NASA CR-176,934

NASA-CR-176934
19860019466

A Reproduced Copy

OF

NASA CR-176,934

Reproduced for NASA

by the

NASA Scientific and Technical Information Facility

DAA/Ames NOC2-150

JOINT INSTITUTE FOR AERONAUTICS AND ACOUSTICS

National Aeronautics and Space Administration
Ames Research Center

JIAA TR - 65

Stanford University

A ZONAL COMPUTATIONAL PROCEDURE ADAPTED TO THE OPTIMIZATION OF TWO-DIMENSIONAL THRUST AUGMENTOR INLETS

BY

T. S. Lund, D. A. Tavella and L. Roberts

Stanford University
Department of Aeronautics and Astronautics
Stanford, CA 94305

JULY 1985

JIAA-TR-65

A ZONAL COMPUTATIONAL PROCEDURE ADAPTED TO THE OPTIMIZATION OF TWO-DIMENSIONAL THRUST AUGMENTOR INLETS

Thomas s. Lund, Domingo A. Tavella, and Leonard Roberts

STANFORD UNIVERSITY
Department of Aeronautics and Astronautics
Stanford, California 94305

JULY 1985

ABSTRACT

A viscous-inviscid interaction methodology based on a zonal description of the flowfield is developed as a means of predicting the performance of two-dimensional thrust augmenting ejectors. An inviscid zone comprising the irrotational flow about the device is patched together with a viscous zone containing the turbulent mixing flow. The inviscid region is computed by a higher order panel method, while an integral method is used for the description of the viscous part. A non-linear, constrained optimization study is undertaken for the design of the inlet region. In this study, the viscous-inviscid analysis is complemented with a boundary layer calculation to account for flow separation from the walls of the inlet region. The thrust-based Reynolds number as well as the free stream velocity are shown to be important parameters in the design of a thrust augmentor inlet.

ACKNOWLEDGEMENTS

This work was supported by NASA grant NCC 2-150, and was monitored by Mr. D. Koenig of the Large-Scale Aerodynamics branch, AMES Research Center.

TABLE OF CONTENTS

NOMENCLATURE

[A], [$\bar{A}$]	coupling coefficient matricies for reduced equations
{B}, {C}	right hand sides of reduced equations
b	characteristic width of turbulent region
C_i	weighting coefficients in the penalty function transformation
C_p	pressure coefficient
d	shroud thickness
g	objective function after the penalty function transformation
H	shroud inner half-width
k	eddy viscosity scaling constant
L	augmentor shroud length
L_j	horizontal length of the viscous-inviscid interaction zone
p	pressure
R	ramp function
R_n	shroud nose radius
R_T	Reynolds number based on jet characteristic velocity and shroud inner width
r, t	constants in Eq. (2.14)
T	2-D gross thrust
T_o	2-D primary thrust
u,v	x,y velocity components
u_o	velocity at outer edge of jet velocity profile

u_1	maximum excess velocity in viscous zone
u_∞	free stream velocity
u_c	jet characteristic velocity
$\hat{u}$	approximate viscous solution
V_N	velocity component normal to the jet boundary
x, y	coordinates in the viscous solution
x_{end}	final station at which the viscous and inviscid solutions are matched
X, Y	coordinates for the shroud description
X_o	jet nozzle location
X_L	length of inlet lip
α	ln(2)
γ	ratio of free stream velocity to jet characteristic velocity
Γ	x-momentum conservation operator
δ_i	penalty functions
θ	lip rotation angle
μ	molecular viscosity
ν_t	eddy viscosity
ξ	dummy variable of integration
ρ	fluid density
τ	Reynolds stress in 2-D boundary layer approximation
ϕ	thrust augmentation ratio
$\bar{\phi}$	velocity potential
ω	relaxation parameter

$( )_{inv}$	quantity computed from the inviscid solution
$( )^{n}$	iteration level n
$( )_{vis}$	quantity computed from the viscous solution
$(\cdot)$	denotes differentiation with respect to x

LIST OF FIGURES

1. INTRODUCTION

A thrust augmentor consists of a high momentum primary jet which is exhausted into the confines of an aerodynamic shroud. As the jet evolves, it undergoes turbulent mixing with the surrounding stream, and as a result induces an entrained flowfield about the device. Augmentation in thrust is realized through the combined effects of the jet being discharged into a region of lowered pressure, and as a result of the induced pressure distribution on the surface of the shroud.

One important application of the thrust augmentor is found in vertical takeoff aircraft in which the weight of the aircraft exceeds the thrust produced by the jet engines in a standard configuration. The thrust is boosted to a level sufficient to overcome the weight of the aircraft through the use of a pair of thrust augmenting ejectors mounted along the fuselage at the wing roots. Figure 1 shows a cross-section of such a vertical takeoff aircraft. Study of this configuration is the primary motive of the present work, but the methodology developed here is general and may be used to study other VSTOL configurations.

In the design of VSTOL aircraft to be fitted with thrust augmentors, a means of evaluating the performance of various ejector configurations is needed in order to realize the optimal benefit. Considerable work has been undertaken in recent years in order to develop theories aimed at predicting thrust augmentor

performance. While much progress has been made in this quest, as of the present time no methods exist which are efficient or robust enough to be used in detailed parametric or optimization studies. The goal of this work is to develop a robust model suitable to be used by the engineer as a design tool.

The earliest attempts at modeling the thrust augmentor[1] were based on control volume approaches which rely on a uniform flow assumption for the inviscid portion of the flowfield and satisfaction of the equations of motion only in a global sense. Such theories possess the advantages of utmost simplicity, yielding closed form analytic solutions, but at the same time they suffer from the fact that the details of the flowfield in the near vicinity of the shroud are not resolved. With their lack of ability to provide surface velocity or pressure information, the global formulations are not able to predict the outcome of perturbations to the shroud geometry. Furthermore, the uniform flow assumption for the inviscid flow entering the shroud can be considered dubious for cases in which the nozzle is located ahead of the entrance of the shroud since experiments have shown that this assumption is by no means valid[10].

At the opposite extremes of both robustness and computational simplicity lies a solution to the problem through the use of a numerical analysis of the full Navier-Stokes equations. A detailed simulation such as this would provide with good assurance all the necessary information needed to evaluate the performance of any arbitrary geometrical configuration. However, the

time required to evaluate one flowfield in this fashion is on the order of ten hours on the CDC 7600[2]. Optimization studies typically require hundreds of flowfield evaluations. Thus one can expect on the order of a thousand CDC 7600 hours of computational time if a full Navier-Stokes simulation is used to optimize an augmentor design.

In light of the individual shortcomings of the two afore mentioned methods, much of the current effort in thrust augmentor modeling has focused on a methodology which retains much of the detailed information provided by a Navier-Stokes solution, while at the same time requiring only a modest computational effort. Fore-runners of such methodologies are viscous-inviscid interaction algorithms.

In the viscous-inviscid approach, the flowfield is subdivided into separate regions or "zones" in which the character of the flow is distinctly different. Typically an inviscid zone is established which is postulated to be free of shear. As a counterpart, a viscous zone is established in which shear and rotational effects are expected to play an important role. Within each zone the simplest justifiable approximations to the equations of motion are made. Each zone may then be solved quasi-independently with coupling information appearing through the common boundary conditions existing at the interface between them.

Bevilaqua et al.[3,4] developed a viscous-inviscid algorithm for the performance prediction of two-dimensional thrust augmentors. Bevilaqua's code makes use of a combined panel/vortex lattice

method to compute the inviscid flow about the shroud, while the turbulent zone is computed using a finite difference solution to a parabolic set of equations. This method has been used to predict the performance of two-dimensional thrust augmentors of low aspect ratio in which the turbulent jet has not yet expanded to the point of the channel walls by the time of exit from the shroud.

Tavella and Roberts[5] have studied the limit of large aspect ratio thrust augmentors in which the jet has encountered the walls by the time of exit. In their algorithm, the inviscid solution was obtained using the technique of conformal mapping, and the viscous turbulent zone was computed using an integral method. This algorithm proved to be extremely economical and several parametric studies were performed. Aside from the attractiveness of its efficiency, the methodology lacked robustness with the conformal mapping only admitting shroud geometries which could be described by small perturbations to flat plates.

In the present work, the integral formulation of Tavella's model is retained, while the inviscid solution is generated using a higher order panel method[6]. Use of the panel method for the inviscid solution removes the limitations of slightly perturbed shroud geometries imposed by the conformal mapping technique of Tavella's model. The model is still restricted to augmentors of high aspect ratio, but can be extended to study the low aspect ratio regime as well. With most practical augmentor designs falling within the capabilities of the model, this restric-

tion on aspect ratio is not a serious handicap.

Unlike all other models contained in the literature to date which are limited to a quiescent far field, the present one can treat thrust augmentors which are immersed in a moving free stream. The advantage of this feature is that it is now possible to explore the differences which occur in the thrust augmentor performance as the aircraft accelerates toward flight speed.

Another feature of the present model is that a boundary layer calculation is performed on the surface of the shroud, thereby providing a means of predicting flow separation over the inlet region of the device. This detail is important because boundary layers subject to the adverse pressure gradient inherently present within the entrance region of the thrust augmentor are prone to separation. Separated boundary layers in the inlet region of the thrust augmentor imply a significant loss of stagnation pressure, and hence a reduction in performance. With this problem understood, and properly modeled, configurations may be designed which do not suffer from inlet stall.

2. ZONAL DESCRIPTION AND VISCOUS-INVISCID INTERACTION

2.1 DESCRIPTION OF THE PHYSICAL PROBLEM

A typical thrust augmentor has the property that the flow field contains well defined viscous regions imbedded within a largely inviscid field. The viscous effects introduced by the jet are restricted to a finite zone near its axis. The turbulent zone produced by the jet serves to mix the high energy fluid within the jet with the low energy fluid in the inviscid region. This mixing and its associated entrainment provides the mechanism of thrust augmentation.

A second region where viscous effects may be identified is in the boundary layers which develop on the surface of the shroud. Momentum transfer takes place in the boundary layer as manifested by skin friction and a no-slip velocity condition at the shroud surface. Evolution of the boundary layer also determines whether or not the flow will separate from the walls of the shroud in response to the adverse pressure gradients always present within the duct of the thrust augmentor.

Both the viscous zones containing the jet and boundary layer share the property that the streamwise velocity gradients are small when compared with the gradients normal to the flow direction. This fact has long been known for boundary layers[17], and that for jets more recently[18]. In these two cases it is possible to neglect the streamwise velocity diffusion terms

in the Navier-Stokes equations, thereby reducing these elliptic equations to a parabolic set. Solutions to parabolic equations are more tractable in general than elliptic ones, and lend themselves to a variety of efficient numerical marching schemes.

While the viscous regions are small and well contained within the inviscid region, there exists a surprising degree of interaction between these two zones. This interaction is better understood by dividing the flowfield into four regions as shown in Figure 2.

Region 1 becomes important only for configurations in which the nozzle is located ahead of the shroud. When this region is present, the jet develops under the influence of the surrounding entrained inviscid flow. This entrained or secondary flow will depend on the shape of the shroud, suction into the inlet, and entrainment distribution of the primary jet. As the growth characteristics of the jet are intimately related to the secondary flow, it may be expected that the viscous-inviscid interaction will be quite strong in this region.

Region 2 is similar to region 1 with the exception that the jet and its entrained inviscid flow are bounded by the channel walls. The viscous-inviscid interaction still takes place as in region 1, but becomes less intense with increasing distance from the inlet. The inviscid flow becomes approximately uniform as the downstream end of the region is reached. Boundary layers are developing on the shroud walls and merge with the inviscid flow.

Region 3 is characterized by a turbulent flow which completely fills the channel, thus terminating the viscous-inviscid interaction. The pressure increases with downstream distance as a result of momentum dissipation. At the end of region 3 the pressure has become equal to the atmospheric value. The boundary layers on the channel walls continue to grow, but now merge with the turbulent instead of inviscid flow.

Region 4 contains the wake. In this region the exhausting mixed flow acts like a free jet issuing from the shroud exit. Further mixing with the inviscid flow takes place downstream of the shroud, but because the primary momentum has now been greatly diffused, the viscous-inviscid interaction is much weaker here as compared with the region near the jet nozzle. For this reason the wake region is treated as a non-mixing slipstream which extends infinitely far downstream of the ejector. More information on the wake modeling will be given in a subsequent section.

2.2 MATHEMATICAL MODEL

The flowfield of a generic two-dimensional thrust augmentor is divided into a viscous and an inviscid zone as illustrated in Figure 3. The viscous zone originates at the jet nozzle and increases in width at a linear rate with streamwise distance. In order to avoid a sharp corner in the inviscid region, the linear growth of the viscous zone is halted when the jet is a short distance from the channel walls. At this point a semicircle

is used to bridge the gap between the viscous zone and the channel wall (the necessity of this alteration will be illuminated in the forthcoming section which describes the inviscid solution). The inviscid zone encloses much of the shroud and extends far away from the shroud in all directions. The common boundary between the viscous and inviscid region is represented physically by the region where the shear due to the turbulent mixing has become negligible.

The choice for a linearly growing turbulent region downstream of the primary nozzle merits some discussion. For the case in which the nozzle is located well ahead of the shroud, a dimensional analysis provides the justification for a wedge shaped turbulent region, as it is easily shown that for a two-dimensional free jet the characteristic width scale must grow linearly with the distance from the virtual origin. Experiments confirm this scaling law and provide a wedge half angle of 12 degrees for which the shear has diminished to a negligible value[13]. For the case in which the nozzle is located within the shroud, it is harder to justify the wedge shaped viscous zone since the addition of the length scale introduced by the channel width makes it unclear that the growth rate should be a linear function of the streamwise distance. Indeed if the channel extends infinitely far downstream, is expected that all length scales will asymptotically approach some constant fraction of the channel height. However, when the width of the jet is small compared with the channel width, the walls may be thought to be located at infinity

in terms of a local coordinate system assigned to the jet. It is therefore expected that the free jet result for a linear growth rate will hold near the virtual origin, and a slowing of the growth rate will occur as the jet width becomes of the same order as the channel width. Thus if the wedge concept is used for the confined jet, it may be expected that the width of the turbulent zone will be over predicted as the distance from the virtual origin is increased. This fact poses no serious difficulty within the present framework, since the model for the viscous region recovers the inviscid solution at large distances from the centerline of the channel. Thus no harm is done if the viscous calculation is used to predict a small portion of the inviscid flow at large distances from the channel centerline.

The performance of the device is assessed in terms of the augmentation ratio defined as

$$\phi = \frac{T}{T_o}. \qquad (2.1)$$

The augmentation ratio is computed in either of two ways; direct integration of the surface pressure, or using the Blasius theorem for a control volume surrounding the device. The two methods agree within two percent in all cases. All results presented here are based on an integration of the surface pressure.

2.3 INVISCID PROBLEM AND ITS SOLUTION

The inviscid flowfield is assumed to be irrotational and incompressible, and hence a potential flow formulation is appropriate. Within the framework of the potential formulation, the velocity field is derivable from the scalar velocity potential function as follows

$$\vec{V} = \nabla\Phi. \tag{2.2}$$

Substitution of the above form for the velocity into the incompressible continuity relation leads to the result that the velocity potential satisfies Laplace's equation

$$\nabla^2\Phi = 0. \tag{2.3}$$

An integral of the momentum equation gives the familiar Bernoulli equation which relates the pressure to the velocity potential

$$\frac{p}{\rho} + \frac{1}{2}\nabla\Phi\cdot\nabla\Phi = \text{const.} \tag{2.4}$$

Several approximate methods exist for solving Laplace's equation on an arbitrary domain. As a subset of these, both surface singularity and finite difference schemes will be considered in more detail below. Before discussing various solution schemes, a description of the geometry and boundary conditions which

define the solution domain is provided.

Exploiting the assumed symmetry of the shroud geometries, it is possible to focus attention on only the upper half plane. Abstracting from Figure 3, the geometry and boundary conditions defining the inviscid problem is defined as shown in Figure 4. The solid boundary extending in front of the shroud represents the dividing streamline which approaches the device. The following linear segment represents the jet boundary with permeable boundary conditions included to account for the jet entrainment. The half circle at the upper end of the jet boundary serves as a control station with a uniform flow transpiration boundary condition imposed there. The need for the control station arises from the fact that panel methods become inaccurate in a concave corner region[6,7]. Inserting a smooth curve such as a half circle greatly reduces this inherent difficulty. The uniform flow condition imposed at the control station is justifiable since experiments have shown that the inviscid flow becomes nearly uniform after about one half channel width into the shroud.
The augmentor shroud is modeled as an impermeable boundary. For the wake, the assumption is made that the slipstream surface which exists between the exhausting jet and the inviscid flow at the exit of the device is a continuation of the same streamline which defines the body shape. Physically this assumption is equivalent to ignoring the fluid shear which exists at the slipstream interface. This is found to be justifiable, however, since computations have shown that modeling of the mixing taking place

downstream of the shroud has a negligible effect on global quantities such as the augmentation ratio.

Several methods are investigated for the computation of the inviscid region. Surface singularity schemes such as a panel methods are highly desirable because of their ease of implementation and computational efficiency. However, the need to compute the flow in an internal region of the thrust augmentor inlet requires special treatment. A finite difference calculation requires no special treatment in an internal region, but requires a grid generation and is not nearly as efficient as a panel method. A hybrid scheme composed of both surface singularity and finite difference calculations has certain advantages. These three particular methods are analyzed in more detail below.

2.3.1 PANEL METHOD DESCRIPTION

Panel methods belong to a general class of surface singularity methods in which a solid boundary is replaced with various forms of singular elementary solutions (sources, sinks, doublets, vortices, etc.). In panel methods the surface of a solid body immersed in the field is decomposed into many small surface elements or "panels" over which sources are distributed. The intensity of each of these source panels is determined by requiring that when the influence of all panels as well as the free stream are considered, specified boundary conditions be satisfied at a discrete number of collocation points. In most non-lifting problems which arise in aerodynamics, the flow tangency condition

imposed at the geometric center of each panel provides the necessary boundary conditions to close the system. The solution process involves a system of simultaneous linear equations for the source strength on each panel. The coupling matrix in this system contains the so called aerodynamic influence coefficients which describe the net induced velocity at the collocation point of one panel as a result of the presence of another panel. With the order of the system equal to the number of surface elements, (on the order of 100 surface elements are needed) solution of this system is easily achieved in a direct fashion using Gaussian elimination.

Various forms of panel methods arise from different approximations to the surface shape as well as the source distribution. In the classical form of the panel method, the surface is described by linear elements formed by joining adjacent discrete points which lie on the body. In this formulation, the source strength takes on a constant value over each panel. In the classical method, the aerodynamic influence coefficients are easy to derive and may be rapidly computed numerically.

So called higher order schemes result when better approximations are made to the source distribution and/or the surface shape. In practical higher order methods, the surface curvature is accounted for by fitting parabolas to the surface contained within two points on the body, and the source distribution is described by linear or quadratic sections. These more sophisticated methods result in a more tedious derivation for the aerodynamic

influence coefficients as well as increased effort in numerical computation. However, as shall be shown, they lead to increased accuracy.

2.3.2 CLASSICAL PANEL METHOD

Use of a classical panel method[7] in which the singularity strength is constant over each linear surface element is first investigated. Figure 5 shows that this method provides an accurate description of the inviscid flow field in the regions external to the shroud, but produces poor results in the internal region between the two lobes of the shroud. It is observed that mass is not conserved in the internal region, and that a discontinuity in both velocity magnitude and direction is present at the control station. Increasing the number of surface elements in the internal region weakens the velocity discontinuity, but satisfactory results can not be obtained even with as many as 100 panels in the internal region alone. The failure of the first order panel method may be attributed to "leakage" effects caused by the jump in singularity intensity between adjacent panels. Because of a symmetry property present in the error caused by the discontinuous source distribution, errors cancel fortuitously in external regions and compound in internal regions[6]. Evidently as a result of this compounding of the error, leakage persists even as the surface element size is dramatically reduced.

It is concluded that the first order panel method alone will not provide sufficient accuracy for the purposes of this

study. However the fact that the first order panel method is capable of efficiently producing accurate results for the external region of the inviscid flow field leads to the development of a novel hybrid scheme in which the external flow is computed with the first order panel method, and the internal portion of the inviscid flow is computed using a finite difference technique. This method is discussed in the following section.

2.3.2 CLASSICAL PANEL METHOD WITH A "CFD PATCH"

The basic idea of the hybrid scheme is to first compute the entire inviscid flowfield using the first order panel method. The internal region is then refined by inserting a computational mesh and performing a finite difference calculation. Figure 6 shows the idea more clearly.

The panel solution provides values of the velocity potential to be used as a Dirichlet boundary condition for the finite difference calculation at the inflow boundaries of the grid. Neumann boundary conditions for the normal derivative of the velocity potential (equivalent to the velocity component normal to the boundary) are specified at all other boundaries.

The grid is generated using a simple algebraic scheme which results in a non-orthogonal grid. A coordinate transformation is used to map the physical domain into an indicial computational space. The Laplacian operator is not invariant under this tansformation and consequently takes on a more complicated form[14]. The solution to the problem is carried out in the computational

domain and the result mapped back into the physical plane.

The finite difference approximation results in a large order system of simultaneous linear equations which are beyond the capabilities of direct solver routines[14]. In this case a successive over relaxation scheme (SOR) is used to obtain an approximate solution to the finite difference equations. The Neumann boundary conditions are incorporated in the relaxation process as suggested by Steger[15].

As displayed in Figure 7, it is found that this hybrid scheme produces an inviscid flow field with good accuracy both in the internal and external regions. The solution blends smoothly between the panel and finite difference regions and mass is conserved in the internal region. While the method provides the desired accuracy, the numerical efficiency is greatly compromised with the finite difference calculation taking roughly an order of magnitude more computational time than is required for the panel solution alone. On the other hand, the hybrid scheme is expected to be an order of magnitude faster than a finite difference calculation used to model the entire inviscid field.

As will be discussed in the next section, for the cases treated here consisting of irrotational incompressible flow, a more sophisticated panel method formulation may be used to obtain results with accuracy comparable to the hybrid scheme. However for cases in which compressible and especially rotational effects play a role in the secondary flow, the hybrid scheme developed here would be of greater advantage. As an example,

a transonic full potential formulation could be used to model the external region with an Euler formulation used in the internal region. The merits of combining transonic full potential and Euler equation calculations for computations involving airfoils has already been shown[15].

2.3.3 HIGHER ORDER PANEL METHOD

The problem of "leakage" inherent to a first order panel method in an internal region may be remedied through a slightly different formulation. The jump in singularity strength between two adjacent panels responsible for the leakage is removed by allowing the singularity distribution to be described by higher order curves[6]. Use of a linear variation in the singularity intensity over each surface element produces an overall singularity distribution which is continuous. Further improvement in accuracy for a given number of panels is achieved by using quadratic sections to describe the singularity intensity variation over each panel. In this case it is necessary for mathematical consistency to describe the surface geometry with second order curves as well[6].

In this work the formulation given by Hess[6] which uses quadratic surface elements and allows for quadratic variation in singularity intensity is followed (see Appendix A for the details of the method). This higher order formulation involves more complicated calculations in order to establish the aerodynamic influence coefficients. In addition, the need for the local

curvature of the surface geometry requires a second order interpolation scheme. Despite these increases in complexity, the higher order formulation still produces a system of simultaneous linear equations of order equal to the number of panels. Thus the solution can easily be obtained with a direct solver as in the case of the classical panel method.

Shown in Figure 8 are the results of the higher order formulation, which produces an inviscid flow field which is accurate in the internal regions as well as the external regions. Mass is conserved in the internal region and the discontinuity in velocity magnitude and direction at the control station is less than five percent of the velocity there. The time required to compute the higher order solution is roughly one and a half times that needed to compute the classical panel solution.

It is found that this panel method formulation is highly sensitive to discontinuities in the curvature of the body surface, a problem which is absent in a classical panel method formulation due to neglect of the surface curvature. Clustering panels in regions of inherent curvature discontinuities as well as using cubic spline fits to describe the surface whenever possible is necessary to achieve a smooth velocity distribution on the surface of the shroud. Figure 9 shows a typical panel distribution required to produce a good velocity distribution. Out of three inviscid methodologies investigated here, the higher order formulation provides the most economical route to an accurate solution, and thus has been chosen for the inviscid flow method of solution.

2.4 VISCOUS SOLUTION

As mentioned above, the governing equations for the jet flow may be approximated by a parabolic set identical to the familiar boundary layer equations. One class of solutions to these equations is comprised of finite difference schemes for which straightforeward numerical techniques are well developed[19]. While finite difference formulations have associated with them large array storage requirements and time-consuming iteration processes, they have the advantage that almost all schemes approach the exact solution as the grid becomes infinitely fine.

An alternate approximate form of solution to the boundary layer equations exists which does not become exact in any practical limit, but which still provides an accurate solution with good numerical efficiency. This form of solution is known as the integral method as first developed by Von Karman[20] and Pohlhausen[21].

In the integral formulation, the streamwise momentum equation is converted from a statement of local force balance to one of average force balance by first integrating in a direction normal to the flow. The resulting integro-differential equation is then further simplified by assuming a functional form of the solution which has a number of free parameters and satisfies the given boundary conditions. Depending on the number of free parameters, the system is closed by requiring that the error made by the assumed solution form be orthogonal to a set of

independent functions (an idea borrowed from the method of Galerkin). The end result is a set of coupled ordinary differential equations which may be readily solved by marching downstream from initial data.

From a numerical standpoint, the integral method is ideal since it poses no large storage requirement, and better still requires no iteration. Success of the integral method has been demonstrated for boundary layers[22] as well as for free and confined jets[23]. The acknowledged approximate nature of the integral method is of no great concern when computing a turbulent flow as done here since uncertainties in the turbulent stress model are expected to obscure any increases in accuracy gained in a fine mesh finite difference calculation.

A thorough study by Tavella and Roberts[8] provides justification for the use of the integral method for the thrust augmentor problem. In their report, regressions to experimental data are given which validate the use of selected approximate solution forms. Solution of the problem using the integral method are also shown to yield good correlation with experiment for the jet velocity profile and pressure evolution within the mixing channel. The calculations prove to be rapid indeed, requiring only a fraction of a second of CPU time on the IBM 3081 machine.

The demonstrated accuracy and efficiency of the integral method provide the motivation to include it in the model as the method of solution for the viscous zone. The particulars of the method used here are well described elsewhere[5,8,24].

