the Mach number is increased. The general trend of a decrease in wave drag with an increase in Mach number is present, but the rate at which the wave drag results decrease is a problem. In the wing-body results in Figure 5.35b, the wave drag code begins by under predicting the experimental values and then decreases at a slower rate than the experimental curve.

In both the body alone and the wing-body case the wave drag code does not match the experimental wave drag very well. It would be difficult to use the code's results except to conclude that the wave drag decreases with Mach number. However, other cases tested showed the wave drag to drop with Mach number initially and then begin to rise again. The wing-body shape should be well suited for the code, but it is not well suited as a practical hypersonic transport. A more realistic series of hypersonic configurations are the waverider type aircraft.

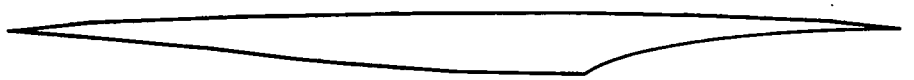
JA11 Waverider

The JA11 is a hypersonic waverider configuration. The JA refers to the configuration's developer, John Anderson, and the 11 refers to the wave angle in degrees. The JA11 configuration along with details of waverider theory can be found in Reference 30. The JA11, shown in Figure 5.36, has a parabolic planform. A nozzle has been added to the rear section of the configuration, and can be seen in the side view of Figure 5.36.

There is no wind tunnel data to compare the JA11 wave drag results to, but it was still tested to gauge the code's performance on a configuration of this shape. The analysis was performed for Mach numbers from 1.05 to 2.0 at an angle of attack of zero. The results, Figure 5.37, are for cases using NX values of 40, 60, and 80. The blunt nose on the JA11 makes the wave drag code valid for only a limited range of Mach numbers before the slope of the body exceeds the slope of the Mach cone. The Mach number at which the



a) Top View



b) Side View

c) Front View

Figure 5.36. The JA11 Waverider

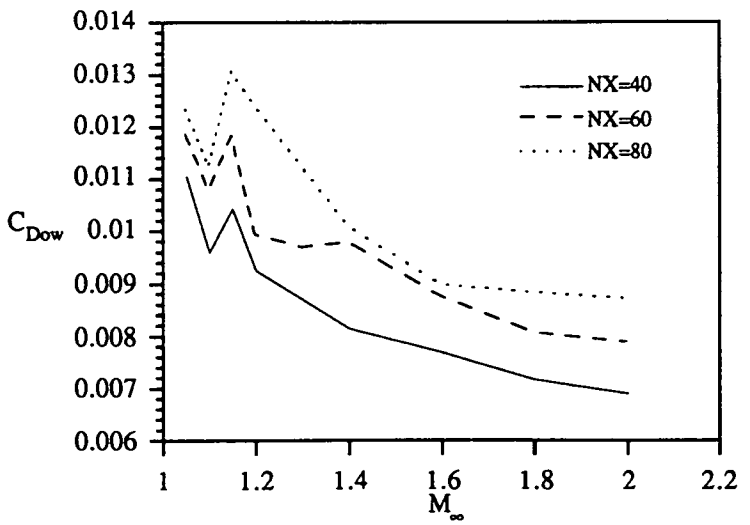


Figure 5.37. JA11 Zero-Lift Wave Drag vs Mach Number for NX = 40, 60, and 80 with Nθ = 12 and $\alpha = 0°$

slope violations begin is around 1.15. All Mach numbers greater than this produce results which must be taken cautiously.

The results of the three cases differ dramatically. All three curves have spikes which appear near $M_\infty = 1.15$, where the body slope violations begin. Before that the curves are the same shape, just offset from each other. After about $M_\infty = 1.1$, the differences between curves increases greatly. Runs at other values of $N\theta$ showed similar results. Based on these results the wave drag code does not seem well suited for handling this type of blunt nose configuration which is typical of many NASP designs.

Discussion

From the analysis results and the comparisons of the three codes, TRANAIR is selected as a potential method. From the all-body analysis TRANAIR demonstrated it has the potential to compete with a more advanced code without an increase in complexity. Subsonically TRANAIR did very well against RANS3D. The only area of major difference was in the magnitude of the pressure loss where neither method appeared to be more correct. The RANS3D method also exhibited difficulty predicting the pressure along the leading edge. This was attributed to an improper handling of the boundary condition by the code. This can be weighed in TRANAIR's favor since it implies more time and care must be given to solving a problem using RANS3D. For a subsonic freestream Mach number in the transonic range the results for both codes were poorer. Considering the peculiarity of the flow, the results are still good. The area of difference between the codes again was in the magnitude of the pressure loss. While neither method completely and accurately predicted the expansion region the RANS3D code was more consistent.

For the supersonic freestream cases RANS3D did very well while TRANAIR had some trouble with the pressure loss again. Although TRANAIR is not as well suited to handle supersonic flow as RANS3D, it could have provided better results if the adaptive gridding method was used in place of the grid sequencing method. This was done for a

subsonic case on the all-body and there were noticeable improvements. If it were done for a supersonic case much more significant improvements would be expected. Also the Oblique Flying Wing was successfully tested using the adaptive gridding method. When the grid sequencing method was used for the Oblique Flying Wing the solution would not converge.

The comparison of force and moment results showed that TRANAIR and RANS3D both did equally well at predicting the lift and the pitching moment. RANS3D, however, did much better at the more difficult task of predicting the drag. Although TRANAIR provides only the pressure drag, it seems to have difficulty being either consistently lower or higher than the expected pressure drag.

The major drawback of the TRANAIR code is the required computing time. Although a CPU time under 1.5 hours for transonic calculations is good in the CFD field, it is very high for a synthesis code. A complete integration of TRANAIR into HAVOC is not feasible with these large computation times. However, TRANAIR can be very valuable in evaluating configurations which are still early in the design process, for several reasons. One reason is that transonic performance is a critical driver for these hypersonic vehicles. Therefore, significant importance should be given to having a reasonable analysis technique. For the bulk of the preliminary calculations, empirical or semi-empirical techniques may be adequate, but as illustrated with the wave drag code, they are often very limited and unreliable. Another reason to use TRANAIR over more accurate methods such as Euler or Navier-Stokes codes, is the faster computing time and even more importantly the grid generation time. The biggest advantage TRANAIR has over these types of codes is the ability to automatically generate the computational grid.

Another possibility that was not fully examined is a lower order CFD method such as the transonic small disturbance WIBCO-PPW code. The use of this code, which also has automated grid generation, would reduce the computational time significantly from TRANAIR. The reduction in CPU time would probably still not be enough to allow

integration into HAVOC, but it would allow additional calculations to be done at earlier stages in the design. The penalty, however, would be a reduction in accuracy and capability, such as handling supersonic freestream Mach numbers. For these reasons TRANAIR was chosen to be researched further.

CHAPTER 6

Geometry Interface

Objective

A goal of this study is to integrate the selected transonic flow analysis method into the hypersonic optimization code, HAVOC. Based upon the results in Chapter 5, it is not feasible to completely integrate TRANAIR into HAVOC. Nevertheless, if TRANAIR is to be used in any form with HAVOC, it is necessary to develop an interface to allow some interaction between the codes. The geometry supplied to HAVOC is generated analytically from the arbitrary body geometry modeling system [Ref. 31]. The modeling package is capable of providing geometric output in the SHADE format [Ref. 32]. Generating a TRANAIR geometry file can be a time consuming process. An interface between the arbitrary geometry modeling system and TRANAIR will allow changes in the case geometry to be quickly reflected in the TRANAIR geometry.

Input Geometry Definition

The input geometry is in the form used by the graphics program SHADE. The SHADE geometry file consists of one or more "parts". These "parts" are made up of at least two cross sections each. The cross section contains a cross section-size record, which specifies the number of points in that cross section, and the actual x-y-z data points. At the end of each "part" a "-999" appears as an end of cross section signal. The end-of-file record is a "-99999". The Fortran format for the x-y-z points is 3F20.16. A sample cross section in SHADE format is shown in Figure 6.1.

The geometry modeling package is continuously being modified and upgraded to make it more general and flexible. Several modifications were made to handle special requirements needed for the TRANAIR geometric models, specifically the oblique wing. Currently the geometry package can provide cross sections at user specified locations using one of several point spacing options. The point spacing in a cross section can be divided into segments based on equal divisions of the cross section length, or by equal angular divisions. An additional option is a cosine distribution of points, which clusters points near the edges of the cross sections.

```
   25
0.0097872866317630 0.4964111447334290 0.0000000000000000
0.0103906793519855 0.4950547218322754 0.0000246604868153
0.0121597349643707 0.4910778999328613 0.0001451352582080
0.0149738974869251 0.4847517013549805 0.0002788437122945
0.0186413843184710 0.4765072166919708 0.0003599625779316
0.0229122638702393 0.4669063389301300 0.0003678936336655
0.0274954810738564 0.4566033184528351 0.0003100775938947
0.0320786982774734 0.4463002681732178 0.0002144572790712
0.0363495796918869 0.4366993904113770 0.0001170910327346
0.0400170646607876 0.4284549057483673 0.0000461747185909
0.0428312271833420 0.4221287071704865 0.0000106603501990
0.0446002818644047 0.4181518852710724 0.0000007313094557
0.0452036745846272 0.4167954623699188 0.0000000000000000
0.0446002818644047 0.4181518852710724 -0.0000000000000010
0.0428312271833420 0.4221287071704865 -0.0000000000078279
0.0400170646607876 0.4284549057483673 -0.0000000011391663
0.0363495796918869 0.4366993904113770 -0.0000000289459496
0.0320786982774734 0.4463002681732178 -0.0000002591844463
0.0274954810738564 0.4566033184528351 -0.0000011035049283
0.0229122638702393 0.4669063389301300 -0.0000025537913189
0.0186413843184710 0.4765072166919708 -0.0000032873970213
0.0149738974869251 0.4847517013549805 -0.0000020683244202
0.0121597349643707 0.4910778999328613 -0.0000003668795046
0.0103906793519855 0.4950547218322754 -0.0000000015587208
0.0097872866317630 0.4964111447334290 0.0000000000000000
```

... additional cross sections ...

Figure 6.1. Sample SHADE Geometry Data

TRANAIR Geometry Requirements

The geometry definition for TRANAIR is separated from the input conditions by placing it into a network file. The first item to appear in the network file is the $POInt keyword. It can be followed on the same line by comments describing the data. The second line contains the variable KN, which specifies the number of networks associated with this particular $POI keyword. The $POI keyword can contain the entire geometry of the configuration, but for simplicity is limited to a single network, hence KN = 1. The next line contains the boundary condition, KT, and the switching parameter. The majority of the cases will have an impermeable surface as a boundary condition, KT = 1, except for the wakes which have their own boundary condition. The switching parameter is occasionally used when the rows and the columns of the network need to be interchanged to obtain an outward facing normal. If this is not needed, then the variable is omitted and TRANAIR uses the default setting. The fourth line contains three items, the number of rows, the number of columns, and the network identification string. The number of rows, NM, and number of columns, NN, are those of the current network. Because TRANAIR is a panel code, the NMxNN matrix of points, must be complete. The network identification is a short string of characters which describes the network and is optional, but is recommended for clarity.