For the sake of completeness the essential features of the method are repeated here.

The boundary layer assumptions made for a statistically stationary, steady, incompressible two-dimensional flow in which the molecular and normal turbulent stresses are neglected, gives rise to the following equation for the conservation of momentum in the x-direction

$$\Gamma(u) = u\frac{\partial u}{\partial x} - \frac{\partial u}{\partial y}\int_0^y \frac{\partial u}{\partial x} d\xi + \frac{1}{\rho}\frac{\partial p}{\partial x} - \frac{1}{\rho}\frac{\partial \tau}{\partial y} = 0. \qquad (2.5)$$

The mass conservation relation used implicitly above is

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \qquad (2.6)$$

A further result of the boundary layer approximation is that the pressure does not vary in the direction normal to the stream.

Equations (2.5) and (2.6) are simplified by postulating that the velocity field is expressible from the nozzle to the exit plane in the manner shown in figure 10 by a function of the following form

$$\hat{u}(x,y) = u_o(x) + u_1(x)\exp\left[-\alpha\frac{y^2}{b(x)^2}\right]. \qquad (2.7)$$

It should be noted that this representation ignores the boundary

layer on the inside of the channel walls[8].

The turbulent shear stress is modeled using the simple eddy viscosity concept

$$\frac{\tau}{\rho} = \nu_t \frac{u}{y}, \tag{2.8}$$

with the following scaling hypothesis

$$\nu_t = k u_1 b. \tag{2.9}$$

Experimental observations for the growth rate of a free jet is used to assign a value of 0.0283 to the constant k.

Now substitute Eqs. (2.7)-(2.9) into Eqs. (2.5) and (2.6), to obtain a simplified set of ordinary differential equations in terms of the unknown functions $u_1(x)$, $u_o(x)$, $b(x)$, and $p(x)$. A system of independent equations for these quantities is obtained by taking successive moments of of the momentum equation as follows

$$\int_0^H y^n \Gamma(\hat{u})\, dy = 0. \tag{2.10}$$

These moments together with the continuity relation constitute a closed system of equations expressible in the following matrix notation

$$[A]\begin{Bmatrix} \dot{u}_o \\ \dot{u}_1 \\ \dot{b} \\ \dot{p} \end{Bmatrix} = \{B\}. \qquad (2.11)$$

This coupled non-linear set of ordinary differential equations is solved by marching away from initial data provided at the virtual origin.

2.5 VISCOUS-INVISCID MATCHING

Communication between the viscous and inviscid zones takes place in the form of shared boundary condition data at the zonal interface (the outer edge of the shear layer produced by the jet). The goal of the viscous-inviscid matching is to make as many as possible of the flow variables continuous at the zonal interface. Continuity in both velocity and pressure fields along the jet boundary is achieved in the limit of an iterative process.

The boundary condition for the inviscid solution is the normal component of transpiration velocity used to simulate the entrainment of the jet. Solution of the inviscid problem subject to this boundary condition yields the u and v components of velocity at the zonal interface. A boundary condition for the viscous solution is created by making use of the fact that $\dot{u}_o$ may be found by numerically differentiating the u component of velocity as obtained from the inviscid solution along the

jet boundary. This allows Eq. (2.11) to be written with $\dot{u}_o$ appearing as a forcing term

$$[\tilde{A}] \begin{Bmatrix} \dot{u}_1 \\ \dot{b} \\ \dot{p} \end{Bmatrix} = \{B\} + \{C\}\dot{u}_o. \qquad (2.12)$$

This set of equations is solved together with a given set of initial conditions, and the v component of velocity at the jet boundary found from the solution through use of the continuity relation. Thus the v component of velocity along the jet boundary may be calculated from both the viscous and inviscid formulations. The aim of the iteration scheme is to find the distribution of normal transpiration velocity, used as a boundary condition for the inviscid solution, which matches the v component of velocity along the jet boundary as computed from the viscous and inviscid solutions. Physically this corresponds to finding the proper entrainment distribution for the given jet initial conditions and geometrical boundary conditions. Matching of the pressure field at the jet boundary is achieved automatically when the velocity field is compatible, since the pressure in the inviscid region of the jet profile is required to obey the Bernoulli equation[8].

To start the iteration process, an initial guess for the boundary condition to the inviscid flow problem is chosen (a reasonable choice is a uniform flow boundary condition). The

cycle is terminated when the v components of velocity agree to some specified tolerance.

2.6 STABILITY CHARACTERISTICS OF THE ITERATION SCHEME

It has been found that the iteration process is unstable when the correction to the transpiration velocity boundary condition is made using a classical relaxation scheme with a constant relaxation factor such as

$$V_N^{n+1} = V_N^n + \omega(v_{vis} - v_{inv}), \qquad (2.13)$$

where V_N is the normal component of transpiration velocity at the jet boundary, n the iteration level, and v_{vis} and v_{inv} the y component of velocity as computed by the inviscid and viscous solutions respectively. Under such a scheme, an instability develops near the control station and grows rapidly as is propagates upstream towards the nozzle. Regardless of its value, inclusion of a constant relaxation factor ω is shown to have little effect on controlling the instability.

The assessment is made that the stability characteristics of this scheme vary with the distance from the virtual origin of the jet, and thus the motivation arises to let the relaxation factor become a function of the streamwise coordinate x. In this way the scheme may be damped more in the region which is most sensitive to the boundary condition correction. A linear variation in of the form

$$\omega = (r + tx), \tag{2.14}$$

where

$$r = 1.0, \quad t = -0.7/(x_{end}), \tag{2.15}$$

is found to stabilize the algorithm. Here x_{end} is the x coordinate of the furthest downstream station at which the viscous and inviscid solutions are matched. While this scheme is under-relaxed for most of the jet trajectory, the number of cycles needed to achieve a three decade drop in the solution error is about 4. This scheme is quite robust, and no further stability problems are encountered even for a wide range of geometrical configurations and flight conditions.

Once convergence is achieved, the viscous-inviscid matching is complete and the remainder of the viscous flow within the mixing channel is computed by marching the full set of Eqs. (2.11) downstream from the control station to the shroud exit.

2.7 EXIT PRESSURE MATCHING

Upon exit of the shroud, the requirement is posed that the pressure be continuous across the slipstream created between the viscous and inviscid calculations made there. The pressure computed by the inviscid solution at the primary jet nozzle is used as an initial condition for marching of the viscous solution. At the exit of the shroud, the pressure predicted

there by the viscous solution is compared with the pressure computed by the inviscid solution. Compatibility between these pressure fields is achieved by iterctively adjusting the momentum flux of the primary jet (equivalent to adjusting the initial value of u_1). The following relaxation scheme is used to correct the jet initial conditions in response to a exit pressure inbalance

$$u_1(0)^{n+1} = u_1(0)^n + \omega(p_{inv} - p_{vis}), \tag{2.16}$$

where

$$\omega = 20. \tag{2.17}$$

No stability problems are encountered with this scheme.

The pressures on either side of the slipstream at the shroud exit can be compared only once the velocity field is matched at the jet boundary. For this reason it is necessary to nest the iteration scheme for matching the velocity at the zonal interface within the iteration loop for matching the pressure at the exit. Starting with a given value of u_1 to be used as the initial condition in the integral method, the viscous and inviscid zones are first matched, thereby allowing the viscous and inviscid pressure predictions at the exit to be compared. The outer loop is closed as the initial centerline velocity of the jet (u_1) is adjusted in response to the computed pressure inbalance. The overall scheme is shown schematically in Figure 11.

2.8 BOUNDARY LAYER CALCULATION

The boundary layer calculation performed here is based on von Karman's integral formulation of the boundary layer equations. Transition and separation are predicted by monitoring the local shape factors as suggested by Eppler[9]. As theories for a turbulent boundary layer which evolves in a turbulent outer field are not well developed, the boundary layer calculation is terminated at the point at which the jet first interacts with the surface of the shroud. The lack of ability to calculate the boundary layer throughout the entire s . ud prohibits the computation of skin friction over the entire device. For this reason the boundary layer calculation serves only to indicate separation at the inlet and not to provide a measure of the drag induced by the shroud. In addition to this, no attempt is made to account for the displacement effects of the boundary layers. It is anticipated that the displacement effects do not appreciably affect the overall performance of the device, and without displacement thickness information within the mixing channel, there is little point in making a correction.

3. COMPARISON WITH EXPERIMENT

The augmentor algorithm is compared with a series of measurements taken by Bernal and Sarohia[10] at the Jet Propulsion Laboratory. A cross-section of the two-dimensional model tested is shown in Figure 12. The surface pressure distribution predicted by the augmentor code is compared with experiment in Figure 13. The agreement is seen to be quite good over most of the shroud with the suction peak location and magnitude properly predicted. The pressure deviates most within the mixing channel, which is likely to be a consequence of the crude turbulence model used in this region. Note the strong adverse pressure gradient following the suction peak. The boundary layers which develop on the shroud walls are prone to separate in this region, and for this reason close attention must be paid to the boundary layer development in the design of a thrust augmentor in order to avoid inlet stall.

Figure 14 shows the evolution of the jet velocity profile within the mixing channel. The overall agreement is good, with the computed profiles reproducing the correct shape and velocity magnitude. The code predicts an augmentation ratio of 1.26 while the experiment yields a value of 1.2. The five percent discrepancy in augmentation ratio falls within the uncertainty of the reported value.

The CPU time required for the computation of a single converged flowfield is approximately two minutes on the VAX-11 7/780 system.

The close agreement with the experimental data suggests that the viscous-inviscid assumption as well as the eddy viscosity turbulence model provide a good approximation to the physical processes. The computational time is slight enough to allow an optimization study to be undertaken in which hundreds of separate configurations must be evaluated. Such an optimization study is discussed in the next section.

4. OPTIMIZATION STUDIES

As illustration of the capabilities of the model, a simple study for the optimization of the inlet detail for the shroud studied by Bernal and Sarohia is undertaken. A dimensional analysis is performed in order to determine similarity rules, which aside from their own utility, serve to reduce the number of free parameters. A quasi-Newton optimization routine[11] is coupled with the augmentor code in order to systematically search through the free parameters which survive the dimensional analysis. The constraints imposed both by geometric restrictions and flow separation are incorporated into the optimization scheme through algebraic penalty functions which discount the performance rating once a constraint is violated.

4.1 A SIMPLE INLET DESIGN PROBLEM

Figure 15 shows a perturbed version of the previously studied shroud in which the jet nozzle is free to move along the plane of symmetry, and a section of the inlet lip is allowed to rotate out of the plane of the mixing channel. The geometric design variables are the jet nozzle location X_o, the overall length of the shroud L, the height of the mixing channel 2H, the thickness of the shroud d, the length of the inlet section rotated X_L, and the rotation angle θ. In addition to the geometric parameters, the free-stream speed u_∞, as well as the thrust of the primary nozzle T_o are included as design variables. In order

to isolate the effects of inlet shape, shrouds with diffusers have not been considered, and the mixing channel walls remain parallel up to the exit station in all cases.

4.2 DIMENSIONAL ANALYSIS

The thrust developed by the augmentor is expected to depend on the each of the design parameters. Thus symbolically the statement is made

$$T = T(T_o, \mu, \rho, u_\infty, L, 2H, d, X_o, X_L, \theta). \tag{4.1}$$

The dimensions of the 11 parameters which appear above are all composed of the three basic dimensions of mass, length, and time. The Buckingham pi theorem[25] of dimensional analysis states that the number of dimensionless groups is equal to the number of independent parameters minus the number of independent basic dimensions. Thus in this case 8 dimensionless groups exist. Due to a further result of the pi theorem, one group may be expressed as a function of all the rest. Thus

$$\phi = \frac{T}{T_o} = f\left[\frac{u_\infty}{\sqrt{T_o/\rho 2H}}, \frac{\sqrt{T_o/\rho 2H}(2H)}{\mu/\rho}, \frac{X_o}{2H}, \frac{X_L}{2H}, \theta, \frac{L}{2H}, \frac{d}{2H}\right]. \tag{4.2}$$

This relation states that the thrust augmentation ratio is a function of seven dimensionless parameter groups. The first group appearing above is a a dimensionless measure of the free-stream

speed. The characteristic velocity used in its normalization,

$$u_c = \sqrt{\frac{T_o}{\rho 2H}}, \tag{4.3}$$

is derived from the momentum flux of the primary jet. It has the physical interpretation as being the exit velocity of a fictitious jet with the same momentum flux as the primary jet, but with an exit width equal to the mixing channel height, 2H. This same velocity scale is used in the definition of the Reynolds number, which is the second parameter in Eq. (4.2). The jet characteristic velocity is used in favor of the free-stream velocity in this case so that the Reynolds number remains defined when configurations are evaluated in the absence of a free-stream.

The remaining parameters in Eq. (4.2) are dimensionless measures of the geometric detail of the thrust augmentor configuration. In an effort to contain the scope of the optimization study, the shroud aspect ratio, L/2H is held fixed at 3.28, and the shroud thickness relative to the mixing channel height, d/2H is held fixed at 0.5. With these two parameters fixed, the overall dimensions of the optimized configurations are identical to the shroud tested by Bernal and Sarohia. As an additional consequence, the optimization problem is reduced from a seven to a five parameter search.

4.3 OBJECTIVE FUNCTION AND CONSTRAINTS

The primary objective of the optimization study is to maximize

the thrust augmentation ratio. Naturally then the objective function is taken to be that which defines ϕ (Eq. (4.2) with L/2H and d/2H deleted). A concise statement of the problem is

$$\text{Maximize } \phi = f\left[\frac{X_o}{2H}, \frac{X_L}{2H}, \theta\,;\ \frac{u_o}{\sqrt{T_o/\rho 2H}}, \frac{\sqrt{T_o/\rho 2H}\,(2H)}{\mu/\rho}\right], \tag{4.4}$$

where the elements of f belong to the constraint space defined below. The dimensionless free-stream velocity and Reynolds number are treated as parameters rather than design variables in the optimization process. That is, a free-stream velocity and Reynolds number are chosen, and then the inlet geometry optimized for that particular flight condition.

The constraints arise both from geometric restrictions and boundary layer separation over the inlet region of the shroud. As the search path of the optimization routine can not be controlled, is is necessary to employ geometric constraints in order to avoid situations in which the evaluation of a non-physical configuration is attempted. As an example, without constraints, the optimization routine may call for the evaluation of a configuration whose combination of lip rotation point and negative rotation angle would require that the two lobes of the shroud cross over each other along the symmetry plane. Each geometric design variable is constrained within limits which insure a realizable geometry.

An additional constraint is imposed by the boundary layer behavior along the inlet region of the shroud. Since inlet stall greatly detracts from the performance of the device, configurations for which boundary layer separation is predicted are rejected in the optimization process.

4.4 PENALTY FUNCTION TRANSFORMATION

Unconstrained optimization problems are much simpler to treat than are constrained ones[12]. For this reason the present constrained problem is recast as an unconstrained problem through the use of the penalty function transformation[12]. The use of penalty functions allows the constraints tc be ignored until one of them is violated. When a constraint is violated, the performance rating is artificially lowered in an effort to redirect the search away from the forbidden region. In the present study, algebraic penalty functions based on quadratic relations are used. The transformed objective function is defined as

$$\text{Maximize } g = \phi - \sum C_i(\delta_i)^2, \tag{4.5}$$

where the C_i are weighting coefficients and the δ_i are the penalty functions. The penalty functions are defined as

$$\delta_1 = R\left[\frac{X_L}{2H} - \frac{X_o}{2H} + \frac{L_j}{(2H)\tan(12^o)} - \frac{R_n}{2H}\right], \tag{4.6}$$

$$\delta_2 = R\left[\frac{R_n}{2H} - \frac{X_L}{2H}\right], \tag{4.7}$$

$$\delta_3 = R\left[-\frac{X_o}{2H}\tan(12^o) - \frac{X_L}{2H}\tan(\theta) - \frac{1}{2}\right], \tag{4.8}$$

$$\delta_4 = R\left[-\frac{X_o}{2H} - \frac{L_j}{2H} - \frac{R_n}{2H}\right], \tag{4.9}$$

$$\delta_5 = R(1.0 - S_{crit}), \tag{4.10}$$

where L_j is the horizontal length of the viscous-inviscid interaction zone, R_n is the nose radius, S_{crit} is the surface coordinate at which the boundary layer has separated, and R is the ramp function. The surface coordinate S_{crit} is normalized such that its origin corresponds to the stagnation point and the control station corresponds to a value of 1. If the boundary layer remains attached S_{crit} is assigned a value of 1.

Penalty function δ_1 insures that the lip rotation point is upstream of the control station. Penalty function δ_2 insures that the lip rotation point is downstream of the nose radius. Penalty function δ_3 insures that the lip does not rotate into the viscous region. Penalty function δ_4 limits the distance at which the nozzle may be placed upstream of the shroud. This restriction insures that the jet has not yet expanded to a width greater than that of the mixing chamber upon entrance into the

device. Penalty function δ_5 insures that the boundary layer remains attached over the inlet region of the shroud.

The weighting coefficients are a measure of the relative importance of respect for the constraints and the desire to obtain the highest possible thrust augmentation ratio. Low values of C imply little attention is paid to the constraints, while large values of C increase their importance. Difficulties in achieving convergence arise when the weighting coefficients take on very large values. Each of the weighting coefficients are started at a value of 1, and then increased to a value of 10 as the optimal point is neared. In some instances the values of individual weighting weighting coefficients may have to be adjusted slightly in order to achieve convergence.

4.5 OPTIMAL SOLUTIONS

Optimal configurations are determined for a wide range of Reynolds number (R_T) for three values of the dimensionless free-stream velocity (γ). Figure 16 shows the variation in the performance of a thrust augmentor with an optimized inlet as a function of both Reynolds number and free stream speed. The results show that the performance is an increasing function of Reynolds number, with strongest dependence in the low Reynolds number range. The rapid increase in performance at low Reynolds numbers is associated with transition from a laminar to a turbulent boundary layer. A laminar boundary layer can not withstand the severe adverse pressure gradient which is present in the

inlet region. In an effort to avoid inlet stall, the optimization routine seeks configurations which reduce the pressure rise in the inlet region by decreasing the degree of turbulent mixing within the shroud. In so doing, the performance is decreased since the mechanism of thrust augmentation relies on mixing of the high momentum jet with the ambient fluid. As the Reynolds number is increased to a value sufficient to induce transition to a turbulent boundary layer, the performance is greatly enhanced due to the fact that the turbulent boundary layer is able to negotiate the intensified pressure rise associated with increased mixing within the shroud.

When a non-zero free-stream speed is included, the presence of a strong favorable pressure gradient following the stagnation point at the shroud nose helps to energize the boundary layer, thus making it more resilient to separation as the pressure rise in the inlet region is encountered. In Contrast to this, in the case of static operation, the boundary layer begins at the tail end of the shroud, and due to its lengthy evolution and less favorable pressure gradient, becomes thick and sluggish by the time it has traveled the distance necessary to be swept into the inlet. The resulting thick, poorly energized boundary layer experiences separation at a smaller pressure rise than the one which the more favorably energized boundary layer can withstand. For this reason, increased levels of performance are noted in the laminar regime when a free-stream velocity is present.

In the high Reynolds number regime, performance decreases with increasing free-stream speed. This is due in part to the normalization used in the definition of the thrust augmentation ratio, and due in part to a reduction in shear at the jet boundary. The thrust augmentation ratio is defined as the gross thrust divided by the the thrust produced by the primary nozzle when exhausted in an otherwise quiescent medium. When a free-stream velocity is present, the actual thrust produced by the primary jet is reduced by an amount equal to the momentum flux of the free-stream across the jet nozzle area. As a consequence of this, the gross thrust is also reduced, and thus when compared with the primary thrust produced in a quiescent medium, an apparent drop in the thrust augmentation ratio arises. An alternate definition of thrust augmentation ratio could be developed in which the gross thrust is normalized by the free-stream reduced primary thrust. However the standard definition is used here.

In addition to the apparent drop in performance due to the normalization, there exists a physical loss of performance due to the reduced rate of shear in the jet layer which occurs as the free-stream velocity effectively lowers the difference between the ambient fluid velocity and jet velocity. A reduction in entrainment of the ambient fluid follows the reduction in the rate of shear and thus the performance drops.

A few representative optimal shapes corresponding to the performance curve are shown in Figures 17 and 18. The results show that the optimal design shapes are a much stronger function

of Reynolds number than free-stream speed. At low Reynolds number, Figure 17 shows that the optimal nozzle position is located up to one channel width ahead of the shroud, while the inlet is slightly expanded. This combination serves to minimize the adverse pressure gradient in the inlet region as required by the laminar boundary layer which develops there. In Figure 18 as the Reynolds number is increased and the boundary layers undergo transition, the nozzle moves roughly up to the entrance plane of the shroud. The inlet lips rotate through the horizontal and then towards the jet as the Reynolds number is increased. The length of the inlet lip which is rotated is seen to increase with Reynolds number.

More detail on the behavior of the various design parameters as the Reynolds number and dimensionless free-stream speed are varied is shown in Figures 19-21. Figure 19 illustrates the optimal lip rotation angle as a function of Reynolds number for three values of the dimensionless free-stream speed. It can be seen that the optimal lip rotation angles follow a similar trend for all three values of dimensionless free-stream velocity. As the Reynolds number is increased, and laminar boundary layers give way to turbulent ones, the lips rotate quickly from large positive angles to a position of roughly zero angle. Further increase in the Reynolds number causes a continual gradual decline in the lip rotation angle. Differences in the optimal lip rotation angle due the free-stream speed become increasingly small as in the high Reynolds number regime.

Displayed in Figure 20 is the optimal primary nozzle location as a function of Reynolds number for the three values of the dimensionless free-stream speed. The trends are qualitatively similar for each of the three values of free-stream speed. In the low Reynolds number limit, the nozzle is located well in front of the shroud due to the fragile nature of the laminar boundary layers. As the Reynolds number is increased and the boundary layers become turbulent, the optimal nozzle position moves quickly to a limiting point just inside the shroud. In light of the forward stagnation point induced by the free-stream and its positive effect on the boundary layer development, the optimal nozzle location moves forward more quickly when a free-stream is present as compared to static operation.

Figure 21 illustrates the optimal length of the inlet lip plotted as a function of Reynolds number with the dimensionless free-stream velocity appearing as a parameter. The general trend of a short lip at low Reynolds number, maximum lip length at moderate Reynolds number and a decline in lip length with very large Reynolds number is seen to hold for all three values of the dimensionless free-stream velocity. Again due to the forward stagnation point, there is a shift in Reynolds number when the results for static operation are compared with those for a non-zero free-stream. The rapid change in the lip length when moving out of the low Reynolds number regime is due to boundary layer transition.

5. CONCLUSIONS AND DISCUSSION

A viscous-inviscid methodology for the calculation of two-dimensional thrust augmentor performance is developed, which, through combining a higher order panel method with an integral method, allows for economic and robust computation. Good qualitative and quantitative agreement with experiment is obtained with the thrust augmentation prediction matching the experiment within 5%. The computational time necessary to compute the flowfield is roughly two minutes on the VAX/11-780 machine. The potential for use of the code as a practical design tool is demonstrated through an optimization study for the inlet shape and nozzle placement. Reynolds number effects captured through a boundary layer calculation are shown to be an important design parameter.

An interesting observation is that barring separation, and constraining the mixing channel height to be a constant, the optimal configurations have an inlet which constricts the entering flow. As the inlet is narrowed, the entrained flow is forced to achieve a higher speed as it enters the shroud. This increase in inlet velocity reduces the rate of shear produced at the jet boundary, thereby reducing the entrainment. However, at the same time the increased inlet velocity enhances the convective acceleration about the nose, thereby increasing the induced thrust. Evidently the increase in fluid speed about the nose plays a more important role than does the decrease in entrainment.

Bevilaqua has observed a similar trend for the case of a straight walled inlet in which the ratio formed between the inlet area and the jet nozzle area was varied[3].

An important conclusion of the optimization study is the fact that inlet boundary layer separation is the determining factor in the maximum obtainable performance. In low Reynolds number operation, the presence of laminar boundary layers poses a problem because of their tendency to separate in the region of pressure recovery within the inlet. In order to design a configuration to be free of inlet stall in the laminar regime, the performance must be compromised by reducing the degree of turbulent mixing which takes place within the inlet region of the shroud. The designer should be alerted to this fact and try to either force the boundary layer to become turbulent or employ some other means of boundary layer control. It is clear that some form of boundary layer management will not only substantially increase the performance, but will also allow a single configuration to operate efficiently over a wider range of Reynolds numbers and free-stream speeds. As a means of boundary layer control, the use of a wall jet may be advantageous since it is likely to enhance the turbulent mixing within the shroud.

REFERENCES

1. T. von Karman, "Theoretical Remarks on Thrust Augmentation", in contributions to Applied Mechanics, Reissner Anniversary Volume, pub. by J. W. Edwards, Ann Arbor, Mich., 1949, pp. 461-468.

2. T. A. Lasinski, A. E. Andrews, R. L. Sorenson, D. S. Chaussee, T. H. Pulliam, and P. Kutler, "Computation of the Steady Viscous Flow over a Tri-Element Augmentor-Wing Airfoil", AIAA paper No. 82-0021, 1982.

3. P. M. Bevilaqua and A. D. DeJoode, "Viscid/Inviscid Interaction Analysis of Ejector Wings", Workshop on Thrust Augmenting Ejectors, NASA CP 2093, 1979.

4. P. M. Bevilaqua, C. J. Woan, and E. F. Shum, "Viscid/Inviscid Interaction Analysis of Ejector Wings", Ejector Workshop for Aerospace Applications, AFWAL-TR-82-3059, 1982.

5. D. A. Tavella and L. Roberts, "A Simple Viscous-Inviscid Aerodynamic Analysis of Two-Dimensional Ejectors", AIAA paper NO. 84-0281, 1984.

6. J. L. Hess, "Higher Order Numerical Solution for the Integral

Equation for the Two-Dimensional Neumann Problem", Computer Methods in Applied Mechanics and Engineering, NO. 2, 1973.

7. J. L. Hess and A. M. O. Smith, "Calculation of Potential Flow about Arbitrary Bodies", pp. 1-138, Progress in Aeronautical Sciences, Vol. 8, Pergamon Press, 1967.

8. D. Tavella and L. Roberts, "Analysis of Confined Turbulent Jets", JIAA-TR-51, Stanford University, 1982.

9. R. Eppler, "Praktische Berechnung laminarer und turbulenter Absauge-Grenzschichten", Ing.-Arch. 32, pp. 221-245, 1963.