The fifth line is the beginning of the actual point data. The data is entered by columns with two points per line. The format for one line is 6F10.6. If there is an odd number of rows, the last point of the column will appear by itself on one line with the format 3F10.6. The new column of data will then begin on the next line. When TRANAIR has read NMxNN points, it is finished with the network and looks for the next keyword which should be $POI, for another network, or $END to signal the end of the network file. A sample of a TRANAIR geometry network file is given in Figure 6.2. This portion shows two of the forty-nine columns of points for the upper surface network. The ordering of the points on a line are x, y, z, x, y, z.

```
$POINTS - UPPER SURFACE
1.
1.
      13.      49.                                                  UPSURF
0.009787 0.496411 0.000000 0.010391 0.495055 0.000025
0.012160 0.491078 0.000145 0.014974 0.484752 0.000279
0.018641 0.476507 0.000360 0.022912 0.466906 0.000368
0.027495 0.456603 0.000310 0.032079 0.446300 0.000214
0.036350 0.436699 0.000117 0.040017 0.428455 0.000046
0.042831 0.422129 0.000011 0.044600 0.418152 0.000001
0.045204 0.416795 0.000000
0.006717 0.498313 0.000000 0.007415 0.496745 0.000071
0.009462 0.492150 0.000279 0.012717 0.484839 0.000464
0.016961 0.475312 0.000569 0.021902 0.464218 0.000571
0.027205 0.452312 0.000478 0.032507 0.440406 0.000331
0.037448 0.429311 0.000181 0.041692 0.419784 0.000072
0.044947 0.412473 0.000017 0.046994 0.407878 0.000001
0.047692 0.406310 0.000000

                ... additional columns ...
```

Figure 6.2. Sample TRANAIR Geometry Data

Program Description

The Fortran program TACONV converts a SHADE geometry file into a TRANAIR network geometry file. A listing of the program appears in the Appendix. Modifications to the code were made to handle specific geometric problems as they arose. The need to model the oblique wing caused the incorporation of features such as wake rotation and construction of tip networks. Although specifically intended for the oblique wing, both of these features can also be applied to conventional configurations if needed.

The basic program takes a SHADE file, with its cross sections, and divides it into three networks, upper, lower, and wake. For the oblique wing case, if the tip option is selected, the input data is assumed to have an initial and a final cross section which are not singular points. This results in a gap between the upper and lower surfaces at the tips. The tip option constructs tip end-plate networks, needed to preserve the continuity of the vehicle's surface. Currently upper, lower, wake, and tip networks are the only ones

possible to generate with TACONV. These networks can be edited, or modifications to TACONV can be made, to further meet specific requirements in the geometry definition.

The SHADE file is required to consist of cross sections which all contain the same number of points. This is necessary to generate rectangular panels. TACONV assumes that an equal number of points are in the upper and lower networks. It therefore takes the total number of SHADE points in a cross sections and uses the first half for the upper surface and the second half for the lower surface. The middle point, corresponding to the trailing edge, is duplicated and used as an end point in both networks and also in the wake network. Since the SHADE cross section repeats the first point as last, the total number of points for the SHADE cross section must be odd so that TACONV can divide them equally into upper and lower sections.

The trailing wake is constructed by duplicating the trailing edge points and changing the x values to a user specified value which is greater than the maximum x dimension of the computational domain. If a case with yaw is used, then the wake can be rotated to correspond to the freestream direction.

There is much room for improvement in the TACONV program. The generality of the input geometry can be increased, as well as the ability to control the placement of networks. The code does, however, do a very good job at generating the basic TRANAIR network file for simple geometric cases. Any additions to it would depend upon the desired usage. If more complex geometries are required to be examined, TACONV can be modified to do it. If more interactive control is required, this can also be added. All of these changes, however, will add to the complexity of the program and could considerably slow down the process of converting basic SHADE files into TRANAIR networks.

CHAPTER 7

Conclusions

This study shows that the full potential TRANAIR code can provide good results for an all-body shaped hypersonic vehicle and is also capable of handling unconventional configurations like the oblique wing. The TRANAIR results agree fairly well with the RANS3D Navier-Stokes code results for the all-body, at low angles of attack in the transonic regime. At higher supersonic Mach numbers and at higher angles of attack where the flow becomes separated, TRANAIR does not have the ability to provide as accurate results as RANS3D. Nevertheless, RANS3D is a Navier-Stokes code and using it has some consequences. The RANS3D CPU times for the all-body are only somewhat higher than for TRANAIR. This is partly because RANS3D only uses one grid while TRANAIR cycles through a number of grids. It is the automated grid generation feature of TRANAIR that allows the overall turnaround time for a new configuration analysis to be much faster than with RANS3D.