10. L. Bernal and V. Sarohia, "Entrainment and Mixing in Thrust Augmenting Ejectors", AIAA paper No. 83-0172, 1983.

11. P. E. Gill, W. Murrary, and Pitfield, "The Implementation of Two Revised Quasi-Newton Algorithms for Unconstrained Optimization", National Physics Laboratory, Div. of Numerical Analysis and Computing, Report No. DNAC 11, 1972.

12. P. E. Gill, W. Murrary, and M. H. Wright, Practical Optimization, Academic Press, London, 1981.

13. M. Van Dyke, An Album of Fluid Motion, Parabolic Press,

Stanford, 1982.

14. T. L. Holst, "Numerical Computation of Transonic Flow Governed by the Full Potential Equation", NASA TM 84310, 1983.

15. J. L. Steger, Course Notes, Stanford University, 1983.

16. L. J. Chow, T. H. Pulliam, and J. L. Steger, "A General Perturbation Approach for Computational Fluid Dynamics", AIAA Journal, Vol. 22, No. 12, 1984.

17. L. Prandtl, "Uber Flussigkeitsbewegung bei sehr kleiner Reibung", Pro. Third Intern. Math Congr., Heidelberg, 1904. English Translation NACA TM 452 1928.

18. H. Schlichting, "Laminare Strahlenausbreitung", Zeitschrift fur angewandte Mathematik und Mechanik, 13, pp. 260-263, 1933.

19. D. A. Anderson, J. C. Tannehill, and R. H. Pletcher, Computational Fluid Mechanics and Heat Transfer, McGraw-Hill, New York, 1984.

20. T. von Karman, "Uber laminare und turbulente Reibung", Zeitschrift fur angewandte Mathematik und Mechanik, 1, pp. 233-252, 1921. English Translation NACA TM 1092, 1946.

21. K. Pohlhausen, "Zur nauherungsweisen Integration der Differentialgleichung der laminaren Reibungsschicht", Zeitschrift fur angewandte Mathematik und Mechanik, 1 pp. 252-268, 1921.

22. H. Schlichting, Boundary layer theory, McGraw-Hill, New York, New York, 1979.

23. G. N. Abramovich, The Theory of Turbulent Jets, MIT Press, Cambridge, Mass, 1963.

24. D. A. Tavella and L. Roberts, "Viscous-Inviscid Aerodynamic Analysis of Two-Dimensional Thrust Augmentors", JIAA-TR-57, Stanford University, Stanford CA, 1984.

25. E. Buckingham, "On Physically Similar Systems: Illustrations of the Use of Dimensional Equations", Phys. Rev. vol.4, no. 4, pp. 345-376, 1914.

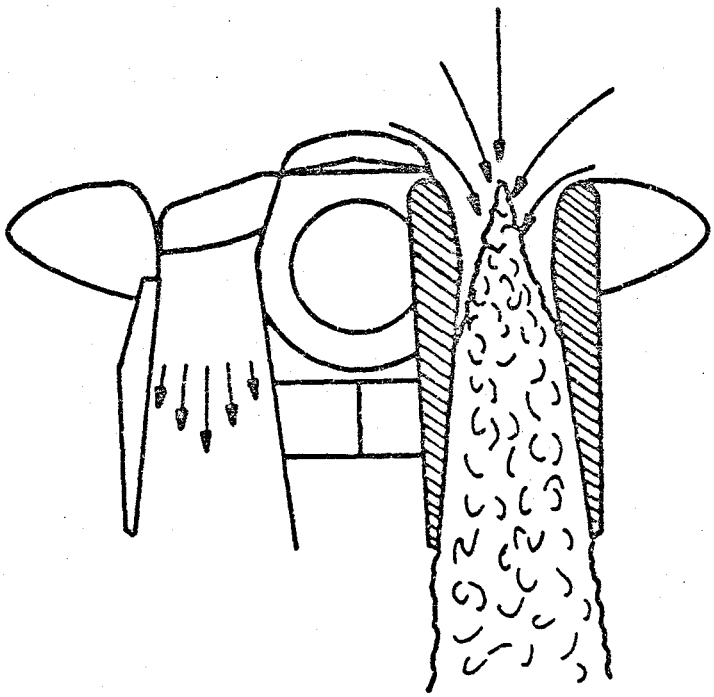


Figure 1. Lifting ejector for vertical takeoff applications. The shaded region shows the idealized ejector in which the geometry is somewhat simplified. In the present study the interference introduced by the aircraft and the other ejector are neglected, leaving a single ejector in isolation to be the focus of attention.

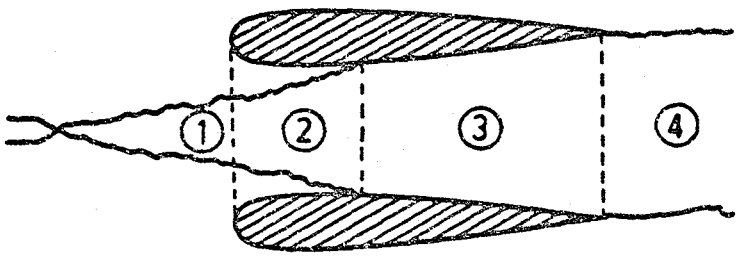
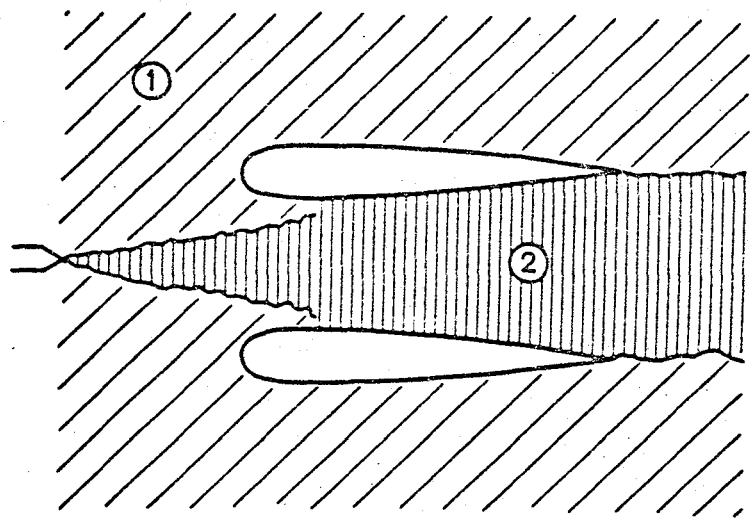


Figure 2. Viscous flow regions.



(1) Inviscid zone computed with a higher order panel method.

(2) Viscous zone computed with an integral method.

Figure 3. Thrust augmentor and the zonal approach.

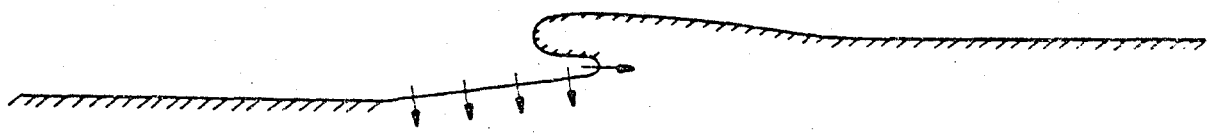


Figure 4. The inviscid problem.

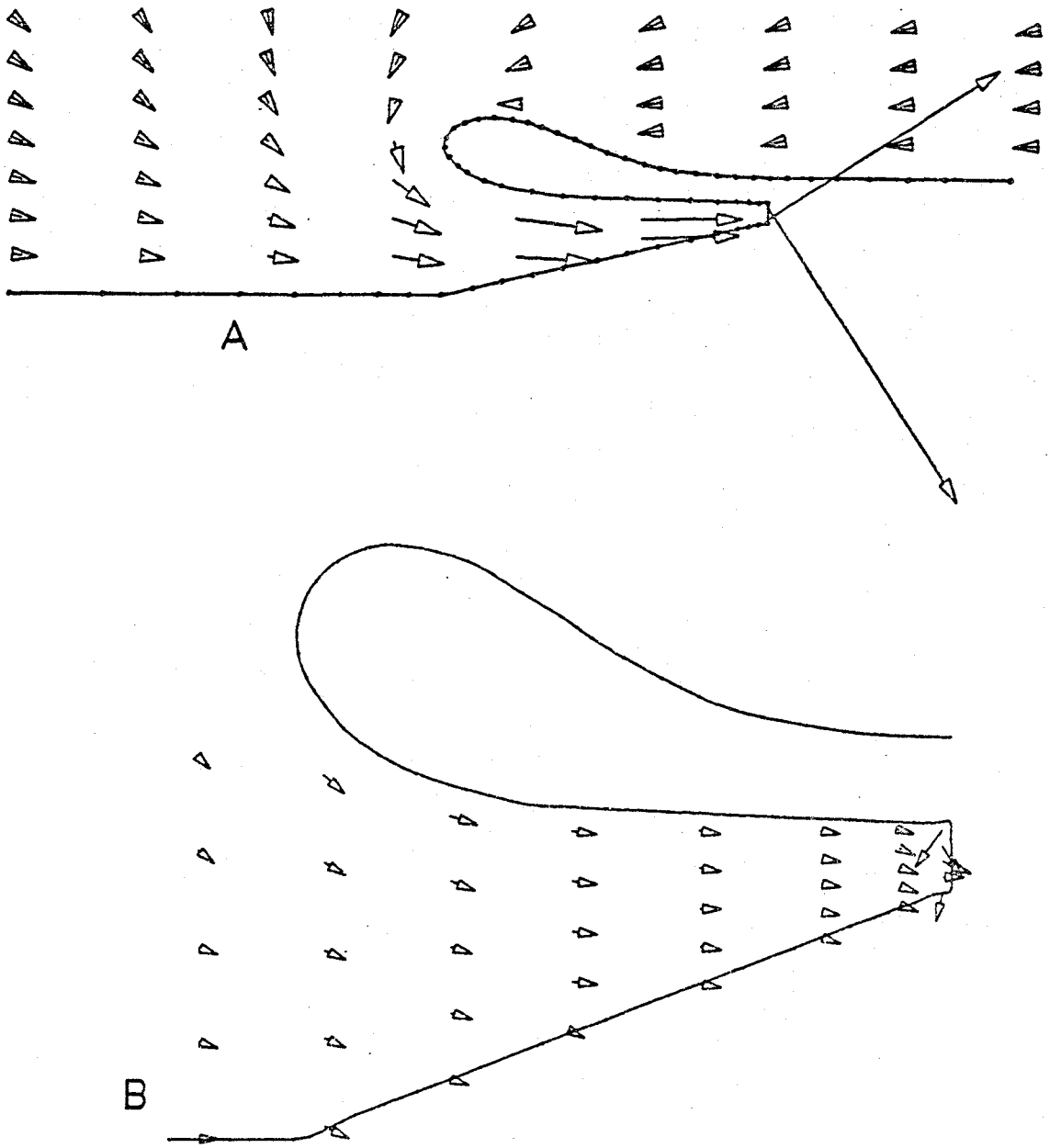


Figure 5. Inviscid flowfield computed using a classical panel method. (A) Far field, (B) Near field. Note well behaved far field and local inaccuracy near the control station.

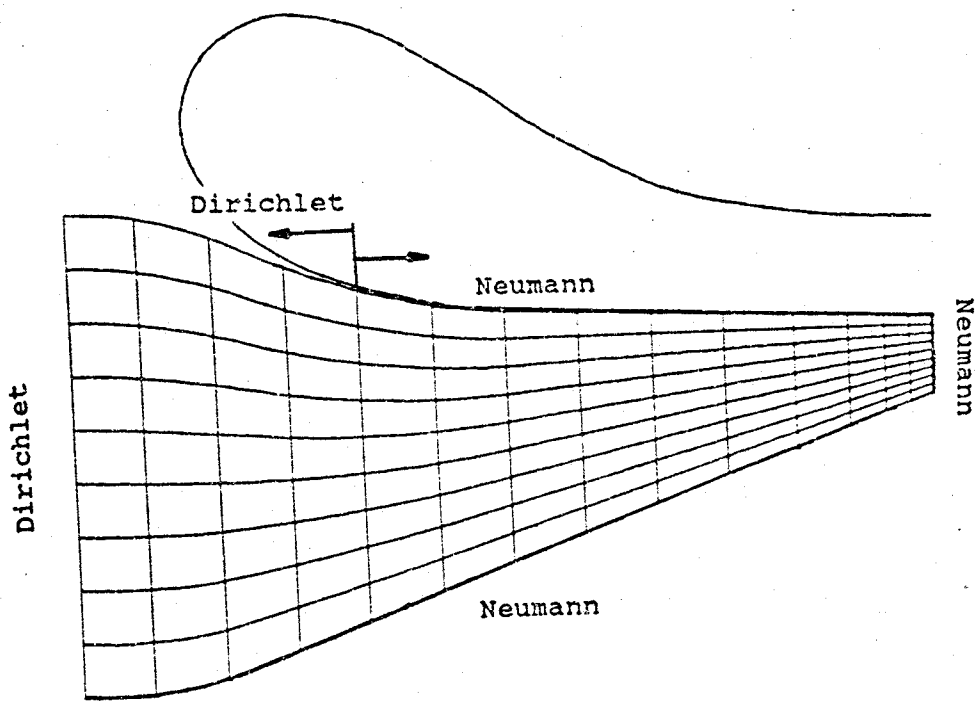


Figure 6. Hybrid scheme. The far field is computed using the classical panel method. Dirichlet data from the panel solution is used at the inflow boundaries to couple the farfield in with the local solution provided by the finite difference calculation.

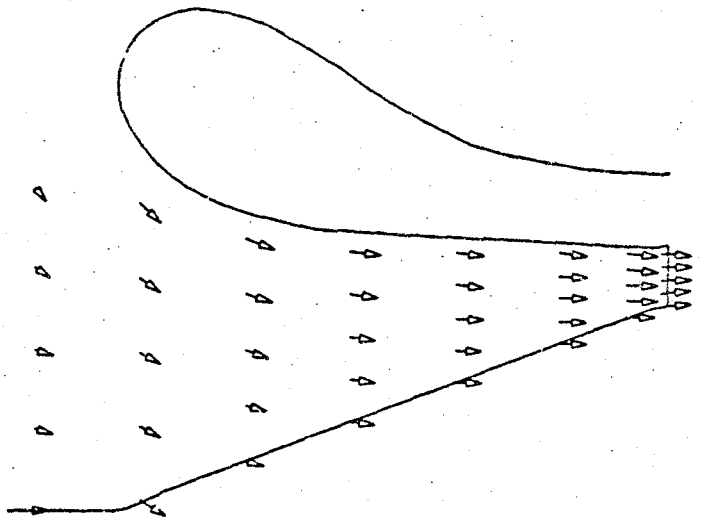


Figure 7. Near field computed using the hybrid scheme. Compare with Figure 5 for the classical panel method alone.

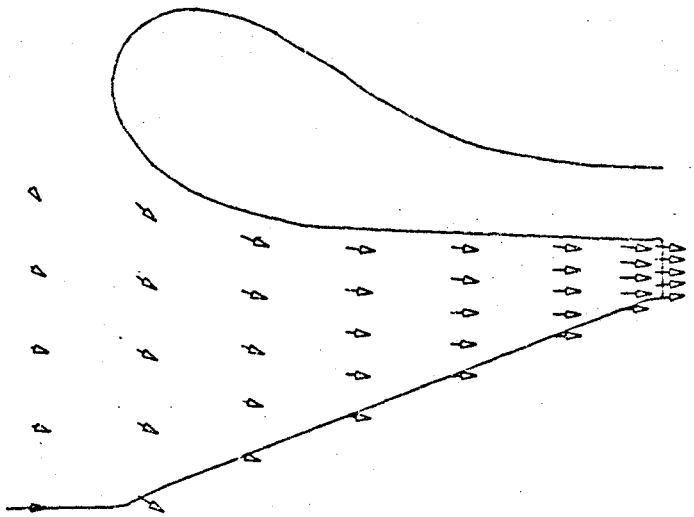


Figure 8. Near field detail of the inviscid solution as computed using the higher order panel method. To plotting accuracy the velocity field is identical to that of the the hybrid scheme.

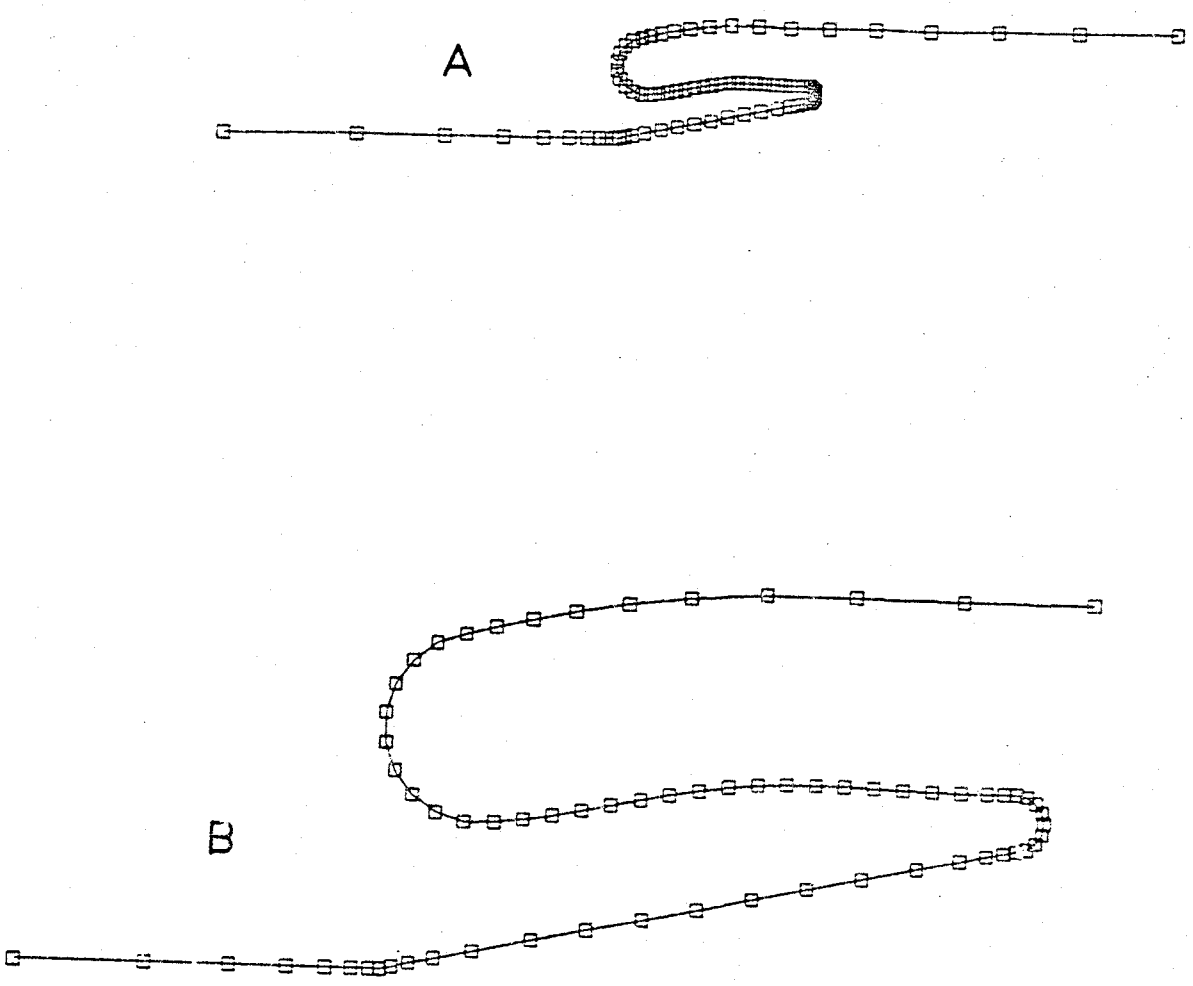


Figure 9. Panel distribution used for the higher order method. (A) Overall view, (B) Inlet detail. Not all the panels extending fore and aft from the thrust augmentor are shown in (A). The panels continue four shroud lengths ahead of and behind the ejector.

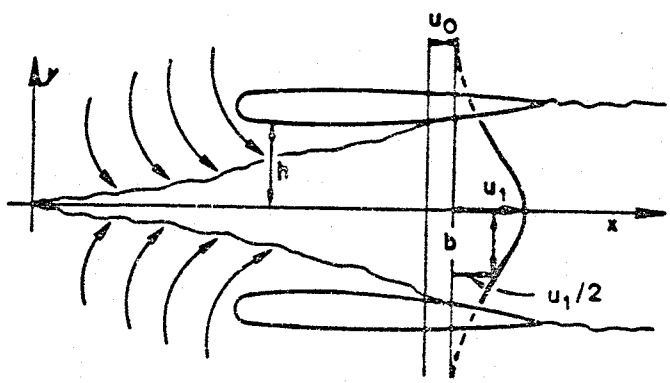


Figure 10. Viscous solution.

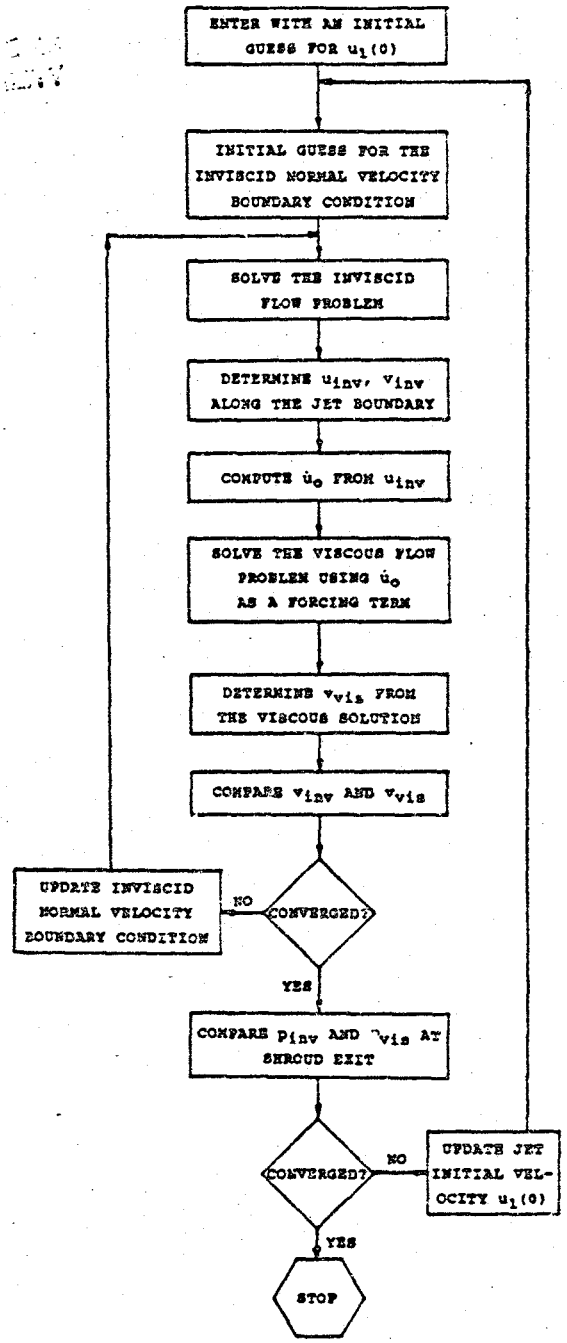


Figure 11. Iteration scheme.

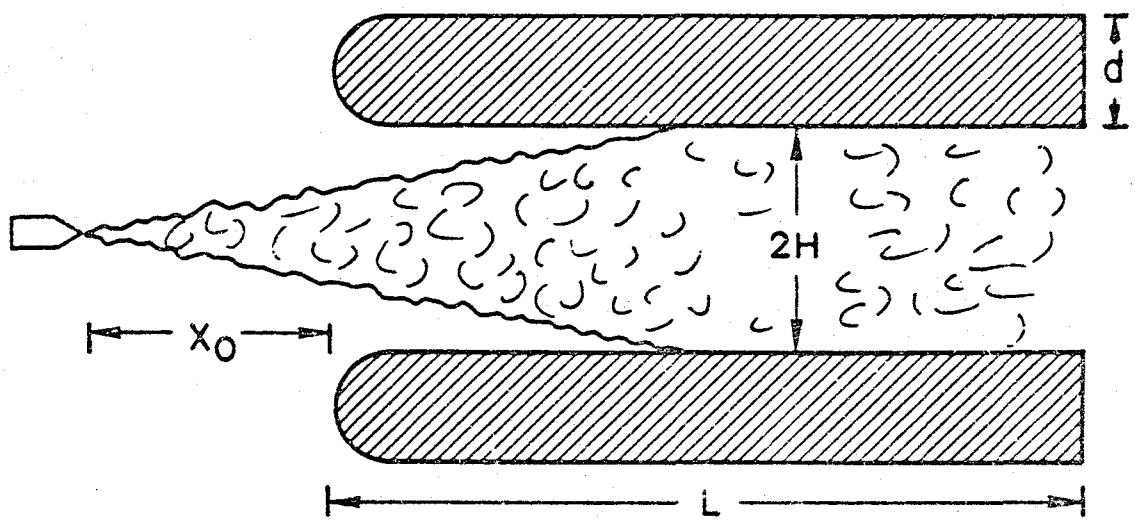


Figure 12. Test configuration. $L/2H = 3.28$, $X_o/2H = 1.0$, $d/2H = 0.5$. (Taken from Ref. 10).