The wave drag code provides some results which capture the general trend of the experimental data, but for the most part the results are unsatisfactory. The various wave drag analyses show that the code is not well suited for hypersonic configurations in the transonic regime. Using the wave drag code also requires using other methods to predict the wave drag due to lift, the induced drag, and the other aerodynamic forces and moments. The fast empirical methods generally have severe difficulty analyzing the highly nonlinear characteristics of transonic flow. These methods can provide usable results if crude

approximations are desired. As the design process expands, however, better techniques must be employed.

At the current level of computational capabilities, TRANAIR cannot be feasibly integrated into the conceptual design hypersonic vehicle synthesis code, HAVOC. The problem with TRANAIR is that the CPU times are much too high to allow a direct incorporation into a synthesis code. However, TRANAIR or something similar can be effectively used at specific junctures in the design process to provide a detailed analysis of the evolving design. By doing this, the limited transonic results of the design code can be periodically updated with TRANAIR results. This would require an interface between the TRANAIR force and moment results and the necessary HAVOC routines. TACONV is an initial step towards this interface. In its present form it can handle the basic conversion for the geometry input file.

There are no easy methods for quickly and accurately analyzing transonic flow behavior. Transonic performance can have a significant influence on the design of a hypersonic configuration. In the preliminary design phase the transonic analysis can be disregarded somewhat, but not completely. If it turns out to be a critical driver of the design, then early analyses can be very advantageous. Clearly a reliable transonic analysis method is needed somewhere in the preliminary design process.

REFERENCES

1. Gregory, Thomas J., Ardema, Mark, D., Waters, Mark, H., "Hypersonic Transport Preliminary Performance Estimates For an All-Body Configuration", AIAA Paper 70-1224, 1970.

2. Harris, Roy V., Jr., "An Analysis and Correlation of Aircraft Wave Drag", NASA TM X-947, 1964.

3. Boppe, Charles W., "Aerodynamic Analysis for Aircraft With Nacelles, Pylons, and Winglets at Transonic Speeds", NASA CR 4066, 1987.

4. Eminton, E. and Lord, W. T., "Note on the Numerical Evaluation of the Wave Drag of Smooth Slender Bodies Using Optimum Area Distributions for Minimum Wave Drag", Journal of the Royal Aeronautical Society, Vol. 60, No. 541, 1956.

5. Craidon, Charlotte B., "User's Guide for a Computer Program for Calculating the Zero-Lift Wave Drag of Complex Aircraft Configurations", NASA TM 86670, 1983.

6. Ashley, Holt and Landahl, Marten, Aerodynamics of Wings and Bodies, Dover Publications, Inc., New York, 1965.

7. Whitcomb, R. T., "A Study of Zero-Lift Drag-Rise Characteristics of Wing-Body Combinations Near the Speed of Sound", NACA Report 1273, 1954.

8. Jones, Robert T., "Theory of Wing-Body Drag at Supersonic Speeds", NACA Report 1284, 1956.

9. "TRANAIR Computer Code Theory Document", NASA Contract NAS2-12513, The Boeing Company, 1989.

10. "TRANAIR Computer Code User's Manual", NASA Contract NAS2-12513, The Boeing Company, 1989.

11. Carmichael, R. L. and Erickson, L. L., "PANAIR - A Higher Order Panel Method for Predicting Subsonic or Supersonic Linear Potential Flows About Arbitrary Configurations", AIAA Paper 81-1255, 1981.

12. Anderson, Dale A., Tannehill, John C. and Pletcher, Richard H., Computational Fluid Mechanics and Heat Transfer, Hemisphere Publishing Co., New York, 1984.

13. Bateman, H., "Irrotational Motion of a Compressible Inviscid Fluid", Proceedings of the National Academy of Sciences, Vol. 16, 1930, p. 816.

14. Saad, Youcef and Schultz, Martin H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems", SIAM J. Sci. Stat. Comput., Vol. 7, No. 3, 1986.

15. Madson, Mike, private communications with.

16. Cosentino, Gary, "Description of Input Parameters for RANS3D V-1.5", Unpublished, NASA Ames Research Center, 1991.

17 Pulliam, T. H. and Steger, J. L., "Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow", AIAA Journal, Vol. 18, 1980.

18. Baldwin, B. S. and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows", AIAA Paper 78-257, 1978.

19. Fujii, Kozo, Obayashi, Shigeru, "Computation of Three-Dimensional Viscous Transonic Flows with the LU Factored Scheme", AIAA Paper 85-1510, 1985.

20. Beam, R. and Warming, R. F., "An Implicit Finite Difference Algorithm for Hyperbolic Systems in Conservation Law Form", Journal of Computational Physics, Vol. 22, 1976.

21. Roe, P. L., "Generalized Formulation of TVD Lax-Wendroff Schemes", ICASE Report No. 84-53, 1984.

22. Goorjian, Peter, "A Streamwise Algorithm for the Euler and Navier-Stokes Equations Applied to Transonic Flows", NASA TM 101019, 1988.

23. Nelms, Walter P., Jr., "Aerodynamic Characteristics of an All-Body Hypersonic Aircraft Configuration at Mach Numbers from 0.65 to 10.6", NASA TN D-6577, 1971.

24. Cosentino, Gary, private communications with.

25. Walatka, Pamela, Buning, Pieter G., Pierce, Larry, Elson, Patricia A., "PLOT3D User's Manual", NASA TM 101067, 1990.

26. Hermstad, Dexter L., "RAID User's Guide", Sterling Federal Systems Inc., Project 2000-6, Technical Note No. 33, 1989.

27. Van Der Velden, Alexander J. M. and Kroo, Ilan,"The Aerodynamic Design of the Oblique Flying Wing Supersonic Transport", NASA CR 177552, 1990.