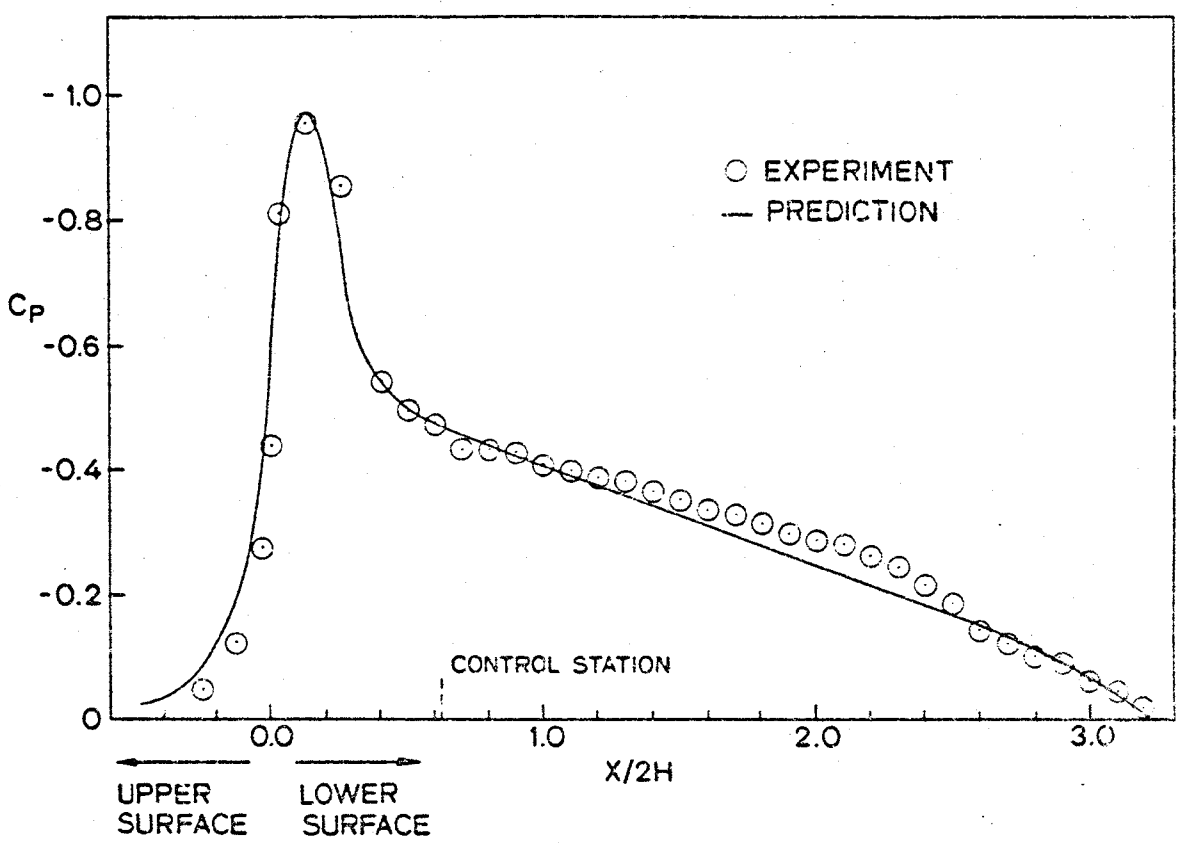


Figure 13. Comparison of surface pressure distribution.

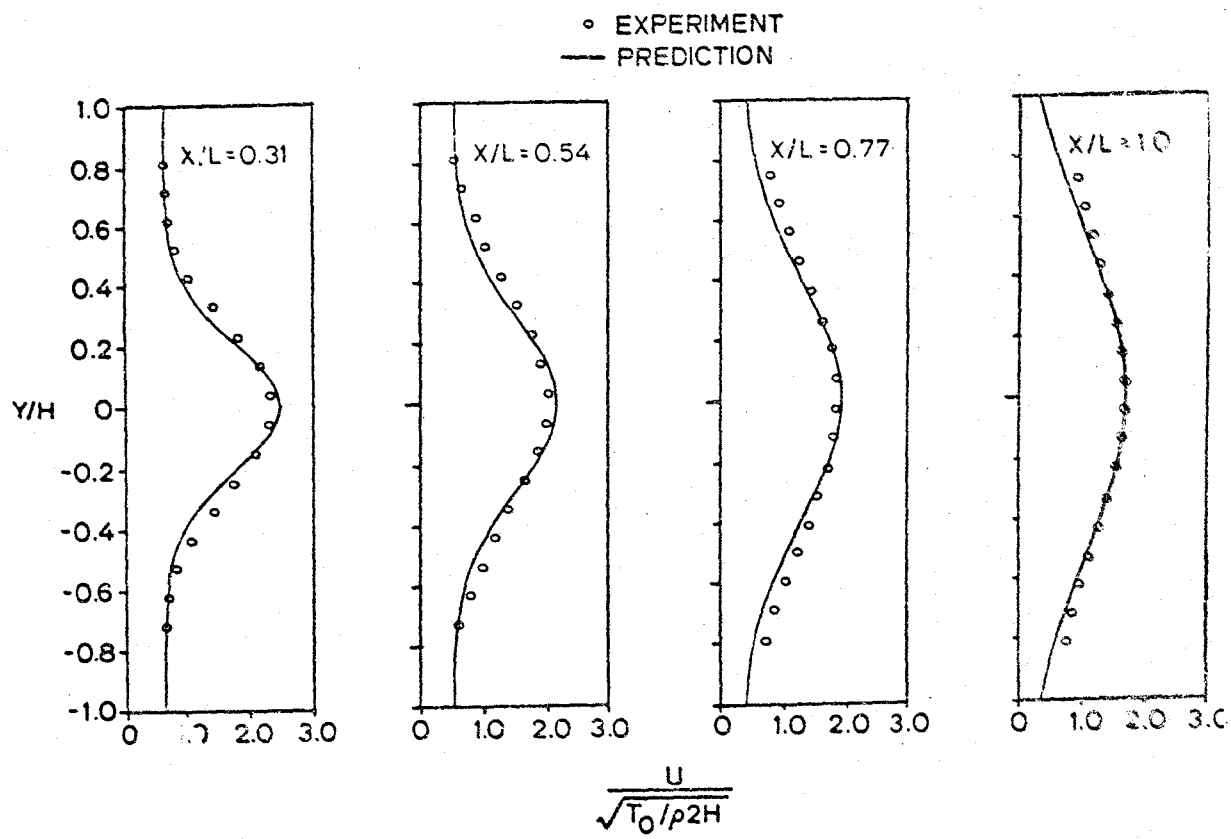


Figure 14. Comparison of jet velocity profile.

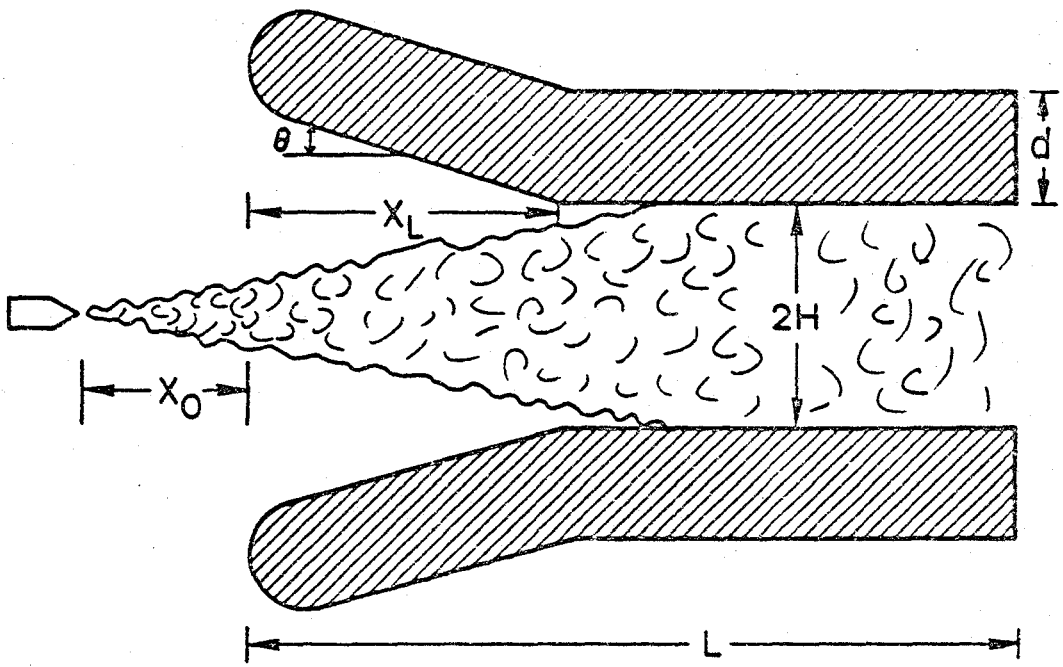


Figure 15. Optimization parameters. X_o, X_L, and θ are variable, L/2H is fixed at 3.28, d/2H is fixed at 0.5. u_o and T_o are additional parameters.

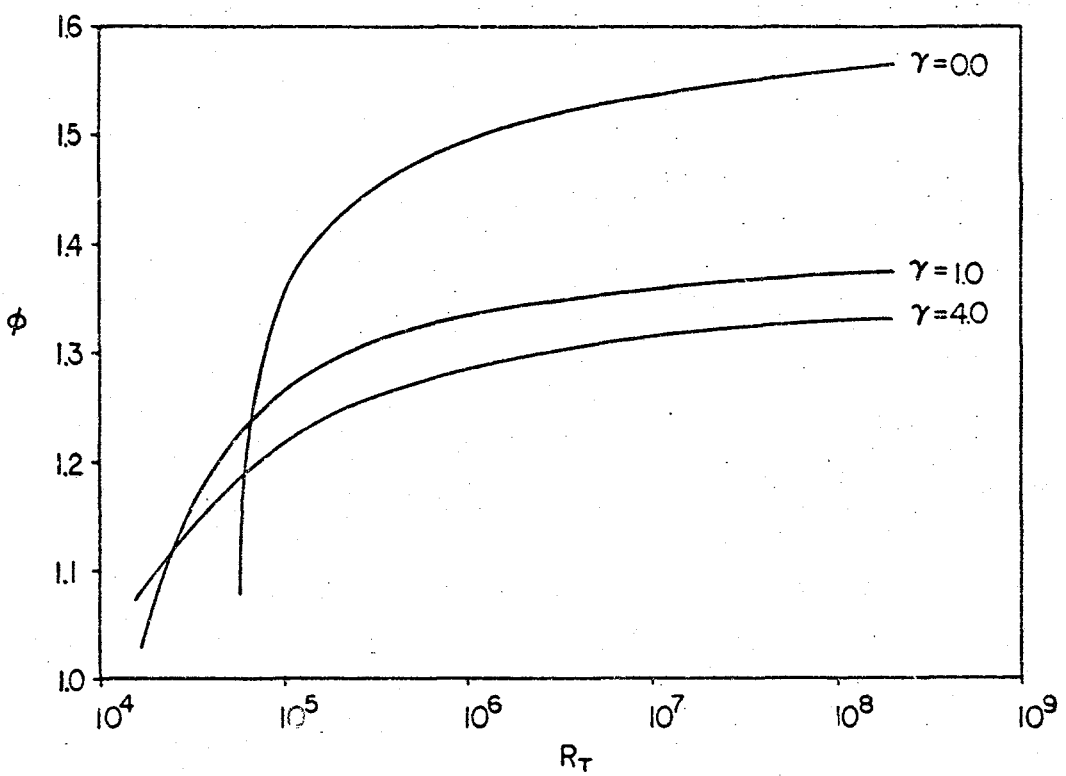


Figure 16. Performance of the thrust augmentor with an optimized inlet.

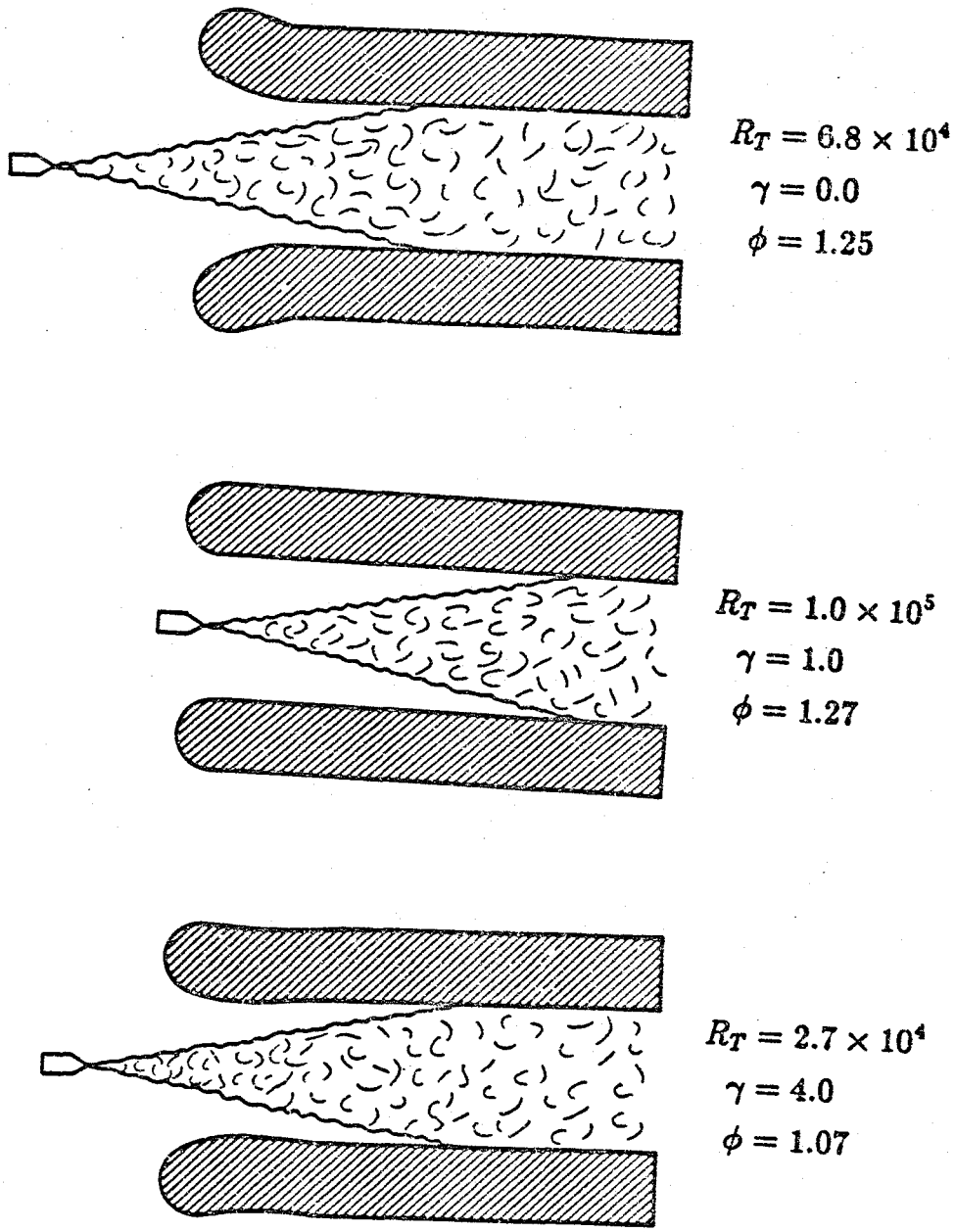
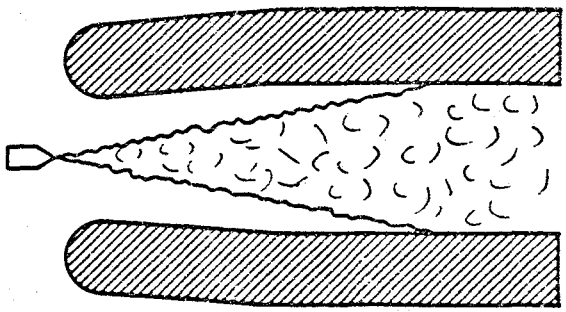
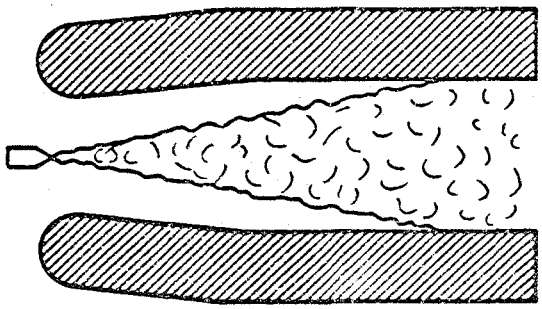


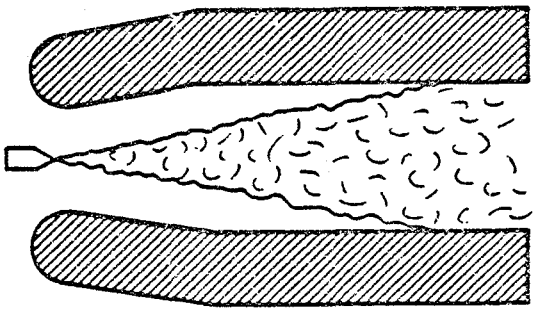
Figure 17. Optimal configurations at low and moderate Reynolds number.



$R_T = 1.0 \times 10^7$
$\gamma = 0.0$
$\phi = 1.54$



$R_T = 6.4 \times 10^6$
$\gamma = 1.0$
$\phi = 1.37$



$R_T = 9.2 \times 10^7$
$\gamma = 4.0$
$\phi = 1.31$

Figure 18. Optimal configurations at high Reynolds number.

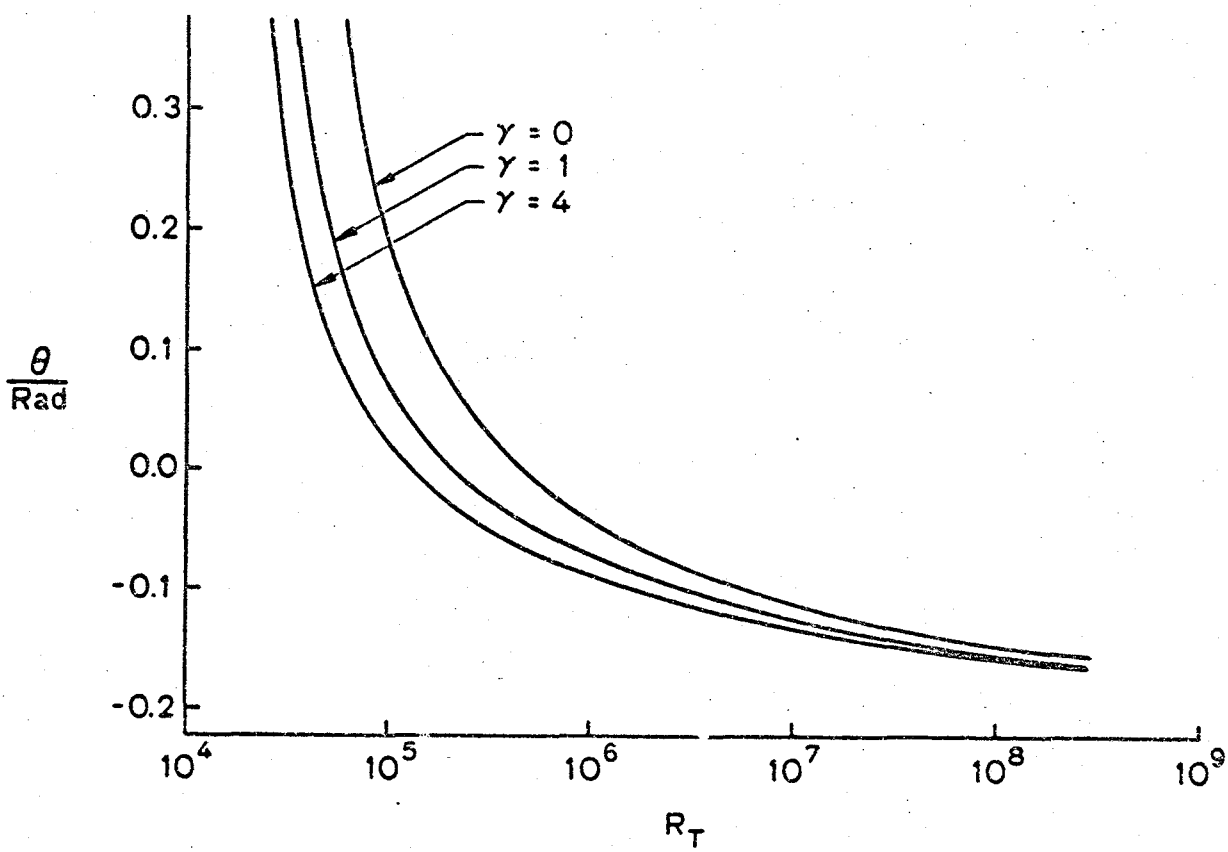


Figure 19. Optimal lip rotation angle.

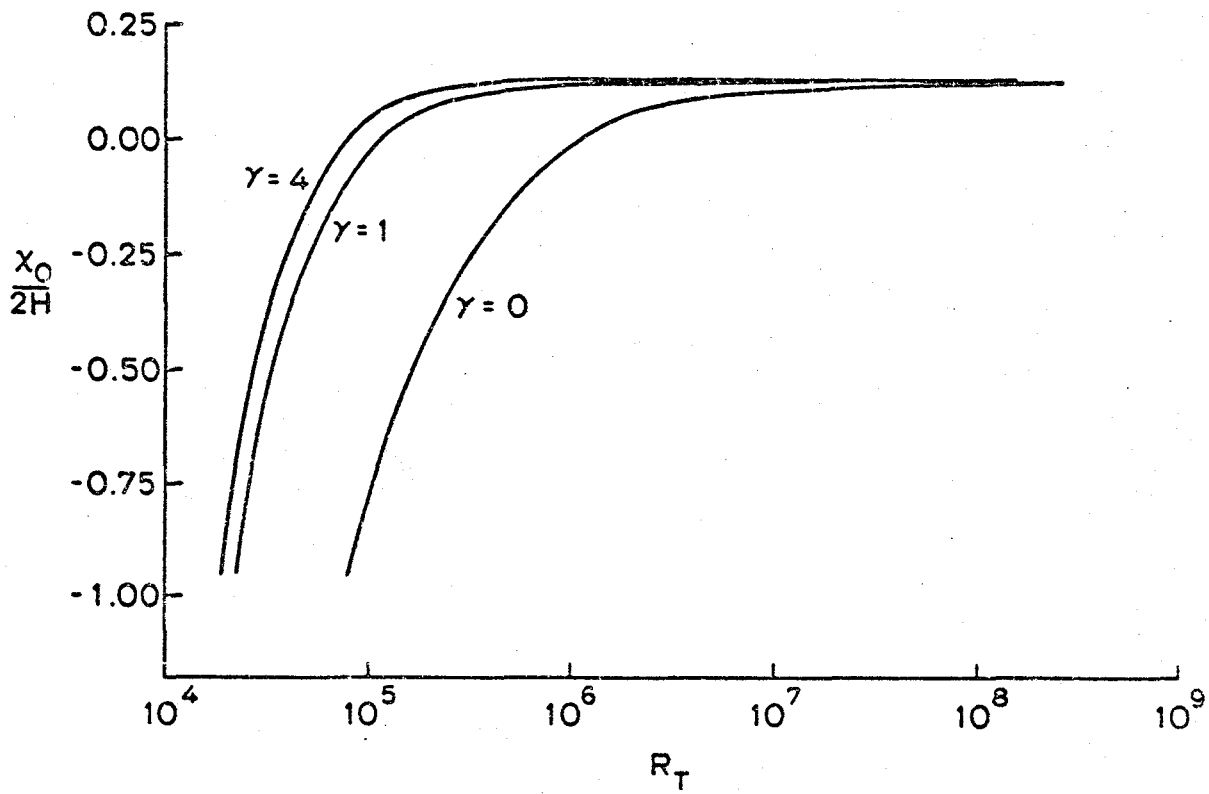


Figure 20. optimal primary nozzle position.

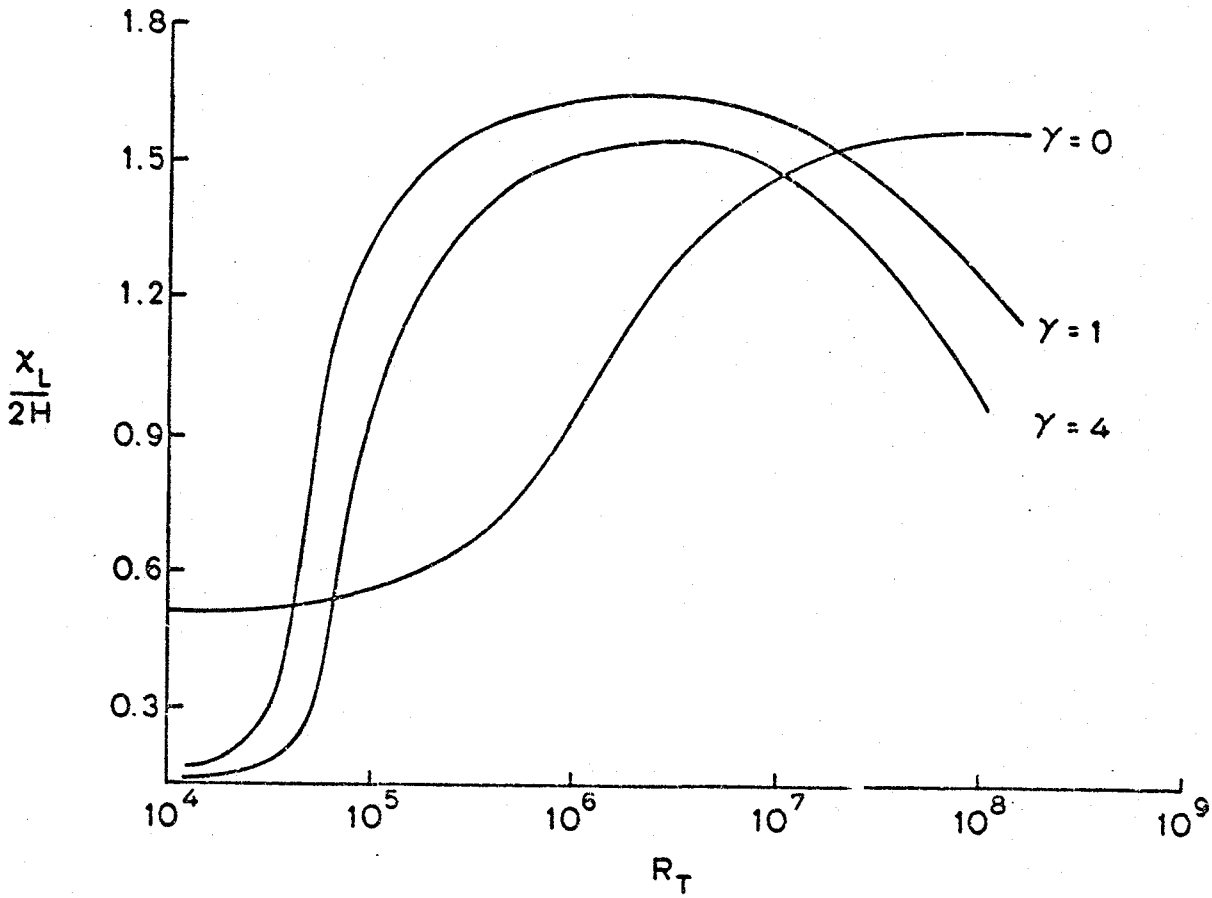


Figure 21. Optimal inlet lip length.

APPENDIX A

FORMULAE FOR THE INDUCED VELOCITY COMPONENTS

The panel method construction requires calculation of the velocity components at an arbitrary field point as induced by an isolated panel (say the j^{th} one) on the body surface. In the higher order method, the geometric features of the two neighboring panels (j-1 and j+1) must be included when considering the j^{th} panel. Figure A.1 shows the geometry and relevant nomenclature for three panels j-1, j, and j+1.

First three coordinate transformations are undertaken which take the global x,y system into local systems based on normal and tangential directions to the surface at the three control points j-1, j, j+1. Each transformation has the same form as illustrated for the j^{th} panel

$$ETA2 = (X-XI_j)\cos\alpha_j + (Y-YI_j)\sin\alpha_j, \qquad (A.1)$$

$$ZETA2 = -(X-XI_j)\sin\alpha_j + (Y-YI_j)\cos\alpha_j. \qquad (A.2)$$

The local coordinates affixed to the panel j-1 are labeled ETA1 and ZETA1, while those for the panel j+1 are labeled ETA3 and ZETA3.

Parabolic fit parameters for the source distribution are defined as

$$PD_j = -\frac{R_j}{Q_j P_j}, \tag{A.3}$$

$$PE_j = \frac{1}{Q_j}\left[\frac{R_j}{P_j} - \frac{P_j}{R_j}\right], \tag{A.4}$$

$$PF_j = \frac{P_j}{Q_j R_j}, \tag{A.5}$$

$$PG_j = \frac{2}{P_j Q_j}, \tag{A.6}$$

$$PH_j = -\frac{4}{P_j R_j}, \tag{A.7}$$

$$PPI_j = \frac{2}{R_j Q_j}, \tag{A.8}$$

where

$$Q_j = D_j + \frac{1}{2}\left[D_{j-1} + D_{j+1}\right], \tag{A.9}$$

$$P_j = D_j + D_{j-1}, \tag{A.10}$$

$$R_j = D_j + D_{j+1}. \tag{A.11}$$

Distances from the extremes of each panel to the field point are computed (for example for the j^{th} panel) according to

$$RP2^2 = (ETA2 + D_j/2)^2 + ZETA2^2, \tag{A.12}$$

$$RM2^2 = (ETA2 - D_j/2)^2 + ZETA2^2. \tag{A.13}$$

The distances for the j-1 panel are labeled RP1 and RM1, while those for the j+1 panel are labeled RP3 and RM3.

The velocity components at the field point are computed from

$$V_x = VE1\cos\alpha_{j-1} + VE2\cos\alpha_j + VE3\cos\alpha_{j+1} - VZ1\sin\alpha_{j-1} - VZ2\sin\alpha_j - VZ3\sin\alpha_{j+1}, \tag{A.14}$$

$$V_y = VE1\sin\alpha_{j-1} + VE2\sin\alpha_j + VE3\sin\alpha_{j+1} + VZ1\cos\alpha_{j-1} + VZ2\cos\alpha_j + VZ3\cos\alpha_{j+1}, \tag{A.15}$$

where

$$VE1 = VE11(PF_{j-1})D_{j-1} + VE21(PPI_{j-1})D^2_{j-1}, \tag{A.16}$$

$$VE2 = VE02 + VE12(PE_j)D_j + VEC2(C_j)D_j + VE22(PH_j + 2C^2_j)D^2_j, \quad (A.17)$$

$$VE3 = VE13(PD_{j+1})D_{j+1} + VE23(PG_{j+1})D^2_{j+1}, \quad (A.18)$$

$$VZ1 = VZ11(PF_{j-1})D_{j-1} + VZ21(PPI_{j-1})D^2_{j-1}, \quad (A.19)$$

$$VZ2 = VZ02 + VZ12(PE_j)D_j + VZC2(C_j)D_j + VZ22(PH_j + 2C^2_j)D^2_j, \quad (A.20)$$

$$VZ3 = VZ13(PD_{j+1})D_{j+1} + VZ23(PG_{j+1})D^2_{j+1}, \quad (A.21)$$

and where

$$VE01 = \ln(RP1/RM1), \quad (A.22)$$

$$VE02 = \ln(RP2/RM2), \quad (A.23)$$

$$VE03 = \ln(RP3/RM3), \quad (A.24)$$

$$VEC2 = \frac{1}{D_j}\left[-2(ETA2)VZ02 + 2(ZETA2)VE02 + \frac{ETA2(ZETA2)D^3_j}{RP2^2(RM2^2)}\right], \quad (A.25)$$

$$VE11 = \frac{1}{D_{j-1}}\left[ZETA1(VZ01) + ETA1(VE01) - 2D_{j-1}\right], \quad (A.26)$$

$$VE12 = \frac{1}{D_j}\left[ZETA2(VZ02) + ETA2(VE02) - 2D_j\right], \tag{A.27}$$

$$VE13 = \frac{1}{D_{j+1}}\left[ZETA3(VZ03) + ETA3(VE03) - 2D_{j+1}\right], \tag{A.28}$$

$$VE21 = \frac{1}{D^2_{j-1}}\left[2(ETA1)ZETA1(VZ01) + (ETA1^2-ZETA1^2)VE01 - 2(ETA1)D_{j-1}\right], \tag{A.29}$$

$$VE22 = \frac{1}{D^2_j}\left[2(ETA2)ZETA2(VZ02) + (ETA2^2-ZETA2^2)VE02 - 2(ETA1)D_j\right], \tag{A.30}$$

$$VE23 = \frac{1}{D^2_{j+1}}\left[2(ETA3)ZETA3(VZ03) + (ETA3^2-ZETA3^2)VE03 - 2(ETA3)D_{j+1}\right], \tag{A.31}$$

$$VZ01 = 2\tan^{-1}\left[\frac{ZETA1(D_{j-1})}{(ETA1^2+ZETA1^2) - (D_{j-1}/2)^2}\right], \tag{A.32}$$

$$VZ02 = 2\tan^{-1}\left[\frac{ZETA2(D_{j-1})}{(ETA2^2+ZETA2^2) - (D_j/2)^2}\right], \tag{A.33}$$

$$VZ03 = 2\tan^{-1}\left[\frac{ZETA3(D_{j+1})}{(ETA3^2+ZETA3^2) - (D_{j+1}/2)^2}\right], \tag{A.34}$$

$$VZC2 = \frac{1}{D_j}\left\{2(ZETA2)VZ02 + 2(ETA2)VE02 - 2(D_j)\left[1 + \frac{(ETA2^2+ZETA2^2)^2 - (ETA2^2-ZETA2^2)(D_j/2)^2}{RP2^2(RM2^2)}\right]\right\}, \tag{A.35}$$

$$VZ11 = \frac{1}{D_{j-1}}\left[ETA1(VZ01) - ZETA1(VE01)\right], \tag{A.36}$$

$$VZ12 = \frac{1}{D_j}\left[ETA2(VZC2) - ZETA2(VE02)\right], \tag{A.37}$$

$$VZ13 = \frac{1}{D_{j+1}}\left[ETA3(VZ03) - ETA3(VE03)\right], \tag{A.38}$$

$$VZ21 = \frac{1}{D^2_{j-1}}\left[-2(ETA1)ZETA1(VE01) + (ETA1^2-ZETA1^2)VZ01 + 2(ZETA1)D_{j-1}\right], \tag{A.39}$$

$$VZ22 = \frac{1}{D^2_j}\Big[-2(ETA2)ZETA2(VE02) + (ETA2^2-ZETA2^2)VZ02 + 2(ZETA2)D_j\Big], \tag{A.40}$$

$$VZ23 = \frac{1}{D^2_{j+1}}\Big[-2(ETA3)ZETA3(VE03) + (ETA3^2-ZETA3^2)VZ03 + 2(ZETA3)D_{j+1}\Big]. \tag{A.41}$$

The local surface curvatures C_j used in Eqs (A.17) and (A.20) are computed from a quadratic spline fit.

Now having established the induced velocity components, a set of simultaneous linear equations is constructed in order to determine the source strength for each panel. The general strategy is to compute the normal component of velocity induced by the j^{th} panel at the control point of the i^{th} panel, assuming unit source strength. The procedure is repeated for each control point, where account is made for the velocities due to the unit strength singularity on each individual panel. A matrix of areodynamic influence coefficients results where a generic element is defined as

$$W_{ij} = -V_{xij}\sin\alpha_i + V_{yij}\cos\alpha_i. \tag{A.42}$$

Now the inner product formed with the i^{th} row of this matrix

and the source strengths q_j results in the net induced velocity at the i^{th} control point. This net induced velocity is set equal to the component of the free-stream velocity normal to the i_{th} panel plus any prescribed transpiration velocity, to give a single constraint equation for the system. The same procedure is repeated for each panel, and a closed system of equations result which may be written in matrix form as

$$[W]\{q\} = \{B\}, \tag{A.43}$$

where the B_i are given as

$$B_i = u_\infty \sin\alpha_i - V_{Ni}, \tag{A.44}$$

and the V_{Ni} are the transpiration velocities defined to be positive for suction.

The system of equations given by Eq. (A.41) is of the order of the number of surface elements, and may be solved in a direct mode.

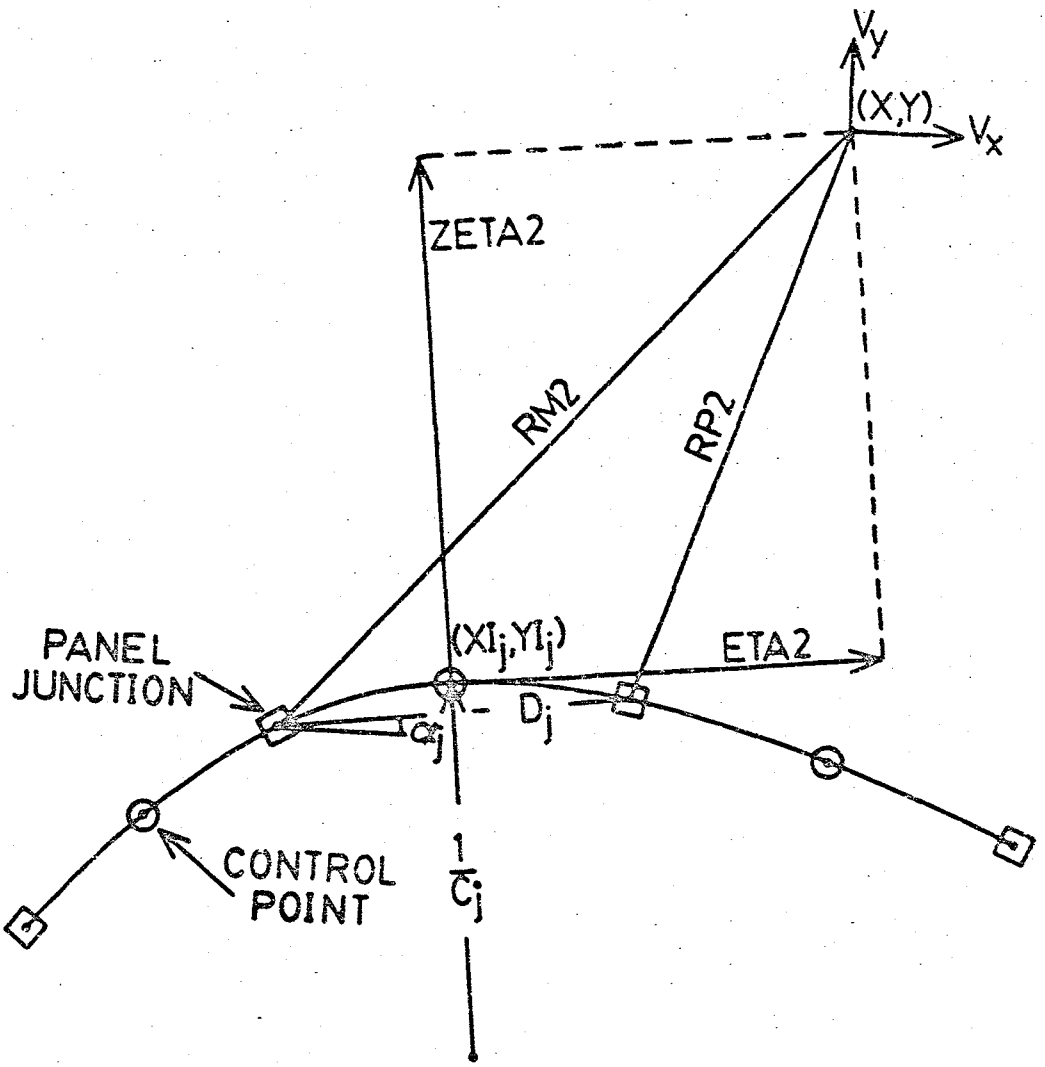


Figure A.1 Relevant nomenclature for the higher order panel method. Depicted are panels j-1, j, and j+1. Quantities are shown for the j^{th} panel only.

APPENDIX B

FORTRAN CODE

This appendix contains a source listing of the FORTRAN program which computes the thrust augmentation ratio and boundary layer behavior for an arbitrary two-dimensional ejector. The code is also currently on disk file at the NASA AMES Research Center in the directory NEP::FFF0:[koenigdg.lund.augment.auglib].

In addition to the subroutines listed here, the program must be linked to the double precision version of the IMSL mathematics library.