28. Smith, J. H. B., "Lift/Drag Ratios of Optimized Slewed Elliptic Wings at Supersonic Speeds", The Aeronautical Quarterly, Vol. 12, 1961.

29. Nelms, Walter P., Jr. and Axelson, John A., "Longitudinal Aerodynamic Characteristics of Three Representative Hypersonic Cruise Configurations at Mach Numbers From 0.65 to 10.70", NASA TM X-2113, 1970.

30. Bowcutt, K.G., Anderson, J.D. and Capriotti, D., "Viscous Optimized Hypersonic Waveriders", AIAA Paper 87-0272, 1987.

31. Ardema, Mark D., Chambers, Mark C., Terjesen, Eric J., Roberts, Cathy D., "Body Weight of Advanced Concept Hypersonic Aircraft", AIAA Paper 91-3180, 1991.

32. Barnes, Richard, "SHADE Software User's Guide and Maintenance Notes", Sterling Federal Systems Inc., Project 7104-321, Technical Note No. 13, 1987.

APPENDIX

Listing of TACONV Program

```
      PROGRAM TACONV
C -----------------------------------------------------------------------------------------------
C
C PROGRAM DESCRIPTION:
C     THIS PROGRAM CONVERTS A SHADE GEOMETRY FILE INTO A TRANAIR
C NETWORK FILE CONSISTING OF UPPER, LOWER, TIP (OPTIONAL) AND
C WAKE NETWORKS. THE WAKE SURFACE CAN ALSO BE ROTATED FOR
C CASES WITH SIGNIFICANT YAW SUCH AS OBLIQUE WINGS.
C
C INPUT FORMATS:
C     THE SHADE INPUT DATA CONSISTS OF CROSS SECTIONS. THE FIRST
C NUMBER READ IS THE NUMBER OF POINTS PER CROSS SECTION (NUM).
C NEXT THE CROSS SECTION IS READ. THE X,Y,Z FORMAT IS 3(F20.16). THE
C PROCESS IS REPEATED UNTIL THE END OF THE FILE IS REACHED OR A
C MARKER OF "-999" IS REACHED. THE INPUT FILE MUST HAVE AN EQUAL
C NUMBER OF UPPER AND LOWER POINTS, THEREFORE SINCE THE LAST
C POINT IS REPEATED THE TOTAL NUMBER OF POINTS (NUM) !!!MUST BE
C ODD!!! IT IS ALSO ASSUMED THAT THE DATA BEGINS ON AN UPPER
C SURFACE EDGE AND CONTINUES AROUND OVER THE LOWER SURFACE.
C THE MAXIMUM X VALUE OF THE WAKE IS ASKED AS INPUT. THIS IS THE
C VALUE THE CODE WILL GIVE THE TRAILING EDGE OF THE WAKE AND IT
C SHOULD BE GREATER THAN THE MAXIMUM X VALUE THAT WILL BE
C USED FOR THE COMPUTATIONAL BOX. THE CODE MUST KNOW WHERE TO
C ATTACH THE TRAILING WAKE. IF THE DATA CROSS SECTIONS ARE
C STREAMWISE CUTS, THE TRAILING EDGE OF THE BODY IS A ROW. IF THE
C CROSS SECTIONS ARE PERPENDICULAR TO THE FLOW, THE TRAILING
C EDGE OF THE BODY IS A COLUMN.
C     IF TIPS ARE USED THEY MUST FOLLOW A SPECIFIC FORM. THE CODE
C ASSUMES THAT THE Z VALUES OF THE FIRST AND LAST COLUMNS ON
C THE UPPER SURFACE DO NOT MATCH THE Z VALUES ON THE LOWER
C SURFACE. IT THEN WILL GENERATE TIP PLATE NETWORKS THAT
C CONNECT THE TWO SURFACES AND MAINTAIN CONTINUITY OF THE
C SURFACE.
C
C -----------------------------------------------------------------------------------------------
C
C VARIABLES:
```

```
C    ANGLE      YAW ANGLE WHICH WAKE IS ROTATED
C    LOUP       1 - IF TIP POINTS ARE FROM LOWER SURFACE
C               2 - IF TIP POINTS ARE FROM UPPER SURFACE
C    NN         NUMBER OF COLUMNS IN A TRANAIR NETWORK
C    NM         NUMBER OF ROWS IN A TRANAIR NETWORK
C    NUM        NUMBER OF POINTS PER X-SECTION IN THE INPUT FILE
C    NUNIT      THE FILE UNIT NUMBER TO READ FROM OR WRITE TO
C    VALUE      ARRAY WHICH INPUT DATA IS INITIALLY STORED
C    X          X VALUE OF POINT (LIKEWISE FOR Y AND Z)
C    XLL,XLR    X VALUES FOR TIP NETWORKS LL-LOWER LEFT(LAST COL)
C               AND LR-LOWER RIGHT(FIRST COL)
C    XMAX       MAXIMUM X VALUE OF WAKE NETWORK
C    XUL,XUR    X VALUES FOR TIP NETWORKS UL-UPPER LEFT(LAST COL)
C               AND UR-UPPER RIGHT(FIRST COL)