Documentation on the use of the program is provided in the form of comment statements included with the source listing.

```
      SUBROUTINE AUGMENT(RE,SCRIT,PHI)
C
C*************************************************************************
C                                                                        *
C     THIS ROUTINE WAS WRITTEN FOR THE JOINT INSTITUTE FOR AERONAUTICS AND *
C ACOUSTICS, STANFORD UNIVERSITY/NASA AMES RESEARCH CENTER, BY THOMAS LUND.  *
C                                                                        *
C LATEST REVISION 2 JULY 1985                                            *
C                                                                        *
C     SUBROUTINE AUGMENT COMPUTES THE THRUST AUGMENTATION RATIO OF A TWO- *
C DIMENSIONAL THRUST AUGMENTOR OF ARBITRARY GEOMETRIC CONFIGURATION.  THE  *
C CODE IS BASED ON A VISCOUS-INVISCID INTREACTION ALGORITHM IN WHICH THE  *
C INVISCID REGION IS COMPUTED USING A HIGHER ORDER PANEL METHOD, AND THE  *
C VISCOUS ZONE IS COMPUTED USING AN INTEGRAL METHOD.                      *
C     THIS CODE IS OF EVOLUTIONARY ORIGIN, AND CONSEQUENTLY CONTAINS ISOLATED *
C REGIONS OF POOR LOGIC STRUCTURE.  COMMENTS ARE INCLUDED FOR CLAIRITY    *
C WHEREVER POSSIBLE TO AIDE IN FOLLOWING THE PROGRAM STRUCTURE.  QUESTIONS *
C REGARDING THE USE OF THIS CODE MAY BE DIRECTED TO THE AUTHOR AT STANFORD *
C UNIVERSITY.                                                            *
C                                                                        *
C   *** SUBROUTINE DESCRIPTIONS ***                                      *
C                                                                        *
C  ANGLE  - COMPUTES THE GEOMETRIC FEATURES OF EACH OF THE SURFACE ELEMENTS *
C           USED IN THE PANEL METHOD.  THE PANEL LENGTHS, RADIUS OF CURVATURE,*
C           AND ORIENTATION IN SPACE ARE CALCULATED.                     *
C                                                                        *
C  AUGLYR - COMPUTES THE BOUNDARY LAYER DEVELOPMENT OVER THE INLET REGION OF *
C           THE DEVICE.                                                  *
C                                                                        *
C  BODGEN - GENERATES A SET OF BODY COORDINATES FOR SIMPLE SHROUD SHAPES USED *
C           IN THE OPTIMIZATION STUDY                                    *
C                                                                        *
C  CHANEL - COMPUTES THE VISCOUS VOLUTION IN THE CHANNEL DOWNSTREAM OF THE  *
C           STATION AT WHICH THE VISCOUS-INVISCID MATCHING IS ENDED.     *
C                                                                        *
C  COEF   - COMPUTES THE AERODYNAMIC INFLUENCE COEFFICIENTS FOR USE IN THE  *
C           PANEL METHOD                                                 *
C                                                                        *
C  DATIN  - READS THE DATA FILE WHICH CONTAINS THE BODY SURFACE COORDINATES  *
C           AND TRANSPIRATION VELOCITIES                                 *
C                                                                        *
C  FAPP   - COMPUTES THE LOCAL SKIN FRICTION COEFFICIENT AND LOCAL DISSIPATION*
C           COEFFICIENT FOR THE LAMINAR BOUNDARY LAYER EQUATIONS          *
C                                                                        *
C  FCN1   - COMPUTES THE DERIVATIVES OF THE JET PARAMETERS FOR USE IN MARCHING*
C           THE VISCOUS SOLUTION THROUGH THE VISCOUS-INVISCID REGION      *
C                                                                        *
C  FCN2   - COMPUTES THE DERIVATIVES OF THE JET PARAMETERS FOR USE IN MARCHING*
C           THE VISCOUS SOLUTION IN THE MIXING CHANEL DOWNSTREAM OF THE REGION*
C           OF VISCOUS-INVISCID MATCHING                                 *
C                                                                        *
C  FCNL   - COMPUTES THE DERIVATIVES OF THE LAMINAR BOUNDARY PARAMETERS FOR  *
```

```
C          USE IN MARCHING OF THE BOUNDARY LAYER EQUATIONS                *
C                                                                         *
C  FCNT    - COMPUTES THE DERIVATIVES OF THE TURBULENT BOUNDARY LAYER      *
C            PARAMETERS FOR USE IN MARCHING OF THE BOUNDARY LAYER EQUATIONS*
C                                                                         *
C  FIX     - UPDATES THE DATA FILE CONTAINING THE TRANSPIRATION VELOCITIES *
C            ONCE CONVERGENCE IS OBTAINED.                                *
C                                                                         *
C  FLDVEL  - COMPUTES THE VELOCITY COMPONENTS AT AN ARBITRARY LOCATION IN THE *
C            INVISCID FIELD FROM THE PANEL SOLUTION                       *
C                                                                         *
C  INTRPV  - INTERPOLATES THE FUNCTION AND DERIVATIVE VALUES FROM A CUBIC  *
C            SPLINE FIT PRODUCED BY IMSL ROUTINE ICSCCU                   *
C                                                                         *
C  JET     - MARCHES THE VISCOUS SOLUTION THROUGH THE VISCOUS-INVISCID REQION *
C                                                                         *
C  LINTRP  - INTERPOLATES THE FUNCTION AND DERIVATIVE VALUES FROM THE LINEAR *
C            SPLINE FIT ROUTINE LNSPLN                                    *
C                                                                         *
C  LNSPLN  - COMPUTES THE PARAMETERS FOR A LINEAR SPLINE FIT              *
C                                                                         *
C  MATRIX  - COMPUTES THE COPLING COEFFICIENT MATRIX AND RIGHT HAND SIDE OF THE*
C            VISCOUS SOLUTION SYSTEM                                      *
C                                                                         *
C  PANVLC  - COMPUTES THE VELOCITY VALUES ALONG THE JET BOUNDARY FROM THE  *
C            INVISCID SOLUTION                                            *
C                                                                         *
C  PARMIN  - READS A DATA FILE CONTAINING GEOMETRIC PARAMETERS ASSOCIATED WITH *
C            THE SHROUD                                                   *
C                                                                         *
C  PERFRM  - COMPUTES THE THRUST AUGMENTATION RATIO FROM THE CONVERGED     *
C            SOLUTION                                                     *
C                                                                         *
C  RK2     - PERFORMS NUMERICAL INTEGRATION ACCORDING TO THE SECOND ORDER RUNGE*
C            KUTTA METHOD                                                 *
C                                                                         *
C  SIMQ    - SOLVES A SET OF SIMULTANEOUS LINEAR EQUATIONS                *
C                                                                         *
C  SIZE    - READS THE DATA SET CONTAINING THE BODY COORDINATES IN ORDER TO *
C            ASSESS ITS SIZE                                              *
C                                                                         *
C  SOLVE3  - PERFORMS THE SIMULTANEOUS LINEAR EQUATION SOLUTION NECESSARY TO *
C            DETERMINE THE DERIVATIVES OF THE JET PARAMETERS              *
C                                                                         *
C  STREN   - COMPUTES THE SINGULARITY INTENSITY DISTRIBUTION FOR USE IN THE *
C            PANEL METHOD                                                 *
C                                                                         *
C  SURFVEL- COMPUTES THE VELOCITY AT THE SURFACE OF THE SHROUD FOR USE IN THE *
C            BOUNDARY LAYER CALCULATION                                   *
C                                                                         *
C   *** PARAMETER DESCRIPTION ***                                         *
C                                                                         *
```

```
C    INPUT:                                                              *
C  RE    - THRUST BASED REYNOLDS NUMBER                                  *
C                                                                        *
C    OUTPUT:                                                             *
C  SCRIT - SURFACE COORDINATE AT WHICH SEPARATION OCCURS.  IF THE BOUNDARY *
C          LAYER REMAINS ATTACHED SCRIT = 1.                             *
C  PHI   - THRUST AUGMENTATION RATIO                                     *
C                                                                        *
C   *** DATA FILES ***                                                   *
C                                                                        *
C    INPUT:                                                              *
C  BODY  - CONTAINS THE COORDINATES OF THE SHROUD SURFACE AS WELL AS THE  *
C          TRANSPIRATION VELOCITY REQUIRED OVER EACH PANEL.  THE FORMAT  *
C          IS X, Y, VN IN A FIELD OF 3F10.4                              *
C  PARAM - CONTAINS SCALE INFORMATION FOR THE BODY GEOMETRY.  THE PARAMETERS *
C          ARE XO,XC,XEXIT,NJS,NJF,NLS,NLF,VO,BETA,U1O, IN A FIELD OF    *
C          3F10.5,4I4,/,2F10.4,/,F10.4.  XO IS THE NOZZLE LOCATION NORMALIZED *
C          BY THE CHANNEL HALF-WIDTH, XC IS THE CONTROL STATION LOCATION *
C          NORMALIZED BY THE CHANNEL HALF-WIDTH, XEND IS THE SHROUD END  *
C          LOCATION NORMALIZED BY THE CHANNEL HALF-WIDTH, NJS IS THE PANEL *
C          NUMBER AT WHICH THE JET STARTS, NJF IS THE PANEL NUMBER AT WHICH *
C          THE CONTROL STATION ENDS, NLS IS THE PANEL NUMBER AT WHICH THE THE *
C          INLET LIP BEGINS, NLF IS THE PANEL NUMBER AT WHICH THE INLET LIP *
C          ENDS, VO IS THE FREE-STREAM VELOCITY NORMALIZED BY THE VELOCITY AT *
C          THE CONTROL STATION, BETA IS THE ANGLE OF ATTACK, AND U1O IS THE *
C          INITIAL CENTERLINE VELOCITY OF THE JET.  THE PANELS ARE NUMBERED *
C          SEQUENTIALLY STARTING FROM THE PANEL FURTHEST UPSTREAM         *
C  VEL   - INITIAL JET CENTERLINE VELOCITY AS WELL AS THE JET ENTRAINMENT *
C          VELOCITY DISTRIBUTION.  THE FORMAT IS U1O IN A FIELD OF F20.4 *
C          AND THEN THE VORMAL VELOCITIES AT THE PANELS WHICH REPRESENT THE *
C          JET IN A FIELD OF F10.4.  VEL IS UPDATED AT EACH ITERATION.    *
C                                                                        *
C   OUTPUT:                                                             *
C  DIAG  - CONTAINS ERROR MESSAGES.  DIAG SHOULD BE CONSULTED AT THE     *
C          COMPLETION OF EACH RUN TO CHECK FOR ERROR CONDITIONS.  IN ADDITION *
C          DIAG CONTAINS DIAGNOSTIC PRINTOUTS IF THE LOGICAL VARIABLE DUMP1 *
C          IS SET TO TRUE                                                *
C  PERFORM-CONTAINS THE AUGMENTATION RATIO AS COMPUTED BOTH BY INTEGRATION OF *
C          THE SURFACE PRESSURE AND A BLASIUS CONTROL VOLUME ANALYSIS.  IN *
C          ADDITION A SUMMARY OF THE BOUNDARY LAYER CALCULATION IS PROVIDED *
C                                                                        *
C   *** LINKING ***                                                     *
C                                                                        *
C   THIS ROUTINE MUST BE LINKED TO THE AUGLIB LIBRARY AS WELL AS THE DOUBLE *
C   PRECISION VERSION OF THE IMSL LIBRARY.                              *
C                                                                        *
C   *** PRECISION ***                                                   *
C                                                                        *
C   ALL PARAMETERS AND INTERNAL VARIABLES ARE DOUBLE PRECISION.         *
C                                                                        *
C   *** ENVIRONMENT ***                                                 *
C                                                                        *
```

```
C   VAX/11-780 OR VAX/11-785                                                    *
C                                                                               *
C*******************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XJ(250,2),XI(250,2),VN(250),ALPHA(250),
     &          D(250),Q(250),W(250,250),P(250,250),VLN(50),
     &          XS(250),VS(250),SC(100),UEXT(100)
      COMMON /UNIF/ VO
      COMMON /AREA4/ PATM
      COMMON /AREA10/ XC
      COMMON /AREA12/ XEXIT
      COMMON /DUMP/ DUMP1
      LOGICAL DUMP1,STAG,DUMP,SEP
      REWIND 1
      REWIND 2
C
C        *** DUMP CONTROLS DIAGNOSTIC PRINTING ***
C
      DUMP1=.FALSE.
C
C        *** TOL1 IS THE CONVERGENCE TOLLERENCE FOR THE VISCOUS-INVISCID ***
C        *** MATCHING, TOL2 IS THE CONVERGENCE TOLLERENCE FOR THE EXIT   ***
C        *** PRESSURE MATCHING                                           ***
C
      TOL1=5.0D-4
      TOL2=5.0D-4
C
      CALL SIZE(M)
      CALL DATIN(XJ,VN,M)
C
C        *** N IS THE NUMBER OF SURFACE ELEMENTS ***
C
      N=M-1
C
      CALL PARMIN(U10,VO,BETA)
      CALL ANGLE(XJ,N,XI,ALPHA,D)
C
C        *** IF L=1 THE AERODYNAMIC INFLUENCE COEFFICIENTS ARE COMPUTED,  ***
C        *** IF L=0 THE COEFFICIENTS FROM THE PREVIOUS ITERATION ARE USED ***
C
      L=1
C
      DO 50 J=1,10
         DO 10 I=1,10
            IF(J.NE.1.OR.I.NE.1) L=0
            CALL STREN(XI,Q,VN,ALPHA,D,W,N,P,VO,BETA,L)
            CALL PANVLC(XI,ALPHA,D,Q,N)
            CALL JET(U10,VN,RES)
            IF(RES.LT.TOL1) GOTO 20
10       CONTINUE
20       CALL CHANEL(PEXIT)
```

```
C
C           *** COMPUTE THE STATIC PRESSURE IN THE INVISCID FIELD AT THE ***
C           *** SHROUD EXIT                                               ***
C
          PINV=PATM-0.5*VO*VO
C
          R=(PINV-PEXIT)
          WW=4.0D0+0.5D0*U1O
          UCOR=WW*R
C
C           *** SET BOUNDS ON CORRECTION FACTOR IN ORDER TO AVOID ***
C           *** INSTABILITIES ***
C
          IF(UCOR.GT.5.0) UCOR=5.0D0
          IF(UCOR.LT.-5.0) UCOR=-5.0D0
C
          U1O=U1O+UCOR
          WRITE(5,40) PINV,PEXIT,U1O
40        FORMAT(/,' IN AUGMENT PINV = ',F12.8,' PEXIT = ',F12.8,
     &             ' U1O = ',F12.8,/)
          IF(DABS(R).LT.TOL2) GOTO 90
50      CONTINUE
90      CONTINUE
        CALL SURFVEL(XI,ALPHA,D,Q,N,SC,UEXT,NEXT,XLEN,STAG)
        DUMP=.FALSE.
        NSTEP=20
        CALL AUGLYR(SC,UEXT,NEXT,RE,STAG,DUMP,NSTEP,SEP,SCRIT)
        IF(SEP) THEN
          WRITE(4,100)
100       FORMAT(' SEPARATED BOUNDARY LAYER',/)
        ELSE
          WRITE(4,110)
110       FORMAT(' NO SEPARATION',/)
        END IF
        CALL PERFRM(XI,ALPHA,D,Q,N,U1O,PHI)
        RETURN
        END
```

```
      SUBROUTINE ANGLE(XP,N,XI,ALPHA,D)
C
C*************************************************************************
C                                                                        *
C     SUBROUTINE ANGLE COMPUTES THE SURFACE ELEMENT LENGTH, RADIUS OF    *
C CURVATURE, AND ORIENTIATION IN SPACE FOR USE IN THE PANEL METHOD.      *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  XP     - SURFACE COORDINATES STORED AS X,Y PAIRS IN A (N-1) X 2 MATRIX *
C  N      - NUMBER OF SURFACE ELEMENTS                                   *
C                                                                        *
C     OUTPUT:                                                            *
C  XI     - COORDINATES OF THE CONTROL POINTS STORED AS XY PAIRS IN A N X 2 *
C           MATRIX                                                       *
C  ALPHA  - VECTOR OF INVERSE TANGENTS OF THE SLOPE OF EACH PANEL        *
C           (ORIENTATION ANGLE)                                          *
C  D      - VECTOR CONTAINING THE LENGTHS OF EACH PANE                   *
C                                                                        *
C  SENT DIRECTLY TO SUBROUTINE COEF IN A COMMON STATEMENT ARE THE PARABOLA *
C  FIT PARAMETER VECTORS PD,PE,PF,PF,PH,PPI, AS WELL AS THE VECTOR CONTAINING *
C  THE CURVATURE OF EACH PANEL,C, AND A LOGICAL VARIABLE PERDT SET TO TRUE *
C  FOR A BODY WITH PERIODIC GEOMETRY                                     *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XP(N+1,2),XI(N,2),ALPHA(N),D(N)
      COMMON /DUMP/ DUMP1
      LOGICAL DUMP1
      COMMON /ANGLE1/ PD(100),PE(100),PF(100),PG(100),PH(100),
     &                PPI(100),C(100)
      COMMON /ANGLE2/ PERDT
      LOGICAL PERDT
      PI=3.141593
C
C        *** CHECK FOR PERIODIC GEOMETRY ***
C
      XDIFF=XP(M,1)-XP(1,1)
      YDIFF=XP(M,2)-XP(1,2)
      IF(DABS(XDIFF).LT.1.E-3.AND.DABS(YDIFF).LT.1.E-3) PERDT=.TRUE.
C
C        *** INDIVIDUAL PANEL ORIENTATION ANGLE AND LENGTH ***
C
      DO 10 I=1,N
         DX=XP((I+1),1)-XP(I,1)
         DY=XP((I+1),2)-XP(I,2)
         IF(DABS(DX).LT.1.E-5.AND.DY.GT.0.0) GOTO 1
         IF(DABS(DX).LT.1.E-5.AND.DY.LT.0.0) GOTO 2
         IF(DABS(DY).LT.1.E-5.AND.DX.LT.0.0) GOTO 3
         ALPHA(I)=DATAN(DY/DX)
```

```
          IF(DY.LT.0.0.AND.DX.LT.0.0) ALPHA(I)=ALPHA(I)-PI
          IF(DY.GT.0.0.AND.DX.LT.0.0) ALPHA(I)=ALPHA(I)+PI
          GOTO 4
1         ALPHA(I)=PI/2.0
          GOTO 4
2         ALPHA(I)=-PI/2.0
          GOTO 4
3         ALPHA(I)=PI
4         D(I)=DSQRT(DX**2+DY**2)
10     CONTINUE
C
C         *** COMPUTE THE CURVATURE OF THE FIRST PANEL ***
C
       IF(.NOT.PERDT) GOTO 12
       L1=N-1
       L2=N
       L3=1
       DENOM=(XP(L1,1)-XP(L2,1))*(XP(L2,2)-XP(L3,2))
      &     -(XP(L2,1)-XP(L3,1))*(XP(L1,2)-XP(L2,2))
       IF(DABS(DENOM).LT.1.E-6) GOTO 12
       XO=.5*((XP(L1,1)**2-XP(L2,1)**2+XP(L1,2)**2-XP(L2,2)**2)
      &       *(XP(L2,2)-XP(L3,2))
      &      -(XP(L2,1)**2-XP(L3,1)**2+XP(L2,2)**2-XP(L3,2)**2)
      &       *(XP(L1,2)-XP(L2,2)))/DENOM
       YO=.5*((XP(L2,1)**2-XP(L3,1)**2+XP(L2,2)**2-XP(L3,2)**2)
      &       *(XP(L1,1)-XP(L2,1))
      &      -(XP(L1,1)**2-XP(L2,1)**2+XP(L1,2)**2-XP(L2,2)**2)
      &       *(XP(L2,1)-XP(L3,1)))/DENOM
       C1=1./DSQRT((XP(L2,1)-XO)**2+(XP(L2,2)-YO)**2)
       IF(ALPHA(L1).LT.0..AND.ALPHA(L2).GT.0.) GOTO 13
       IF(ALPHA(L2).GT.ALPHA(L1)) C1=-C1
       GOTO 13
12     C1=0.
C
C         *** COMPUTE THE SURFACE CURVATURES AND PARABOLA FIT PARAMETERS ***
C
13     DO 100 I=1,N
          IF(.NOT.PERDT.AND.(I.EQ.1.OR.I.EQ.N)) GOTO 50
          L1=I-1
          L2=I
          L3=I+1
          IF(I.EQ.1) L1=N
          IF(I.EQ.N) L3=1
          DENOM=(XP(L1,1)-XP(L2,1))*(XP(L2,2)-XP(L3,2))
      &        -(XP(L2,1)-XP(L3,1))*(XP(L1,2)-XP(L2,2))
          IF(DABS(DENOM).LT.1.E-6) GOTO 20
          XO=.5*((XP(L1,1)**2-XP(L2,1)**2+XP(L1,2)**2-XP(L2,2)**2)
      &          *(XP(L2,2)-XP(L3,2))
      &         -(XP(L2,1)**2-XP(L3,1)**2+XP(L2,2)**2-XP(L3,2)**2)
      &          *(XP(L1,2)-XP(L2,2)))/DENOM
          YO=.5*((XP(L2,1)**2-XP(L3,1)**2+XP(L2,2)**2-XP(L3,2)**2)
      &          *(XP(L1,1)-XP(L2,1))
```

```
     &          -(XP(L1,1)**2-XP(L2,1)**2+XP(L1,2)**2-XP(L2,2)**2)
     &           *(XP(L2,1)-XP(L3,1)))/DENOM
           C2=1./DSQRT((XP(L2,1)-XO)**2+(XP(L2,2)-YO)**2)
           IF(ALPHA(L1).LT.0..AND.ALPHA(L2).GT.0.) GOTO 30
           IF(ALPHA(L2).GT.ALPHA(L1)) C2=-C2
           GOTO 30
20         C2=0.
30         C(L1)=.5*(C1+C2)
C          IF(C1.LT.0..AND.C2.GT.0.) C(L1)=0.
C          IF(C1.GT.0..AND.C2.LT.0.) C(L1)=0.
           C1=C2
           Q=D(L2)+.5*(D(L1)+D(L2))
           R=D(L2)+D(L3)
           P=D(L2)+D(L1)
           PD(L2)=-R/Q/P
           PE(L2)=(R/P-P/R)/Q
           PF(L2)=P/R/Q
           PG(L2)=2./P/Q
           PH(L2)=-4./P/R
           PPI(L2)=2./R/Q
           IF(ABS(C(L1)).LT.1.E-3) GOTO 35
           DEL=1./DABS(C(L1))-DSQRT((1./C(L1))**2-(D(L1)/2.)**2)
           GOTO 38
35         DEL=0.
38         IF(C(L1).LT.0.) GOTO 40
           XI(L1,1)=.5*(XP(L1,1)+XP(L2,1))-DEL*DSIN(ALPHA(L1))
           XI(L1,2)=.5*(XP(L1,2)+XP(L2,2))+DEL*DCOS(ALPHA(L1))
           GOTO 50
40         XI(L1,1)=.5*(XP(L1,1)+XP(L2,1))+DEL*DSIN(ALPHA(L1))
           XI(L1,2)=.5*(XP(L1,2)+XP(L2,2))-DEL*DCOS(ALPHA(L1))
50         CONTINUE
100    CONTINUE
       IF(PERDT) GOTO 110
       XI(N-1,1)=.5*(XP(N-1,1)+XP(N,1))
       XI(N-1,2)=.5*(XP(N-1,2)+XP(N,2))
       XI(N,1)=.5*(XP(N,1)+XP(N+1,1))
       XI(N,2)=.5*(XP(N,2)+XP(N+1,2))
       IF(.NOT.DUMP1) GOTO 200
110    WRITE(3,120)
120    FORMAT(4X,'XJ',2X,'YJ',6X,'XI',2X,'YI',4X,'ALPHA J',1X,
     &       'LENGTH J',' C',//)
       K=N+1
       DO 140 I=1,K
          IF(I.EQ.(N+1)) GOTO 132
          WRITE(3,130) XP(I,1),XP(I,2),XI(I,1),XI(I,2),ALPHA(I),D(I),
     &                 C(I)
130       FORMAT(7F10.4)
          GOTO 140
132       WRITE(3,135) XP(I,1),XP(I,2)
135       FORMAT(2F10.4)
140    CONTINUE
200    CONTINUE
```

```
RETURN
END
```

```
      SUBROUTINE AUGLYR(X,V,N,R,STAG,DUMP,NSTEP,SL.,SCRIT)
C
C*****************************************. ********************
C    THIS CODE WAS WRITTEN FOR THE JOINT INSTITUTE FOR AERONAUTICS    *
C AND ACOUSTICS BY THOMAS LUND.  LATEST REVISION 8 SEPT. 1984.        *
C                                                                     *
C    THIS SUBROUTINE COMPUTES LAMINAR AND TURBULENT BOUNDARY LAYER    *
C DEVELOPMENT, GIVEN AN EXTERNAL VELOCITY DISTRIBUTION. THE EQUATIONS *
C SOLVED HERE ARE BASED ON AN INTEG'.L FORMULATION OF T E BOUNDARY    *
C LAYER EQUATIONS.  IN THE TURBULENT CASE, THE NORMAL TURBULENT       *
C STRESSES ARE NEGL ' ED IN COMPARISON WITH THE TURBULENT SHEARING    *
C STRESS.  THE TURBULENT BOUNDARY LAYER EQUATIONS USED HERE ARE FOUND *
C IN SCHLICHTING (7TH ED) P. 676, EQS. (22.7a,b), (22.8a,b), AND      *
C FIG 22.7                                                            *
C    THE VELOCITY DISTRIBUTION DESCRIBED NEED NOT HAVE A              *
C STAGNATION POINT (SEE DESCRIPTION OF PARAMETER STAG).  THE CODE     *
C ASSUMES THAT ALL BOUNDARY LAYERS HAVE A LAMINAR ORIGIN.  TO AVOID   *
C SINGULRITIES AT THE ORIGIN, INITIAL VALUES OF THE VARIOUS CHARAC-   *
C TERISTIC THICKNESSES AND SHAPE FACTORS ARE ASSUMED BY COMPUTING     *
C THESE QUANTITIES AT A SMALL DSTANCE FROM THE ORIGIN USING ANALYTIC  *
C EXPRESSIONS FOR A LAMINAR BOUNDARY LAYER IN A ZERO-PRESSURE GRAD-   *
C IENT OUTER STREAM.                                                  *
C    THE LAMINAR BOUNDARY LAYER EQUATIONS ARE MARCHED AWAY FROM THE   *
C INITIAL DATA UNTIL THE END OF THE BODY IS REACHED, OR EITHER TRANS-  *
C ITION TO TURBULENT FLOW, OR LAMINAR SEPARATION IS DETECTED.  IF     *
C LAMINAR SEPARATION IS DETECTED, THE CODE HALTS AT THE POINT OF      *
C SEPARATION.  IF TRANSITION IS DETECTED, THE CODE SWITCHES TO THE     *
C TURBULENT BOUNDARY LAYER EQUATIONS, AND CONTINUES TO MARCH UNTIL     *
C EITHER THE END OF THE BODY IS REACHED, OR TURBULENT SEPARATION IS    *
C DETECTED.  IF TURBULENT SPARATION IS DETECTED, THE CODE HALTS AT     *
C THE POINT OF SEPARATION.                                            *
C    IF OUTPUT IS SPECIFIED (SEE DESCRIPTION OF PARAMETERS DUMP AND   *
C NSTEP) THE FOLLOWING DATA WILL BE PRINTED TO UNIT 3 FOR SPECIFIED   *
C VALUES OF THE SURFACE COORDINATE: SHAPE FACTOR H32, DISPLACEMENT    *
C THICKNESS, MOMENTUM THICKNESS, ENERGY THICKNESS, AND LOCAL SKIN     *
C FRICTION COEFFICIENT                                                *
C                                                                     *
C **PARAMETER DESCRIPTIONS**                                          *
C                                                                     *
C    INPUT:                                                           *
C       X - VECTOR OF LENGTH N CONTAINING THE VALUES OF THE SURFACE   *
C           COORDINATE AT WHICH EXTERNAL VELOCITIES ARE GIVEN.  THE   *
C           SURFACE COORDINATES MUST START FROM ZERO (X(1)=0.0), BE   *
C           IN INCREASING ORDER, AND BE NORMALIZED BY THE SURFACE     *
C           LENGTH (X(N)=1.0).                                        *
C       V - VECTOR OF LENGTH N CONTAINING THE VALUES OF THE EXTERNAL  *
C           VELOCITY WHICH CORRESPOND TO THE SURFACE CORRDINATES      *
C           CONTAINED IN VECTOR X.  THE EXTERNAL VELOCITY MUST BE     *
C           NORMALIZED BY THE CHARACTERISTIC VELOCITY OF THE PROBLEM  *
C       N - NUMBER OF SURFACE COORDINATE AND EXTERNAL VELOCITY DATA   *
C           PAIRS (LENGTH OF VECTORS X AND V).                        *
C       R - GLOBAL REYNOLDS NUMBER DEFINED AS R=Uc*L/vis, WHERE Uc    *
```

```
C            IS THE CHARACTERISTIC VELOCITY OF THE PROBLEM, L IS THE  *
C            SURFACE LENGTH, AND vis IS THE COEFFICIENT OF KINEMATIC  *
C            VISCOSITY.                                               *
C     STAG - LOGICAL VARIABLE USED TO SPECIFY WHETHER OR NOT A        *
C            STAGNATION POINT EXISTS.  IF STAG IS SET TO .TRUE. A     *
C            STAGNATION POINT IS ASSUMED, IF SET TO .FALSE. NO STAG-  *
C            NATION POINT IS ASSUMED.                                 *
C     DUMP - LOGICAL VARIABLE USED TO SPECIFY WHETHER OR NOT OUTPUT   *
C            IS TO BE GENERATED.  IF DUMP IS SET TO .TRUE. OUTPUT IS  *
C            SENT TO UNIT 3, IF DUMP IS SET TO .FALSE. NO OUTPUT IS   *
C            GENERATED.                                               *
C    NSTEP - INTEGER VALUE USED TO SPECIFY THE NUMBER OF STATIONS AT  *
C            WHICH OUTPUT IS TO BE GENERATED.  THE STATIONS ARE EQUI- *
C            SPACED.                                                  *
C                                                                     *
C    OUTPUT:                                                          *
C      SEP - LOGICAL VARIABLE USED TO INDICATE A SEPARATED BOUNDARY   *
C            LAYER.  IF EITHER LAMINAR OR TURBULENT SEPARATION IS     *
C            DETECTED, SEP IS SET TO .TRUE.  IF NO SEPARATION IS      *
C            DETECTED, SEP IS SET TO .FALSE.                          *
C   SCRIT  - DIMENSIONLESS SURFACE COORDINATE AT WHICH THE BOUNDARY   *
C            LAYER HAS SEPARATED.  IF NO SEPARATION OCCURES SCRIT = 1 *
C            INDICATING THE END OF THE BODY                           *
C                                                                     *
C*********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      EXTERNAL FCNL,FCNT
      DIMENSION X(100),V(100),C(24),W(2,9),Y(2),YD(2)
      COMMON /BLCVEL/ XX(100),VV(100),RR
      COMMON /BLCSPLN/ SPLN(100),NN
      COMMON /AREA10/ XC
      COMMON /AREA12/ XEXIT
      LOGICAL STAG,LMNR,SEP,DUMP
C
C        *** FUNCTION F1 RETURNS H12 GIVEN H32 ***
C
      F1(H32)=H32/(3.0D0*H32-4.0D0)
C
C        *** FUNCTION WSHR RETURNS THE LOCAL TURBULENT SKIN FRICTION   ***
C        *** COEFFICIENT DIVIDED BY 2, GIVEN THE SHAPE FACTOR H12      ***
C        *** AND THE REYNOLDS NUMBER BASED ON MOMENTUM THEICKNESS RD2. ***
C
      WSHR(H12,RD2)=0.0245D0*(1.0D0-2.0959D0*DLOG10(H12))**1.705D0
     &              /RD2**0.268D0
C
C        *** IN ORDER TO PASS SUBROUTINE ARGUMENTS IN COMMON AS WELL, ***
C        *** WE HAVE TO DEFINE REDUNDANT ARRAYS XX AND VV, AND        ***
C        *** CONSTANTS RR AND NN                                      ***
C
      DO 1 I=1,N
         XX(I)=X(I)
```


C-2

```
         VV(I)=V(I)
1      CONTINUE
       RR=R
       NN=N
       NF=N-1
C
C         *** SPLINE FIT THE VELOCITY DATA USING AUGLIB ROUTINE LNSPLN ***
C
       CALL LNSPLN(X,V,N,SPLN,IER)
       IF(IER.NE.0) THEN
          WRITE(3,642) IER
642       FORMAT(' IN SUGROUTINE AUGLYR LNSLPLN RETURNED WITH THE ERROR'
     &           'CONDITION IER =',I5)
          STOP
       END IF
C
C         *** DS IS THE INTEGRATION STEP SIZE, SI IS THE INITIAL CONDITION ***
C         *** STATION                                                      ***
C
       DS=5.0D-4
       SI=0.05D0
C
C       *** DEFINE INTEGRATION DO LOOP UPPER LIMIT ***
C
       IEND=NINT((1.0D0-SI)/DS)
C
C         *** DEFINE THE NUMBER OF INTEGRATION STEPS BETWEEN PRINTOUTS ***
C
       NPRINT=1.0D0/(DFLOAT(NSTEP)*DS)
       CALL LINTRP(SI,X,V,SPLN,N,VI,VID,IER)
       IF(IER.EQ.1) THEN
          WRITE(3,71) SI
71        FORMAT(' IN AUGLYR LINTRP RETURNED WITH AN ERROR FLAG',/,
     &           ' X HAD THE VALUE',F10.6,' ON ENTRY')
          STOP
       END IF
C
C         *** CHECK FOR STAGNATION PIONT, AND SET INTITIAL VALUES ***
C         *** ACCORDINGLY                                         ***
C
       IF(STAG) THEN
          IF(DUMP) WRITE(3,3)
3         FORMAT(10X,' STAGNATION POINT ')
          H32=1.61998D0
          D2=0.29004D0/DSQRT(R*VID)
          D3=H32*D2
       ELSE
          IF(DUMP) WRITE(3,5)
5         FORMAT(10X,' NO STAGNATION POINT ')
          H32=1.57258
          D2=0.66411*DSQRT(SI/(R*VI))
          D3=H32*D2
```

```
      END IF
      IF(DUMP) THEN
         WRITE(3,6) R,H32,D2,D3,SI,VI
6        FORMAT(/,10X,' REYNOLDS NUMBER = ',E10.4,//,10X,
     &           ' INITIAL VALUES',//,10X,' H32 = ',E10.4,/,10X,
     &           ' MOMENTUM THICKNESS = ',E10.4,/,10X,
     &           ' ENERGY THICKNESS = ',E10.4,/,10X,' ABSCISSA = ',
     &          E10.4,/,10X,' VELOCITY = ',E10.4,/)
         WRITE(3,7)
7        FORMAT('       X     VELOCITY')
         DO 9 I=1,N
            WRITE(3,8) X(I),V(I)
8           FORMAT(2F10.4)
9        CONTINUE
      END IF
      RD2=R*VI*D2
C
C        *** COMPUTE INITIAL LAMINAR SKIN FRICTION ***
C
      CALL FAPP(H32,H12,EPS,D,KAPS)
      CFL=EPS/RD2
      CD=2.0D0*CFL*VI*VI
      D1=H12*D2
      IF(DUMP) WRITE(3,10)
10    FORMAT(//,7X,'X',11X,'H32',10X,'D1',11X,'D2',11X,'D3',11X,'CD',/)
      DUM=0.058D0
      IF(DUMP) WRITE(3,20) SI,H32,D1,D2,D3,CD,DUM
20    FORMAT(7E11.4)
C
C        *** INITIALIZE PARAMETERS FOR THE INTEGRATION LOOP ***
C
      LMNR=.TRUE.
      SEP=.FALSE.
      S=SI
      Y(1)=D2
      Y(2)=D3
      RMARGN=1.0D0
      R2=0.058D0
      R3=R2
      R4=R2
      R5=R2
      R6=R2
      NE=2
      TOL=0.001D0
      IND=1
      K=0
C
C        *** ENTER THE INTEGRATION LOOP ***
C
      DO 50 I=1,IEND
         K=K+1
         S=S+DS
```

```
C
C          *** INTEGRATE EITHER THE LAMINAR OR TURBULENT BOUNDARY LAYER ***
C          *** EQUATIONS DEPENDING ON THE VALUE OF LMNR USING RK2       ***
C
      IF(LMNR) THEN
         CALL RK2(NE,FCNL,SI,Y,S)
      ELSE
         CALL RK2(NE,FCNT,SI,Y,S)
      END IF
      D2=Y(1)
      D3=Y(2)
      H32=D3/D2
      CALL LINTRP(S,X,V,SPLN,N,VS,VSD,IER)
      IF(IER.EQ.1) THEN
         WRITE(3,72) S
72       FORMAT(' IN AUGLYR LINTRP RETURNED WITH AN ERROR FLAG',/,
     &          ' X HAD THE VALUE',F10.6,' ON ENTRY')
         STOP
      END IF
      RD2=R*VS*D2
C
C          *** IF STILL LAMINAR, CHECK FOR TRANSITION ***
C
      IF(LMNR) THEN
         IF((H32-(DLOG(RD2)+46.78D0)/34.2D0).LE.0.0) THEN
            STRANS=S
            LMNR=.FALSE.
         END IF
      END IF
      IF(LMNR) THEN
C
C             *** CHECK FOR LAMINAR SEPARATION IGNORE SEPARATION WHICH IS ***
C             *** IS PREDICTED DUE TO NOISY VELOCITY DISTRIBUTION BEFORE  ***
C             *** THE NOSE                                                ***
C
         IF(H32.LT.1.51509.AND.S.GT.0.7) GOTO 70
C
C             *** COMPUTE LAMINAR SKIN FRICTION ***
C
         CALL FAPP(H32,H12,EPS,D,KAPS)
         CFL=EPS/RD2
         CF=2.0D0*CFL*VS*VS
      ELSE
C
C             *** CHECK FOR TURBULENT SEPARATION ***
C
         IF(H32.LT.1.5) GOTO 70
C
C             *** COMPUTE TURBULENT SKIN FRICTION ***
C
         H12=F1(H32)
         CFT=WSHR(H12,RD2)
```

```
            CF=2.0D0*CFT
         END IF
         D1=H12*D2
         IF(K.EQ.NPRINT.AND.DUMP) WRITE(3,20) S,H32,D1,D2,D3,CF,RMARGN
         IF(K.EQ.NPRINT) K=0
50    CONTINUE
      SCRIT=1.0D0
      IF(LMNR) THEN
         IF(DUMP) WRITE(3,60)
60       FORMAT(//,10X,' LAMINAR THROUGHOUT',/10X,' NO SEPARATION')
      ELSE
         IF(DUMP) WRITE(3,65) STRANS
65       FORMAT(//,10X,' TRANSITION AT S = ',F8.4,
     &           /,10X,' NO SEPARATION')
      END IF
      GOTO 200
C
C        *** IF CONTROL IS PASSED TO LINE 70 SEPARATION HAS OCCURED AND THE ***
C        *** INTEGRATION IS SUSPENDED AT THE POINT OF SEPARATION.           ***
C
70    SEP=.TRUE.
      SCRIT=S
      IF(LMNR) THEN
         CALL FAPP(H32,H12,EPS,D,KAPS)
         CF=2.0D0*EPS/RD2*VS*VS
         D1=H12*D2
         IF(DUMP) WRITE(3,20) S,H32,D1,D2,D3,CF,RMARGN
         IF(DUMP) WRITE(3,80) S
80       FORMAT(//,10X,' LAMINAR SEPARATION AT S = ',F8.4)
      ELSE
         H12=2.9999D0
         CF=2.0*WSHR(H12,RD2)
         D1=H12*D2
         IF(DUMP) WRITE(3,20) S,H32,D1,D2,D3,CF,RMARGN
         IF(DUMP) WRITE(3,90) STRANS,S
90       FORMAT(//,10X,' TRANSITION AT S = ',F8.4,
     &           /,10X,' TURBULENT SEPARATION AT S = ',F8.4)
      END IF
      GOTO 200
200   RETURN
      END
```

```
      SUBROUTINE BODGEN(XODIVL,XLDIVL,THETA,HDIVL,VO)
C
C*************************************************************************** C
C                                                                           *
C THIS SUBROUTINE GENERATES AN AUGMENTOR BODY WITH VARIABLE PARAMETERS NOZZLE * C
C LOCATION, LIP ROTATION POINT, LIP ROTATION ANGLE, AND MIXING CHAMBER HEIGHT.* C
C ON INPUT ALL GEOMETRIC PARAMETERS ARE NORMALIZED BY THE BODY LENGTH.  THE   *IS
C SCALING IS CHANGED INTERNALLY WITH ALL GEOMETRIC QUANTITIES BEING REFERENCED* *
C TO THE CHANEL HALF WIDTH, WHICH IS ASSUMED TO BE UNITY.                    *
C                                                                           *
C   *** PARAMETER DESCRIPTION ***                                           *
C                                                                           *
C     INPUT:                                                                *
C  XODIVL - JET NOZZLE POSITION DIVIDED BY THE BODY LENGTH                  *
C  XLDIVL - LIP ROTATION POINT DIVIDED BY THE BODY LENGTH                   *
C  THETA  - LIP ROTATION ANGLE IN RADIANS                                   *
C  HDIVL  - CHANEL HALF-WITDH DIVIDED BY THE BODY LENGTH                    *
C  VO     - FREE-STREAM VELOCITY NORMALIZED BY THE VELOCITY AT THE CONTROL   *
C           STATION                                                         *
C                                                                           *
C     OUTPUT:                                                               *
C  THE OUTPUT IS PROVIDED IN THE FORM OF DATA FILES.  BODY.DAT CONTAINS THE  *
C  SURFACE COORDINATE PAIRS AS WELL AS THE TRANSPIRATION VELOCITY OVER EACH  *
C  PANEL.  PARAM.DAT CONTAINS SCALE INFORMATION FOR THE BODY.               *
C                                                                           *
C*************************************************************************** C

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XP(20),XTEMP(150),YTEMP(150),VNTEMP(150)
      DIMENSION XNOSE(25),YNOSE(25),XSPLN(10),YSPLN(10),SPLN(30)
      LOGICAL FLAG
      REWIND 1
      REWIND 2
      PI=3.1415926
5     FORMAT(3F10.4)
C
C        *** DEFINE JET BOUNDARY SLOPE TO BE 12 DEG ***
C
      SLOPE=DTAN(12.0D0/180.0D0*PI)
C
C        *** COMPUTE THE SCALED BODY LENGTH ***
C
      XEND=1.0/HDIVL
C
C        *** COMPUTE THE SCALED NOZZLE LOCATION ***
C
      XO=XODIVL*XEND
C
C        *** COMPUTE THE CONTROL STATION LOCATION ***
C
      XCONT=(XODIVL+0.7*HDIVL/SLOPE)*XEND
C
```

```
C        *** COMPUTE THE NOSE RADIUS (CONSTANT FRACTION OF L) ***
C
      RNDIVL=1.0D0/12.0D0
      RN=RNDIVL*XEND
C
C        *** COMPUTE THE LIP ROTATION POINT ***
C
      XLIP=XLDIVL*XEND
C
C        *** IF THE LIP ROTATION POINT IS LESS THAN THE NOSE RADIUS, SET ***
C        *** THE LIP ROTATION POINT EQUAL TO THE NOSE RADIUS IN ORDER TO ***
C        *** AVOID A CONTORTED BODY SHAPE ***
C
      IF(XLIP.LT.RN) XLIP=RN
C
C        *** CKECK TO INSURE THAT THE CONTROL STATION IS BEHIND THE LIP ***
C        *** ROTATION POINT, IF NOT PRINT ERROR MESSAGE AND SUSPEND     ***
C        *** EXECUTION                                                  ***
C
      IF(XLIP.GT.XCONT) THEN
         WRITE(3,10) XODIVL,XLDIVL,HDIVL,THETA
10       FORMAT(' IN BODGEN XLIP WAS GREATER THAN XCONT. PARAMETERS',
     &          ' ON ENTRY WERE',/,' XODIVL =',F8.4,' XLDIVL =',F8.4,
     &          ' HDIVL =',F8.4,' THETA =',F8.4)
         STOP
      END IF
C
C        *** CHECK TO INSURE THAT THE CONTROL STATION IS AHEAD OF THE BODY ***
C        ***  END, IF NOT WRITE AN ERROR MESSAGE AND SUSPEND EXECUTION     ***
C
      IF(XCONT.GT.XEND) THEN
         WRITE(3,11) XODIVL,XLDIVL,HDIVL,THETA
11       FORMAT(' IN BODGEN XCONT WAS GREATER THAN XEND.  PARAMETERS',
     &          ' ON ENTRY WERE',/,' XODIVL =',F8.4,' XLDIVL =',F8.4,
     &          ' HDIVL =',F8.4,' THETA =',F8.4)
         STOP
      END IF
C
C        *** DEFINE EXTREMITIES OF THE SYMMETRY PLANES ***
C
      X1=-20.
      XM=26.
C
C        *** INITIALAIZE PARAMETERS ***
C
      FLAG=.TRUE.
      DIST=XO-X1
      XI=.06
      XIM1=0.
C
C        *** GENERATE A STRING OF COORDINATES WHICH HAVE A RATIO OF ***
C        *** SUCCESSIVE LENGTHS EQUAL TO 1.5 ***
```

```
C
      DO 50 I=1,20
         XP(I)=XI
         XI=2.5*XI-1.5*XIM1
         XIM1=XP(I)
         IF(XIM1.GT.DIST) GOTO 60
50    CONTINUE
60    N=I
      Y=0.
      J=0.
      DO 70 I=1,N
         X=XO-XP(N-I+1)
         J=J+1
         XTEMP(J)=X
         YTEMP(J)=Y
         VNTEMP(J)=VN
70    CONTINUE
C
C       *** GENERATE A SET OF COORDINATES FOR THE JET BOUNDARY WHICH HAS   ***
C       *** THE FOLLOWING PROPERTIES: PANEL LENGTHS INCREASE IN A RATIO OF ***
C       *** 1.5 AS ONE TRAVERSES AWAY FROM THE JET NOZZLE, AND AS ONE      ***
C       *** TRAVERSES AWAY FROM THE CONTROL STATION MOVING TOWARDS THE     ***
C       *** NOZZLE.  THE INCREASING PANEL LENGTH IS HALTED WHEN THE LENGTH ***
C       *** IS APROXIMATELY 0.3.  THE MIDDLE SECTION OF THE JET BOUNDARY   ***
C       *** CONSTANT X INCREMENT OF 0.2851.                                ***
C
      DX=0.2851D0
      DO 80 I=1,16
         VN=.15*DSQRT(1./(X-XO+0.1))+.2
         J=J+1
         IF(I.EQ.1) THEN
            JS=J
            X=XO
            Y=0.0D0
         END IF
         IF(1.LT.I.AND.I.LE.6) THEN
            X=XO+XP(I-1)
            Y=SLOPE*(X-XO)
         END IF
         IF(6.LT.I.AND.I.LE.13) THEN
            X=X+DX
            Y=SLOPE*(X-XO)
         END IF
         IF(13.LT.I.AND.I.LE.18) THEN
            X=XCONT-XP(17-I)
            Y=SLOPE*(X-XO)
         END IF
         XTEMP(J)=X
         YTEMP(J)=Y
         VNTEMP(J)=VN
80    CONTINUE
90    JF=J
```

```
C
C        *** GENERATE THE POINTS WHICH DEFINE THE CONTROL STATION ***
C
      X=XCONT
      Y=SLOPE*(X-XO)
      R=.5*(1.-Y)
      YC=Y+R
      DANG=PI/8.
      ANG=-PI/2.
      DO 100 I=1,8
         VN=DCOS(ANG+DANG/2.)
         J=J+1
         XTEMP(J)=X
         YTEMP(J)=Y
         VNTEMP(J)=VN
         ANG=ANG+DANG
         X=XCONT+R*DCOS(ANG)
         Y=YC+R*DSIN(ANG)
100   CONTINUE
C
C        *** GENERATE NOSE POINTS AND STORE ***
C
      XS=RN*(1.-DSIN(THETA))
      X=XS
      Y=1.+DTAN(THETA)*(XLIP-X)
      XC=X+RN*DSIN(THETA)
      YC=Y+RN*DCOS(THETA)
      DEL=0.0
      IF(DABS(DSIN(THETA)).GT.1.E-3)
     &   DEL=2.0D0*RN*(DTAN(THETA)-(1.0D0-DCOS(THETA))/DSIN(THETA))
      ANG=PI
      NRN=NINT(RN*PI/0.15D0)
      DANG=PI/DFLOAT(NRN)
      NCIR=NRN+1
      DO 150 I=1,NCIR
         XNOSE(I)=X
         YNOSE(I)=Y
         ANG=ANG-DANG
         XCI=RN*DCOS(ANG)
         ETA=RN*DSIN(ANG)
         XIM1=X
         X=XC+XCI*DCOS(PI/2.-THETA)-ETA*DSIN(PI/2.-THETA)
         XTMP=X
         Y=YC+XCI*DSIN(PI/2.-THETA)+ETA*DCOS(PI/2.-THETA)
150   CONTINUE
C
C        *** SPLINE FIT THE SECTION BETWEEN THE CONTROL STATION AND NOSE ***
C
      DO 105 I=1,3
         XSPLN(I)=XNOSE(4-I)
         YSPLN(I)=YNOSE(4-I)
105   CONTINUE
```

```
      XSPLN(4)=XLIP
      YSPLN(4)=1.0D0
      DO 107 I=1,3
         XSPLN(4+I)=XCONT-XP(4-I)
         YSPLN(4+I)=1.0D0
107   CONTINUE
      NSPL=7
      NFSPL=6
      CALL ICSCCU(XSPLN,YSPLN,NSPL,SPLN,NFSPL,IER)
      IF(IER.EQ.129.OR.IER.EQ.130.OR.IER.EQ.130) THEN
         WRITE(3,109) IER
109      FORMAT(' IN BODGEN ICSCCU RETURNED WITH THE ERROR VALUE ',I5)
         STOP
      END IF
C
C        *** GENERATE POINTS BETWEEN THE CONTROL STATION AND NOSE USING THE ***
C        *** SPLINE FIT                                                     ***
C
      X=XCONT
      Y=1.
      VN=0.0
      J=J+1
      XTEMP(J)=X
      YTEMP(J)=Y
      DO 110 I=1,3
         J=J+1
         X=XCONT-XP(I)
         XTEMP(J)=X
         CALL INTRPV(X,XSPLN,YSPLN,SPLN,NSPL,Y,YD)
         YTEMP(J)=Y
         VNTEMP(J)=VN
110   CONTINUE
      DX=0.15D0
      IEND=NINT((X-XS)/DX)
      DX=(X-XS)/DFLOAT(IEND)
      XN=RN*DCOS(THETA)
      FLAG=.TRUE.
      L=0
      DO 120 I=1,IEND-1
         J=J+1
         X=X-DX
         XTEMP(J)=X
         CALL INTRPV(X,XSPLN,YSPLN,SPLN,NSPL,Y,YD)
         YTEMP(J)=Y
         VNTEMP(J)=VN
         IF(X.LE.XLIP) THEN
            L=L+1
           IF(L.EQ.1) THEN
              JLS=J-1
              FLAG=.FALSE.
            END IF
         END IF
```

```
120   CONTINUE
C
C     *** GENTERATE THE NOISE POINTS USING THE STORED DATA ***
C
      X=XS
      DO 151 I=1,NCIR
         IF(FLAG.AND.I.EQ.1) JLS=J
         J=J+1
         XTEMP(J)=XNOSE(I)
         YTEMP(J)=YNOSE(I)
         VNTEMP(J)=VN
151   CONTINUE
C
C        *** GENERATE POINTS FROM NOSE TO INFINITY ***
C
      DO 155 I=1,2
         XSPLN(I)=XNOSE(NCIR-2+I)
         YSPLN(I)=YNOSE(NCIR-2+I)
155   CONTINUE
      Y=(1.+2.*RN)+DTAN(THETA)*(XLIP+DEL-XTMP)
      XSPLN(3)=XTMP
      YSPLN(3)=Y
      XSPLN(4)=XLIP+DEL
      YSPLN(4)=1.0D0+2.0D0*RN
      DO 157 I=1,3
         YSPLN(I+4)=1.0D0+2.0D0*RN
157   CONTINUE
      CALL ICSCCU(XSPLN,YSPLN,NSPL,SPLN,NFSPL,IER)
      IF(IER.EQ.129.OR.IER.EQ.130.OR.IER.EQ.130) THEN
         WRITE(3,109) IER
         STOP
      END IF
      XI=XTMP
      L=0
160   DO 170 I=1,80
         X=XI
         J=J+1
         IF(X.LT.XCONT-.1) THEN
            CALL INTRPV(X,XSPLN,YSPLN,SPLN,NSPL,Y,YD)
         ELSE
            Y=1.0D0+2.0D0*RN
         END IF
         IF(X.GT.(XLIP+DEL)) THEN
            L=L+1
            IF(L.EQ.1) JLF=J
         END IF
         XTEMP(J)=X
         YTEMP(J)=Y
         VNTEMP(J)=VN
         XI=2.2*XI-1.2*XIM1
         XIM1=X
         IF(XI.GT.XM) GOTO 220
```

```
170   CONTINUE
220   NMAX=J
      BETA=0.
      U1O=24.1066 + 6.6881*XO
      DO 240 I=1,NMAX
         WRITE(1,5) XTEMP(I),YTEMP(I),VNTEMP(I)
240   CONTINUE
250   WRITE(2,260) XO,XCONT,XEND,JS,JF,JLS,JLF,VO,BETA,U1O
260   FORMAT(3F10.4,4I4,/,2F10.4,/,F10.4)
      RETURN
      END
```

```
      SUBROUTINE CHANEL(PEXIT)
C
C*************************************************************************
C                                                                        *
C     SUBROUTINE CHANEL MARCHES THE JET EQUATIONS FROM THE STATION AT WHICH  *
C THE OUTER VELOCITY HAS BECOME CONSTANT TO THE SHROUD EXIT.  THE INITIAL  *
C CONDITIONS FOR THE TIME MARCH ARE PASSED VIA COMMON BLOCK FROM SUBROUTINE *
C JET.  SINCE THERE ARE NOW FOUR UNKNOWN QUANTITIES, THE INITIAL CONDITION *
C VECTOR IS EXTENDED TO 4 ELEMENTS BY INCLUDING AN INITIAL VALUE FOR UO OF *
C 1.0.                                                                   *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  NONE                                                                  *
C                                                                        *
C     OUTPUT:                                                            *
C  PEXIT - PRESSURE AT THE SHROUD EXIT AS COMPUTED BY THE VISCOUS SOLUTION *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION C(24),W(4,9),RD(4)
      COMMON /AREA3/ S(3),X,UO
      COMMON /AREA4/ PATM
      COMMON /AREA7/ R(4)
      COMMON /AREA10/ XC
      COMMON /AREA12/ XEXIT
      EXTERNAL FCN2
C
C        *** INITIALIZE DEPENDENT VARIABLE VECTOR ***
C
      R(1)=UO
      R(2)=S(1)
      R(3)=S(2)
      R(4)=S(3)
      N=4
      NW=4
      TOL=.01
      IND=1
      NEND=20
      DX=(XEXIT-X)/DFLOAT(NEND)
C
C        *** MARCH THE VISCOUS SOLUTION ***
C
      DO 50 I=1,NEND
         XEND=X+DX
         CALL DVERK(N,FCN2,X,R,XEND,TOL,IND,C,NW,W,IER)
         CALL FCN2(N,X,R,RD)
50    CONTINUE
C
C        *** DEFINE THE EXIT PRESSURE ***
C
```

```
PEXIT=R(4)
RETURN
END
```

```
      SUBROUTINE COEF(XI,X,Y,J,ALPHA,D,N,A,B)
C
C*************************************************************************
C                                                                        *
C     SUBROUTINE  COEF COMPUTES THE AERODYNAMIC INFLUENCE COEFFICIENTS FOR *
C USE IN THE HIGHER ORDER PANEL METHOD.                                  *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  XI     - COORDINATES OF THE CONTROL POINTS STORED AS X,Y PAIRS        *
C  X      - X COORDINATE AT WHICH THE INFLUENCE COEFFICIENT IS TO BE     *
C           CALCULATED                                                   *
C  Y      - Y COORDINATE AT WHICH THE INFLUENCE COEFFICIENT IS TO BE     *
C           CALCULATED                                                   *
C  J      - PANEL NUMBER AT WHICH THE INFLUENCE COEFFICIENT IS TO BE     *
C           CALCULATED                                                   *
C  ALPHA - VECTOR OF SURFACE SLOPES FOR EACH PANEL                       *
C  D      - VECTOR OF PANEL LENGTHS                                      *
C  N      - NUMBER OF PANELS                                             *
C  SENT IN COMMON FROM SUBROUTINE ANGLE ARE THE PARABOLA FIT COEFFIECINTS AS *
C  WELL AS THE VECTOR OF SURFACE CURVATURES                              *
C                                                                        *
C     OUTPUT:                                                            *
C  A      - INFLUENCE COEFFICIENT FOR THE X COMPONENT OF VELOCITY        *
C  B      - INFLUENCE COEFFICIENT FOR THE Y COMPONENT OF VELOCITY        *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XI(N,2),ALPHA(N),D(N)
      COMMON /ANGLE1/ PD(100),PE(100),PF(100),PG(100),PH(100),
     &               PPI(100),C(100)
      COMMON /ANGLE2/ PERDT
C
C     *** PERDT IS A LOGICAL VARIABLE SET TO TRUE IF THE GEOMETRY IS ***
C     *** PERIODIC                                                   ***
C
      LOGICAL PERDT
      PI=3.141593
      IF(.NOT.PERDT.AND.(J.EQ.1.OR.J.EQ.N)) GOTO 100
      L1=J-1
      L2=J
      L3=J+1
      IF(J.EQ.1) L1=N
      IF(J.EQ.N) L3=1
C
C        *** PERFORM A SET OF COORDINATE TRANSFORMATIONS TO LOCAL ***
C        *** SYSTEMS BASED ON THE PANEL WHOSE SOURCE INFLUENCE IS ***
C        *** BEING CONSIDERED AND ITS TWO NEIGHBORS               ***
C
      XSTAR1=X-XI(L1,1)
```

```
      YSTAR1=Y-XI(L1,2)
      ETA1= XSTAR1*DCOS(ALPHA(L1))+YSTAR1*DSIN(ALPHA(L1))
      ZETA1=YSTAR1*DCOS(ALPHA(L1))-XSTAR1*DSIN(ALPHA(L1))
      XSTAR2=X-XI(L2,1)
      YSTAR2=Y-XI(L2,2)
      ETA2= XSTAR2*DCOS(ALPHA(L2))+YSTAR2*DSIN(ALPHA(L2))
      ZETA2=YSTAR2*DCOS(ALPHA(L2))-XSTAR2*DSIN(ALPHA(L2))
      XSTAR3=X-XI(L3,1)
      YSTAR3=Y-XI(L3,2)
      ETA3= XSTAR3*DCOS(ALPHA(L3))+YSTAR3*DSIN(ALPHA(L3))
      ZETA3=YSTAR3*DCOS(ALPHA(L3))-XSTAR3*DSIN(ALPHA(L3))
C
C        *** COMPUTE DISTANCES FROM THE PANEL WHOSE SOURCE IS BEING ***
C        *** CONSIDERED AND ITS TWO NEIGHBORS TO THE POINT X,Y      ***
C
      RP1=(ETA1+D(L1)/2.)**2+ZETA1**2
      RM1=(ETA1-D(L1)/2.)**2+ZETA1**2
      RP2=(ETA2+D(L2)/2.)**2+ZETA2**2
      RM2=(ETA2-D(L2)/2.)**2+ZETA2**2
      RP3=(ETA3+D(L3)/2.)**2+ZETA3**2
      RM3=(ETA3-D(L3)/2.)**2+ZETA3**2
C
C        *** COMPUTE THE COEFFICIENTS IN THE POWER SERIES EXPANSION ***
C        *** FOR THE INFLUENCE COEFFICIENTS                         ***
C
      VEO2=DLOG(RP2/RM2)
      DEN=ETA2**2+ZETA2**2-(D(L2)/2.)**2
      RNUM=ZETA2*D(L2)
      VZO2=2.*DATAN(RNUM/DEN)
      IF(DABS(RNUM).LT.1.E-3.AND.DEN.LT.0.) GOTO 2
      IF(RNUM.GT.0..AND.DEN.LT.0.) VZO2=VZO2+2.*PI
      IF(RNUM.LT.0..AND.DEN.LT.0.) VZO2=VZO2-2.*PI
      GOTO 3
2     VZO2=2.*PI
3     VE12=(ZETA2*VZO2+ETA2*VEO2-2.*D(L2))/D(L2)
      VZ12=(ETA2*VZO2-ZETA2*VEO2)/D(L2)
      VEC2=(-2.*ETA2*VZO2+2.*ZETA2*VEO2+ETA2*ZETA2*D(L2)**3
     &     /RP2/RM2)/D(L2)
      VZC2=(2.*ZETA2*VZO2+2.*ETA2*VEO2-2.*D(L2)*(1.+
     &     ((ETA2**2+ZETA2**2)**2-(ETA2**2-ZETA2**2)
     &       *((D(L2)/2.)**2))/RP2/RM2))/D(L2)
      VE22=(2.*ETA2*ZETA2*VZO2+(ETA2**2-ZETA2**2)*VEO2
     &            -2.*ETA2*D(L2))/(D(L2)**2)
      VZ22=((ETA2**2-ZETA2**2)*VZO2-2.*ETA2*ZETA2*VEO2
     &            +2.*ZETA2*D(L2))/(D(L2)**2)
      VEO1=DLOG(RP1/RM1)
      DEN=ETA1**2+ZETA1**2-(D(L1)/2.)**2
      RNUM=ZETA1*D(L1)
      VZO1=2.*DATAN(RNUM/DEN)
      IF(DABS(RNUM).LT.1.E-3.AND.DEN.LT.0.) GOTO 4
      IF(RNUM.GT.0..AND.DEN.LT.0.) VZO1=VZO1+2.*PI
      IF(RNUM.LT.0..AND.DEN.LT.0.) VZO1=VZO1-2.*PI
```

```
      GOTO 5
4     VZO1=2.*PI
5     VE11=(ZETA1*VZO1+ETA1*VEO1-2.*D(L1))/D(L1)
      VZ11=(ETA1*VZO1-ZETA1*VEO1)/D(L1)
      VE21=(2.*ETA1*ZETA1*VZO1+(ETA1**2-ZETA1**2)*VEO1
     &            -2.*ETA1*D(L1))/(D(L1)**2)
      VZ21=((ETA1**2-ZETA1**2)*VZO1-2.*ETA1*ZETA1*VEO1
     &            +2.*ZETA1*D(L1))/(D(L1)**2)
      VEO3=DLOG(RP3/RM3)
      DEN=ETA3**2+ZETA3**2-(D(L3)/2.)**2
      RNUM=ZETA3*D(L3)
      VZO3=2.*DATAN(RNUM/DEN)
      IF(DABS(RNUM).LT.1.E-3.AND.DEN.LT.0.) GOTO 6
      IF(RNUM.GT.0..AND.DEN.LT.0.) VZO3=VZO3+2.*PI
      IF(RNUM.LT.0..AND.DEN.LT.0.) VZO3=VZO3-2.*PI
      GOTO 7
6     VZO3=2.*PI
7     VE13=(ZETA3*VZO3+ETA3*VEO3-2.*D(L3))/D(L3)
      VZ13=(ETA3*VZO3-ZETA3*VEO3)/D(L3)
      VE23=(2.*ETA3*ZETA3*VZO3+(ETA3**2-ZETA3**2)*VEO3