C
C-----------------------------------------------------------------------------------------------
C
C AUTHOR:
C    PAUL C. DAVIS
C    AERONAUTICAL ENGINEERING DEPARTMENT
C    CALIFORNIA POLYTECHNIC STATE UNIVERSITY, SAN LUIS OBISPO
C
C FOR:
C    NASA/AMES SYSTEMS ANALYSIS BRANCH (CODE FAS)
C    AND AS PART OF MASTERS THESIS
C
C DATE:
C    6/6/91
C
C-----------------------------------------------------------------------------------------------
C
      CHARACTER*80 INFILE,NETFILE
      CHARACTER*1 RESP,ROT,RORC
      REAL VALUE(6),XUR(90),XUL(90),XLR(90),XLL(90),YUR(90),YUL(90),
     + YLR(90),YLL(90),ZUR(90),ZUL(90),ZLR(90),ZLL(90),X(2),Y(2),Z(2)
C
      WRITE(6,*) 'ENTER NAME OF SHADE INPUT DATA FILE'
      READ(5,10) INFILE
      OPEN (UNIT=1,FILE=INFILE,STATUS='OLD')
C
      WRITE(6,*) 'ENTER NAME OF TRANAIR NETWORK OUTPUT FILE'
      READ(5,10) NETFILE
      OPEN (UNIT=4,FILE=NETFILE,STATUS='NEW')
C
      WRITE(6,*) 'ENTER THE MAXIMUM X VALUE DESIRED FOR THE
     +TRAILING WAKE'
      READ(5,'(F10.3)') XMAX
C
      WRITE(6,*) 'IS TRAILING EDGE A ROW (X-SECTIONS STREAMWISE) OR A
     + COLUMN (X-SECTIONS ORTHOGANAL TO FREESTREAM) (R/C)?'
      READ(5,'(A)') RORC
C
      WRITE(6,*) 'ROTATE WAKE (FOR CASES WITH YAW) (Y/N)?'
      READ(5,'(A)') ROT
```

```
C
      IF(ROT.EQ.'Y') THEN
      WRITE(6,*) 'ENTER THE ANGLE TO ROTATE WAKE (YAW ANGLE : XX.X)'
      READ(5,'(F6.2)') ANGLE
      ELSE
      ENDIF
C
      WRITE(6,*) 'GENERATE TIP END PLATES (Y/N)?'
      READ(5,'(A)') RESP
C
      OPEN (UNIT=2,FILE='TEMPLO',STATUS='NEW')
      OPEN (UNIT=3,FILE='TEMPUP',STATUS='NEW')
      OPEN (UNIT=5,FILE='WAKE',STATUS='NEW')
C
      NN=0
C
C READ IN THE NUMBER OF POINTS IN THE SHADE CROSS SECTION
 300 READ(1,20,END=200) NUM
C
C IF END OF INPUT FILE(-999) THEN STOP
      IF(NUM.EQ.-999) GOTO 200
C
C KEEP TRACK OF THE NUMBER OF CROSS SECTIONS OR COLUMNS, NN
      NN=NN+1
C
C ------------------------------------------------------------------------------------------------
C  IF NUM IS EVEN
C
      IF(MOD(NUM,2).EQ.0.0) THEN
      WRITE(6,*) 'CANNOT HAVE AN EVEN NUMBER OF POINTS IN A CROSS
     + SECTION'
      GOTO 999
C
C ------------------------------------------------------------------------------------------------
C  IF NUM IS ODD
C
      ELSE
        NL=(NUM-1)/2
C
C NM IS THE NUMBER OF ROWS IN A COLUMN
      NM=NL+1
C
      IF(MOD(NL,2).EQ.0.0) THEN
C ********************************************************************
C CALL SUBROUTINE RW(READS TWO POINTS) NL/2 TIMES FOR UPPER NET
C READ ONE MORE POINT AND STORE IN UPPER, WAKE, AND LOWER NETS
C READ NEXT POINT AND CALL RW NL/2-1 TIMES FOR LOWER NETWORK
C READ LAST POINT FOR LOWER NETWORK
C
C  STORE UPPER NETWORK
      DO 110 I=1,(NL/2)
        CALL RW(3,VALUE)
 110 CONTINUE
C
```

```
C  STORE LAST POINT OF UPPER NETWORK
      READ(1,30) VALUE(1),VALUE(2),VALUE(3)
      WRITE(3,50) (VALUE(J),J=1,3)
C
C  STORE LAST POINT ON UPPER NETWORK TO WAKE NETWORK
      IF(RORC.EQ.'R') THEN
        WRITE(5,50) (VALUE(J),J=1,3)
      ENDIF
C
C  STORE LOWER NETWORK (FIRST POINT SAME AS LAST FROM UPPER NET)
      READ(1,30) VALUE(4),VALUE(5),VALUE(6)
      WRITE(2,40) (VALUE(J),J=1,6)
      DO 120 I=2,(NL/2)
        CALL RW(2,VALUE)
  120 CONTINUE
C
C  STORE LAST POINT OF LOWER NETWORK
      READ(1,30) VALUE(1),VALUE(2),VALUE(3)
      WRITE(2,50) (VALUE(J),J=1,3)
C
      ELSE
C ***************************************************************************
C    CALL SUB RW(READS 2 POINTS) (NL-1)/2+1=NM/2 TIMES FOR UPPER NET
C    STORE LAST POINT READ INTO WAKE, AND LOWER NETWORKS ALSO
C    READ NEXT POINT AND STORE IN LOWER NETWORK
C    CALL RW (NL-1)/2 TIMES FOR LOWER NETWORK
C
C  STORE UPPER SURFACE NETWORK
      DO 130 I=1,((NL-1)/2+1)
        CALL RW(3,VALUE)
  130 CONTINUE
C
C  WRITE LAST POINT ON UPPER NETWORK TO WAKE NETWORK
      IF(RORC.EQ.'R') THEN
        WRITE(5,50) (VALUE(J),J=4,6)
      ENDIF
C
C  STORE LOWER NETWORK (FIRST POINT IS LAST FROM UPPER NETWORK)
      VALUE(1)=VALUE(4)
      VALUE(2)=VALUE(5)
      VALUE(3)=VALUE(6)
      READ(1,30) VALUE(4),VALUE(5),VALUE(6)
      WRITE(2,40) (VALUE(J),J=1,6)
      DO 225 I=1,((NL-1)/2)
        CALL RW(2,VALUE)
  225 CONTINUE
      ENDIF
      ENDIF
      GOTO 300
C
C ---------------------------------------------------------------------------------------------------
C  MERGE UPPER LOWER AND WAKE NETWORKS INTO ONE FILE
C
  200 CLOSE(2)
```

```
      CLOSE(3)
      CLOSE(5)
      OPEN(UNIT=2,FILE='TEMPLO',STATUS='OLD')
      OPEN(UNIT=3,FILE='TEMPUP',STATUS='OLD')
      OPEN(UNIT=5,FILE='WAKE',STATUS='OLD')
C
C  IF NM IS EVEN SET NZ = NM/2
C  IF NM IS ODD SET NZ = (NM-1)/2
      IF(MOD(NM,2).EQ.0.0) THEN
        NZ=NM/2
      ELSE
        NZ=(NM-1)/2
      ENDIF
C
C  READ UPPER NETWORK AND WRITE TO TRANAIR NETWORK FILE
      WRITE(4,60) NM,NN
      DO 450 I=1,NN
        NK=1
        DO 470 J=1,NZ
          CALL RWTEMP(3,X,Y,Z)
C
C  STORE UPPER VALUES OF TIP NETWORK
           IF(RESP.EQ.'Y'.AND.I.EQ.1) CALL TIP(X,Y,Z,XUR,YUR,ZUR,NK,2)
           IF(RESP.EQ.'Y'.AND.I.EQ.NN) CALL TIP(X,Y,Z,XUL,YUL,ZUL,NK,2)
          NK=NK+2
           IF(RORC.EQ.'C'.AND.I.EQ.NN) THEN
            WRITE(5,50) X,Y,Z
          ENDIF
 470    CONTINUE
        IF(MOD(NM,2).EQ.0.0) GOTO 450
C
C   IF NM IS ODD READ LAST POINT
          READ(3,40) X(1),Y(1),Z(1)
          WRITE(4,40) X(1),Y(1),Z(1)
           IF(RORC.EQ.'C'.AND.I.EQ.NN) THEN
            WRITE(5,50) X,Y,Z
          ENDIF
           IF(RESP.EQ.'Y'.AND.I.EQ.1) THEN
            XUR(NM)=X(1)
            YUR(NM)=Y(1)
            ZUR(NM)=Z(1)
          ELSE
           IF(RESP.EQ.'Y'.AND.I.EQ.NN) THEN
            XUL(NM)=X(1)
            YUL(NM)=Y(1)
            ZUL(NM)=Z(1)
          ENDIF
        ENDIF
 450 CONTINUE
C
C  READ LOWER NET FROM TEMP FILE; WRITE TO TRANAIR NETWORK FILE
      WRITE(4,70) NM,NN
      DO 480 I=1,NN
        NK=NM
```

```
      DO 490 J=1,NZ
        CALL RWTEMP(2,X,Y,Z)
C
C STORE LOWER VALUES OF TIP NETWORK
          IF(RESP.EQ.'Y'.AND.I.EQ.1) CALL TIP(X,Y,Z,XLR,YLR,ZLR,NK,1)
          IF(RESP.EQ.'Y'.AND.I.EQ.NN) CALL TIP(X,Y,Z,XLL,YLL,ZLL,NK,1)
           NK=NK-2
 490   CONTINUE
       IF(MOD(NM,2).EQ.0.0) GOTO 480
C
C    IF NM IS ODD READ LAST POINT
          READ(2,40) X(1),Y(1),Z(1)
          WRITE(4,40) X(1),Y(1),Z(1)
           IF(RESP.EQ.'Y'.AND.I.EQ.1) THEN
          XLR(1)=X(1)
          YLR(1)=Y(1)
          ZLR(1)=Z(1)
        ELSE
           IF(RESP.EQ.'Y'.AND.I.EQ.NN) THEN
            XLL(1)=X(1)
            YLL(1)=Y(1)
            ZLL(1)=Z(1)
          ENDIF
        ENDIF
 480 CONTINUE
C
      IF(RESP.EQ.'Y') THEN
        CALL TIPNET(NM,XUR,YUR,ZUR,XLR,YLR,ZLR,XLL,YLL,ZLL,XUL,
     +  YUL,ZUL)
      ENDIF
        CALL WAKENET(NN,ROT,XMAX,ANGLE)
C
  10  FORMAT(A)
  20  FORMAT(I7)
  30  FORMAT(3(F20.16))
  40  FORMAT(6(F10.6))
  50  FORMAT(3(F10.6))
  60  FORMAT('$POINTS - UPPER SURFACE',/,'1.',/,'1.',/,I9,'.',I9,'.',50X,
   + 'UPSURF')
  70  FORMAT('$POINTS - LOWER SURFACE',/,'1.',/,'1.',/,I9,'.',I9,'.',50X,
   + 'LOSURF')
 900  WRITE(4,'(A)') '$END'
      CLOSE(4)
      CLOSE(2,STATUS='DELETE')
      CLOSE(3,STATUS='DELETE')
      CLOSE(5,STATUS='DELETE')
 999  STOP
      END
C
C ---------------------------------------------------------------------------------------------------
C
      SUBROUTINE RW(NUNIT,VALUE)
C
C SUBROUTINE READS X,Y,Z VALUES OF 2 SHADE POINTS FORMAT 3(F20.16)
```

```
C  AND WRITES THEM ON ONE LINE IN TRANAIR FORMAT 6(F10.6)
C
      DIMENSION VALUE(6)
      READ(1,10) VALUE(1),VALUE(2),VALUE(3)
      READ(1,10) VALUE(4),VALUE(5),VALUE(6)
      WRITE(NUNIT,20) (VALUE(J),J=1,6)
   10 FORMAT(3(F20.16))
   20 FORMAT(6(1X,F9.6))
      RETURN
      END
C
C ------------------------------------------------------------------------------------------------
C
      SUBROUTINE RWTEMP(NUNIT,X,Y,Z)
C
C  SUBROUTINE READS IN X,Y,Z VALUES FROM A TEMP NETWORK FILE
C  AND WRITES THEM TO THE TRANAIR NETWORK FILE
C
      DIMENSION X(2),Y(2),Z(2)
      READ(NUNIT,10) X(1),Y(1),Z(1),X(2),Y(2),Z(2)
      WRITE(4,10) X(1),Y(1),Z(1),X(2),Y(2),Z(2)
   10 FORMAT(6(F10.6))
      RETURN
      END
C
C ------------------------------------------------------------------------------------------------
C
      SUBROUTINE TIP(X,Y,Z,XTIP,YTIP,ZTIP,NK,LOUP)
C
C  SUBROUTINE WRITES TIP VALUES INTO TIP NETWORK
C
      DIMENSION X(2),Y(2),Z(2),XTIP(90),YTIP(90),ZTIP(90)
      INTEGER NK,LOUP
C
C  LOUP DETERMINES IF THE TIP POINTS ARE FROM THE LOWER OR UPPER
C  SURFACES 1 - LOWER SURFACE, 2 - UPPER SURFACE
C
      IF(LOUP.EQ.1) THEN
        DO 100 J=1,2
          XTIP(NK+1-J)=X(J)
          YTIP(NK+1-J)=Y(J)
          ZTIP(NK+1-J)=Z(J)
  100   CONTINUE
      ELSE
        DO 200 J=1,2
          XTIP(NK-1+J)=X(J)
          YTIP(NK-1+J)=Y(J)
          ZTIP(NK-1+J)=Z(J)
  200   CONTINUE
      ENDIF
      RETURN
      END
C
C ------------------------------------------------------------------------------------------------
```

```
C
      SUBROUTINE TIPNET(NM,XUR,YUR;ZUR,XLR,YLR,ZLR,XLL,YLL,ZLL,
     +XUL,YUL,ZUL)
C
C  SUBROUTINES WRITES OUT TIP NETWORKS TO TRANAIR NETWORK FILE
C
      DIMENSION XUR(90),YUR(90),ZUR(90),XLR(90),YLR(90),ZLR(90),XLL(90),
     +YLL(90),ZLL(90),XUL(90),YUL(90),ZUL(90)
C
C  WRITE RIGHT TIP
      WRITE(4,10) NM
      DO 100 J=1,NM
        WRITE(4,40) XUR(J),YUR(J),ZUR(J),XLR(J),YLR(J),ZLR(J)
  100 CONTINUE
C
C  WRITE LEFT TIP
      WRITE(4,20) NM
      DO 110 J=1,NM
        WRITE(4,40) XLL(J),YLL(J),ZLL(J),XUL(J),YUL(J),ZUL(J)
  110 CONTINUE
C
   10 FORMAT('$POINTS - RIGHT TIP',/,'1.',/,'5.',/,8X,'2.',I9,'.',50X,'RTIP')
   20 FORMAT('$POINTS - LEFT TIP',/,'1.',/,'5.',/,8X,'2.',I9,'.',50X,'LTIP')
   40 FORMAT(6(F10.6))
      RETURN
      END
C
C-------------------------------------------------------------------------------------------------------
C
      SUBROUTINE WAKENET(NN,ROT,XMAX,ANGLE)
C
C  SUBROUTINE WRITES WAKE TO TRANAIR NETWORK FILE
C
      REAL ANGLE,XMAX
      CHARACTER*1 ROT
      INTEGER NN
      PI=3.14159265359
C
      WRITE(4,10) NN
  100 READ(5,20,END=900) XEND,YEND,ZEND
      IF(ROT.EQ.'Y') THEN
        YWAKE=-TAN(ANGLE*PI/180)*(XMAX-XEND)+YEND
      ELSE
        YWAKE=YEND
      ENDIF
      WRITE(4,30) XEND,YEND,ZEND,XMAX,YWAKE,ZEND
      GOTO 100
   10 FORMAT('$POINTS - WAKE',/,'1.',/,'18.',/,8X,'2.',I9,'.',50X,'WAKE')
   20 FORMAT(3F10.6)
   30 FORMAT(6(F10.6))
  900 RETURN
      END
```