     &            -2.*ETA3*D(L3))/(D(L3)**2)
      VZ23=((ETA3**2-ZETA3**2)*VZO3-2.*ETA3*ZETA3*VEO3
     &            +2.*ZETA3*D(L3))/(D(L3)**2)
      VE2=VEO2+VE12*PE(L2)*D(L2)+VEC2*(-0.5*C(L2))*D(L2)+
     &         VE22*(PH(L2)+2.*(-0.5*C(L2))**2)*D(L2)**2
      VE1=VE11*PF(L1)*D(L1)+VE21*PPI(L1)*D(L1)**2
      VE3=VE13*PD(L3)*D(L3)+VE23*PG(L3)*D(L3)**2
      VZ2=VZO2+VZ12*PE(L2)*D(L2)+VZC2*(-0.5*C(L2))*D(L2)+
     &         VZ22*(PH(L2)+2.*(-0.5*C(L2))**2)*D(L2)**2
      VZ1=VZ11*PF(L1)*D(L1)+VZ21*PPI(L1)*D(L1)**2
      VZ3=VZ13*PD(L3)*D(L3)+VZ23*PG(L3)*D(L3)**2
C
C        *** COMPUTE THE INFLUENCE COEFFICIENTS ***
C
      A=VE1*DCOS(ALPHA(L1))+VE2*DCOS(ALPHA(L2))
     &            +VE3*DCOS(ALPHA(L3))
     &            -VZ1*DSIN(ALPHA(L1))-VZ2*DSIN(ALPHA(L2))
     &            -VZ3*DSIN(ALPHA(L3))
      B=VE1*DSIN(ALPHA(L1))+VE2*DSIN(ALPHA(L2))
     &            +VE3*DSIN(ALPHA(L3))
     &            +VZ1*DCOS(ALPHA(L1))+VZ2*DCOS(ALPHA(L2))
     &            +VZ3*DCOS(ALPHA(L3))
      GOTO 9
C
C        *** TREAT THE END PANELS (IE 1 AND N ) SEPARATELY ***
C
100   L2=J
      XSTAR2=X-XI(L2,1)
      YSTAR2=Y-XI(L2,2)
      ETA2= XSTAR2*DCOS(ALPHA(L2))+YSTAR2*DSIN(ALPHA(L2))
      ZETA2=YSTAR2*DCOS(ALPHA(L2))-XSTAR2*DSIN(ALPHA(L2))
      RP2=(ETA2+D(L2)/2.)**2+ZETA2**2
```

```
      RM2=(ETA2-D(L2)/2.)**2+ZETA2**2
      VEO2=DLOG(RP2/RM2)
      DEN=ETA2**2+ZETA2**2-(D(L2)/2.)**2
      RNUM=ZETA2*D(L2)
      VZO2=2.*DATAN(RNUM/DEN)
      IF(DABS(RNUM).LT.1.E-3.AND.DEN.LT.0.) GOTO 102
      IF(RNUM.GT.0..AND.DEN.LT.0.) VZO2=VZO2+2.*PI
      IF(RNUM.LT.0..AND.DEN.LT.0.) VZO2=VZO2-2.*PI
      GOTO 103
102   VZO2=2.*PI
103   A=VEO2*DCOS(ALPHA(L2))-VZO2*DSIN(ALPHA(L2))
      B=VEO2*DSIN(ALPHA(L2))+VZO2*DCOS(ALPHA(L2))
9     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DATIN(XJ,VN,M)
C
C*************************************************************************
C                                                                        *
C     SUBROUTINE DATIN READS THE DATA FILE BODY.DAT TO OBTAIN THE COORDINATES *
C OF THE SHROUD GEOMETRY AS WELL AS THE NORMAL VELOCITY AT EACH PANEL.   *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C     INPUT:                                                             *
C  THE INPUT IS THE DATA FILE BODY.DAT WHICH CONTAINS THE X AND Y COORDINATES *
C  ALONG WITH THE TRANSPIRATION VELOCITY FOR EACH PANEL IN FIELDS OF 3F10.4  *
C                                                                        *
C     OUTPUT:                                                            *
C  XJ     - THE BODY COORDINATES STORED AS X,Y PAIRS                     *
C  VN     - VECTOR OF PANEL TRANSPIRATION VELOCITIES                     *
C  M      - NUMBER OF COORDINATE PAIRS                                   *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XJ(M,2),VN(M)
10    FORMAT(3F10.4)
      REWIND 1
      DO 30 I=1,M
         READ(1,10) XJ(I,1),XJ(I,2),VN(I)
30    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FAPP(H32,H12,EPS,D,KAPS)
C
C*************************************************************************
C                                                                        *
C     THIS SUBROUTINE COMPUTES THE LOCAL SKIN FRICTION COEFFICIENT (EPS), AND *
C THE LOCAL DISSIPATION COEFFICIENT (D) FOR THE LAMINAR BOUNDARY EQUATIONS.   *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  H32  - SHAPE FACTOR                                                   *
C  H12  - SHAPE FACTOR                                                   *
C                                                                        *
C     OUTPUT:                                                            *
C  EPS  - LOCAL SKIN FRICTION COEFFICIENT                                *
C  D    - LOCAL DISSIPATION COEFFICIENT                                  *
C  KAPS - LAMINAR SEPARARION PARAMETER.  KAPS=1 FOR ATTACHED FLOW AND    *
C         KAPS=0 FOR SEPARATED FLOW                                      *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      KAPS=1
      D=7.853976D0-10.260551D0*H32+3.418898*H32*H32
      IF(H32-1.51509D0)10,20,30
10    KAPS=0
      RETURN
20    H12=4.02922D0-(583.60182D0-724.55916D0*H32+227.1822D0
     &               *H32*H32)*DSQRT(H32-1.51509D0)
      EPS=2.512589D0-1.686095D0*H12+0.391541*H12*H12-0.031729*H12**3.D0
      RETURN
30    IF(H32-1.57258D0)21,21,40
21    GOTO 20
40    H12=79.870845D0-89.582142D0*H32+25.715786D0*H32*H32
      EPS=1.372391-4.226253*H32+2.221687*H32*H32
      RETURN
      END
```

```
      SUBROUTINE FCNL(NE,S,Y,YD)
C
C*************************************************************************
C                                                                        *
C     THIS SUBROUTINE COMPUTES THE DERIVATIVES OF D2 AND D3 FOR THE LAMINAR  *
C BOUNDARY LAYER EQUATIONS.  A CALL TO SUBROUTINE FAPP IS NECESSARY.      *
C                                                                        *
C     *** PARAMETER DESCRIPTION ***                                      *
C                                                                        *
C  NE - NUMBER OF DIFFERENTIAL EQUATIONS, IN THIS CASE 2                 *
C  S  - SURFACE COORDINATE                                               *
C  Y  - VECTOR CONTAINING THE VALUES OF D2 AND D3 AT THE STATION S       *
C  YD - VECTOR CONTAINING THE DERIVATIVE VALUES OF D2 AND D3 AT THE STATION S *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(NE),YD(NE)
      COMMON /BLCVEL/ X(100),V(100),R
      COMMON /BLCSPLN/ SPLN(100),N
      D2=Y(1)
      D3=Y(2)
      H32=D3/D2
C
C        *** COMPUTE THE FRICTION AND DISSIPATION COEFFICIENTS ***
C
      CALL FAPP(H32,H12,EPS,D,KAPS)
C
C        *** COMPUTE THE LOCAL SURFACE VELOCITY AND ITS DERIVATIVE ***
C        *** FROM THE LINEAR SPLINE FIT                            ***
C
      CALL LINTRP(S,X,V,SPLN,N,VS,VSD,IER)
      IF(IER.EQ.1) THEN
         WRITE(3,71) S
71       FORMAT(' IN FCNL LINTRP RETURNED WITH AN ERROR FLAG',/,
     &          ' X HAD THE VALUE',F10.6,' ON ENTRY')
         STOP
      END IF
      RD2=R*VS*D2
      CFL1541*H12*H12-0.031729*H12**3.D0
      RETURN
30    IF(H32-1.57258D0)21,21,40
21    GOTO 20
40    H12=79.870845D0-89.582142D0*H32+25.715786D0*H32*H32
      EPS=1.372391-4.226253*H32+2.221687*H32*H32
      RETURN
      END
```

```
      SUBROUTINE FCNT(NE,S,Y,YD)
C
C*************************************************************************
C                                                                        *
C     THIS SUBROUTINE COMPUTES THE DERIVATIVES OF D2 AND D3 FOR USE IN THE    *
C MARCHING OF THE TURBULENT BOUNDARY LAYER EQUATIONS.                    *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  NE - NUMBER OF DIFFERENTIAL EQUATIONS, IN THIS CASE 2                 *
C  S  - SURFACE COORINATE                                                *
C  Y  - VECTOR CONTAINING THE VALUES OF D2 AND D3 AT THE STATION S       *
C  YD - VECTOR CONTAINING THE DERIVATIVE VALUES OF D2 AND D3 AT THE STATION S *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(NE),YD(NE)
      COMMON /BLCVEL/ X(100),V(100),R
      COMMON /BLCSPLN/ SPLN(100),N
C
C        *** FUNCTION F1 RETURNS H12 GIVEN H32 ***
C
      F1(H32)=H32/(3.0D0*H32-4.0D0)
C
C        *** FUNCTION WSHR RETURNS THE LOCAL TURBULENT SKIN FRICTION ***
C        *** COEFFICIENT, GIVEN THE SHAPE FACTOR H12, AND THE REYNOLDS ***
C        *** NUMBER BASED ON MOMENTUM THICKNESS RD2 ***
C
      WSHR(H12,RD2)=0.0245D0*(1.0D0-2.0959D0*DLOG10(H12))**1.705D0
     &              /RD2**0.268D0
C
C        *** FUNCTION CDISS RETURNS THE LOCAL TURBULENT DISSIPATION ***
C        *** COEFFICIENT ***
C
      CDISS(H32,RD2)=(0.00481D0+0.0822D0*(H32-1.5D0)**4.81D0)
     & *(H32/RD3)**(0.2317D0*H32-0.2664D0-0.87D5*(2.0D0-H32)**20)
      D2=Y(1)
      D3=Y(2)
      H32=D3/D2
      H12=F1(H32)
C
C        *** TO AVOID SINGULARITIES AT SEPARATION, PUT BARRIERS ON ***
C        *** H32 AND H12 ***
C
      IF(H32.LT.1.5) H32=1.51
      IF(H12.GT.3.0) H12=2.99
C
C        *** FIND THE LOCAL SURFACE VELOCITY AND ITS DERIVATIVE THROUGH ***
C        *** USE OF THE LINEAR SPLINE FIT ***
C
```

```
      CALL LINTRP(S,X,V,SPLN,N,VS,VSD,IER)
      IF(IER.EQ.1) THEN
         WRITE(3,71) S
71       FORMAT(' IN FCN` LINTRP RETURNED WITH AN ERROR FLAG',/,
     &          ' X HAD THE VALUE',F10.6,' ON ENTRY')
         STOP
      END IF
      RD2=R*VS*D2
      RD3=R*VS*D3
      CT=WSHR(H12,RD2)
      CD=CDISS(H32,RD3)
      YD(1)=-(2.0D0+H12)*D2/VS*VSD + CT
      YD(2)=-3.0D0*Y(2)/VS*VSD + 2.0D0*CD
      RETURN
      END
```

```
      SUBROUTINE FIX
C
C*************************************************************************
C                                                                        *
C     SUBROUTINE FIX UPDATES THE ENTRAINMENT VELOCITY DISTRIBUTION AND JET *
C INITIAL CENTERLINE VELOCITY AS CONTAINED IN THE DATA FILES BODY.DAT AND *
C PARAM.DAT.                                                             *
C                                                                        *
C*************************************************************************
C
      DIMENSION XJ(250),YJ(250),VEL(250)
      REWIND 1
      REWIND 2
      REWIND 12
C
C        *** UPDATE U1O IN DATA FILE PARAM.DAT (UNIT 2) ***
C
      READ(2,10) XO,XC,XEXIT,NJS,NJF,NLS,NLF,VO,BETA
10    FORMAT(3F10.4,4I4,/,2F10.4)
      READ(12,20) U1O
20    FORMAT(F20.4)
      WRITE(2,30) U1O
30    FORMAT(F10.4)
C
C        *** UPDATE THE ENTRAINMENT VELOCITY DISTRIBUTION IN FILE ***
C        *** BODY.DAT (UNIT 3)                                    ***
C
      DO 90 I=1,100
         READ(1,60,END=95) X,Y,VN
60       FORMAT(3F10.4)
         IF(I.LT.NJS.OR.I.GT.NJF) GOTO 70
         READ(12,5) VN
5        FORMAT(F10.4)
70       XJ(I)=X
         YJ(I)=Y
         VEL(I)=VN
90    CONTINUE
95    NMAX=I-1
      REWIND 1
      DO 100 I=1,NMAX
         WRITE(1,60) XJ(I),YJ(I),VEL(I)
100   CONTINUE
110   RETURN
      END
```

```
      SUBROUTINE FLDVEL(XI,ALPHA,D,Q,N,X,Y,U,V)
C
C*************************************************************************
C                                                                        *
C     SUBROUTINE FLDVEL COMPUTES THE VELOCITY AT AN ARBITRARY POINT IN THE    *
C INVISCID FIELD.                                                        *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  XI     - COORDINATES OF THE CONTROL STATION LOCATIONS STORED AT X,Y PAIRS   *
C  ALPHA  - VECTOR OF SURFACE SLOPE ANGLES                               *
C  D      - VECTOR OF PANEL LENGTHS                                      *
C  Q      - VECTOR OF SOURCE STRENGTHS                                   *
C  N      - NUMBER OF PANELS                                             *
C  X      - ABSCISSA OF THE POINT AT WHICH THE VELOCITY IS CALCULATED     *
C  Y      - ORDINATE OF THE POINT AT WHICH THE VELOCITY IS CALCULATED     *
C                                                                        *
C     OUTPUT:                                                            *
C  U      - HORIZONTAL COMPONENT OF VELOCITY AT (X,Y)                     *
C  V      - VERTICAL COMPONENT OF VELOCITY AT (X,Y)                       *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XI(N,2),ALPHA(N),D(N),Q(N)
      COMMON /UNIF/ VO
         SUM1=0.
         SUM2=0.
C
C           *** WEIGHT THEINFLUENCE COEFFICIENTS WITH THE SOURCE STRENGHTS ***
C           *** AND SUM TO FIND THE VELOCITY COMPONENTS                    ***
C
         DO 20 J=1,N
            CALL COEF(XI,X,Y,J,ALPHA,D,N,A,B)
            SUM1=SUM1+Q(J)*A
            SUM2=SUM2+Q(J)*B
20       CONTINUE
         U=(VO+SUM1)
         V=SUM2
60    RETURN
      END
```

```
      SUBROUTINE INTRPV(X,XP,YP,S,N,P,PD)
C
C*************************************************************************
C                                                                        *
C    THIS SUBROUTINE USES CUBIC SPLINE FIT PARAMETERS PRODUCED BY IMSL   *
C ROUTINE ICSCCU TO FIND INTERPOLATED VALUES OF A FUNCTION AND ITS DERIVATIVE *
C AT ANY STATION X.  IN THIS VERSION, THE SPLINE FIT PARAMETER MATRIX, SPLN, *
C IS PASSED THROUGH THE ARGUMENT LIST AS A VECTOR.  WITHIN THE BODY OF THE  *
C ROUTINE IT IS RECONSTRUCTED IN MATRIX FORM.  THIS VERSION IS  HELPFUL WHEN *
C VECTORS OF DATA TO SPLINE FIT ARE OF VARIABLE DIMENSION.                *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  X  - INDEPENDENDENT COORDINATE.  X MUST BE WITHIN THE RANGE OF WHICH WAS *
C       SENT TO SUBROUTINE SPLINE.                                       *
C  XP - VECTOR OF LENGTH N CONTAINING THE X CORDINATES OF A  FUNCTION P.  *
C  YP - VECTOR OF LENGTH N CONTAINING THE  VALUE OF P AT X STATIONS       *
C       CORRESPONDING TO THOSE IN XP.                                    *
C  SPLN - VECTOR OF SPLINE FIT PARAMETERS AS OBTAINED FROM A CALL TO SYSTEM *
C         ROUTINE ICSCCU.                                                *
C  N  - NUMBER OF DATA POINTS USED IN THE SPLINE FIT (DIMENSION OF VECTORS XP *
C       AND YP)                                                          *
C                                                                        *
C     OUTPUTS                                                            *
C  P  - INTERPOLATED VALUE OF THE FUNCTION AT THE STATION X              *
C  DP - INTERPOLATED VALUE OF THE FIRST DERIVATIVE OF THE FUNCTION AT THE *
C       STATION X                                                        *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XP(N),YP(N),S(450),SPLN(150,3)
C
C        *** VERIFY THAT X IS WITHIN THE PROPER RANGE ***
C        *** EPS IS USED AS A TOLLERANCE FOR ROUND-OFF ERROR ***
C
      EPS=1.0D-6
      IF((XP(1)-X).GT.EPS.OR.(X-XP(N)).GT.EPS) GOTO 100
C
C        *** RECONSTRUCT THE SPLN MATRIX ***
C
      NF=N-1
      DO 5 I=1,3
         DO 3 J=1,NF
            SPLN(J,I)=S((I-1)*N+J)
3        CONTINUE
5     CONTINUE
C        *** SEARCH THROUGH THE ABSCISSA VECTOR TO LOCATE THE INTERVAL IN ***
C        *** WHICH X LIES.                                              ***
C
      DO 10 J=1,NF
```

```
         IF(J.EQ.NF) GOTO 20
         IF((XP(J)-EPS).LE.X.AND.X.LT.XP(J+1)) GOTO 20
10    CONTINUE
C
C        *** COMPUTE INPERPOLATED VALUES ***
C
20    D=X-XP(J)
      P=SPLN(J,3)*D**3+SPLN(J,2)*D**2+SPLN(J,1)*D+YP(J)
      PD=3.*SPLN(J,3)*D**2+2.*SPLN(J,2)*D+SPLN(J,1)
      GOTO 200
100   WRITE(3,110) X
110   FORMAT('IN SUBROUTINE INTERPOLATE X IS OUT OF BOUNDS.',/,
     &       'X HAS THE VALUE ',F10.4)
200   RETURN
      END
```

```
      SUBROUTINE JET(U1O,VN,RES)
C
C*************************************************************************
C                                                                        *
C     SUBROUTINE JET PERFORMS THE VISCOUS CALCULATION WITHIN THE VISCOUS- *
C INVISCID INTERACTION REGION.  THE DERIVATIVE OF UO IS FOUND FROM THE   *
C INVISCID SOLUTION VIA A LINEAR SPLINE FIT, AND IS USED AS A FORCING TERM IN *
C THE VISCOUS SOLUTION.                                                  *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  U1O - JET INITIAL CENTERLINE VELOCITY                                 *
C  VN  - VECTOR CONTAINING THE NORMAL VELOCITIES TO THE PANELS ALONG THE JET *
C        BOUNDARY IN THE VISCOUS-INVISCID INTERACTION REGION             *
C                                                                        *
C     OUTPUT:                                                            *
C  VN -  UPDATED NORMAL VELOCITY VECTOR                                  *
C  RES - MAXIMUM RESIDUAL IN THE VISCOUS-INVISCID MATCHING               *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(4,4),T(4),W(3,9),C(24),SD(3),VN(100)
C
C         *** CONTAINED IN AREA1 ARE SPLINE FIT PARAMETERS FOR THE ***
C         *** HORIZONTAL COMPONENT OF VELOCITY ALONG THE JET BOUNDARY ***
C
      COMMON /AREA1/ XE(40),YE(40),UE(40),VE(40),SPLN(150),NE
C
      COMMON /AREA3/ S(3),X,UO
      COMMON /AREA4/ PATM
      COMMON /AREA8/ NJS,NJF
      COMMON /AREA11/ XO
      EXTERNAL FCN1
      REWIND 12
      WRITE(12,1) U1O
1     FORMAT(F20.4)
      NEF=NE-1
C
C        *** SPLINE FIT THE HORIZONTAL COMPONENT OF INVISCID VELOCITY ***
C        *** ALONG THE JET BOUNDARY                                  ***
C
      CALL LNSPLN(XE,UE,NE,SPLN,IER)
      IF(IER.NE.0) GOTO 80
C
      PI=3.141592D0
      THETA=12.*PI/180.
      W1=1.0
      ALP=0.693D0
      N=3
      NW=3
```

```
      TOL=.01D0
      IND=1
      X=XO+0.1
C
C        *** OBTAIN THE INTERPOLATED VALUE OF THE HORIZONTAL COMPONENT OF ***
C        *** INVISCID VELOCITY ALONG THE JET BOUNDARY                     ***
C
      CALL LINTRP(X,XE,UE,SPLN,NE,UO,UOD,IER)
      IF(IER.EQ.1) THEN
         WRITE(3,4) X
4        FORMAT(' IN JET LINTRP RETURNED WITH AN ERROR FLAG',/,
     &          ' X HAD THE VALUE',F10.6,' ON ENTRY')
         STOP
      END IF
C
C        *** COMPUTE THE TOTAL PRESSURE IN THE  INVISCID FIELD ASSUMING ***
C        *** THAT THE STATIC PRESSURE AT THE JET NOZZLE IS ZERO ***
C
      PATM=0.5D0*UO*UO
C
C        *** DEFINE INITIAL VALUES OF THE JET PARAMETERS, B IS SET TO A ***
C        *** SMALL VALUE TO AVOID THE SINGULARITY AT THE ORIGIN         ***
C
      S(1)=U1O
      S(2)=.01D0
      S(3)=PATM-0.5D0*UO*UO
      RES=0.0
C
C        *** ENTER LOOP TO MARCH THE VISCOUS EQUATIONS
C
      DO 10 J=2,NE
         XEND =XE(J)
C
C           *** IMSL SUBROUTINE DVERK MARCHES THE SOLUTION ***
C
         CALL DVERK(N,FCN1,X,S,XEND,TOL,IND,C,NW,W,IER)
         IF(IND.LT.0.OR.IER.GT.0) GOTO 100
C
C           *** OBTAIN THE LOCAL DERIVATIVE VALUES OF THE JET PARAMETERS ***
C
         CALL FCN1(N,XEND,S,SD)
C
C           *** COMPUTE THE LOCAL INVISCID VELOCITY AND ITS FDERIVATIVE ***
C
         CALL LINTRP(XEND,XE,UE,SPLN,NE,UO,UOD,IER)
         IF(IER.EQ.1) THEN
            WRITE(3,12) XEND
12          FORMAT(' IN JET LINTRP RETURNED WITH AN ERROR FLAG',/,
     &             ' X HAD THE VALUE',F10.6,' ON ENTRY')
            STOP
         END IF
         BET=ALP/(S(2)**2)
```

```
      YSTAR=2.4*S(2)
C
C        *** COMPUTE THE VERTICAL COMPONENT OF VELOCITY AT THE JET ***
C        *** BOUNDARY FROM THE VISCOUS SOLUTION                    ***
C
      V=-(UOD*YSTAR+SD(1)/2.*DSQRT(PI/BET)
     &     *DERF(DSQRT(BET)*YSTAR)
     &     +2.*S(1)*ALP/(S(2)**3)*SD(2)*
     &      (-YSTAR/2./BET*DEXP(-BET*YSTAR**2)
     &       +.25/BET*DSQRT(PI/BET)*DERF(DSQRT(BET)*YSTAR)))
C
C        *** COMPUTE THE LOCAL RESIDUAL BY COMPARING THE VISCOUS AND ***
C        *** INVISCID VERTICAL COMPONENTS OF VELOCITY ALONG THE JET  ***
C        *** BOUNDARY                                                ***
C
      R=V-VE(J)
      IF(DABS(R).GT.RES) RES=DABS(R)
C
C        *** MAKE A CORRECTION TO THE LOCAL ENTRAINMENT VELOCITY ***
C
      W1=1.-.7/DFLOAT(NE-2)*DFLOAT(J-2)
      VNEW= VN(NJS-1+J)-W1*R
C
C        *** MAKE FIRST PANEL SUCTION EQUAL TO THE SECOND TO ENHANCE ***
C        *** STABILITY                                               ***
C
      IF(J.EQ.2) VN(NJS)=VNEW
      VN(NJS-1+J)=VNEW
      IF(J.EQ.2) WRITE(12,7) VNEW
      WRITE(12,7) VNEW
7     FORMAT(F10.4)
9     CONTINUE
10  CONTINUE
    GOTO 200
80  WRITE(3,90) IER
90  FORMAT('AFTER CALL TO SPLINE IER HAS THE ERROR VALUE ',I5)
    GOTO 200
100 WRITE(3,150) IND,IER
150 FORMAT(/,'IN JET IND= ',I5,'  IER= ',I5,/)
200 RETURN
    END
```

```
      SUBROUTINE LINTRP(X,XP,YP,SLOPE,N,P,PD,IER)
C
C*************************************************************************
C                                                                        *
C    SUBROUTINE LINTRP WAS WRITTEN FOR THE JOINT INSTITUTE FOR AERONAUTICS    *
C AND ACOUSTICS AT STANFORD UNIVERSITY BY THOMAS LUND.  LATEST REVISION 9     *
C OCTOBER 1984.                                                          *
C                                                                        *
C    THIS SUBROUTINE USES SLOPES GENERATED BY SUBROUTINE LNSPLN TO FIND       *
C INTERPOLATED VALUES OF A FUNCTION AND ITS DERIVATIVE AT ANY  STATION X.     *
C                                                                        *
C                                                                        *
C  **PARAMETER DESCRIPTION**                                             *
C                                                                        *
C     INPUTS:                                                            *
C  X  - INDEPENDENDENT COORDINATE.  X MUST BE WITHIN THE RANGE OF WHICH WAS   *
C       SENT TO SUBROUTINE LNSPLN.                                       *
C  XP - VECTOR OF LENGTH N CONTAINING THE X CORDINATES OF A FUNCTION P.      *
C  YP - VECTOR OF LENGTH N CONTAINING THE  VALUE OF P AT X STATIONS          *
C       CORRESPONDING TO THOSE IN XP.                                    *
C  SLOPE - VECTOR OF SLOPES AS OBTAINED FROM A CALL TO SUBROUTINE LNSPLN.    *
C  N  - NUMBER OF DATA POINTS USED IN THE SPLINE FIT (DIMENSION OF VECTORS XP *
C       AND YP)                                                          *
C                                                                        *
C     OUTPUTS:                                                           *
C  P  - INTERPOLATED VALUE OF THE FUNCTION AT THE STATION X              *
C  DP - INTERPOLATED VALUE OF THE FIRST DERIVATIVE OF THE FUNCTION AT THE    *
C       STATION X                                                        *
C  IER - ERROR PARAMETER, ON SUCCESSFUL TERMINATION IER IS SET TO ZERO, IER=1 *
C        INDICATES THAT X WAS OUT OF BOUNDS OF THE                       *
C        SPLINE FIT SLOPES.                                              *
C                                                                        *
C                                                                        *
C  **PRECISION** - ALL PARAMETERS AND INTERNAL VARIABLES ARE DOUBLE PRECISION *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XP(N),YP(N),SLOPE(N-1)
      IER=0
      NF=N-1
C
C        *** VERIFY THAT X IS WITHIN THE PROPER RANGE ***
C        *** EPS IS USED AS A TOLLERANCE FOR ROUND-OFF ERROR ***
C
      EPS=1.0D-6
      IF((XP(1)-X).GT.EPS.OR.(X-XP(N)).GT.EPS) THEN
         IER=1
         RETURN
      END IF
C
C        *** SEARCH THROUGH THE ABSCISSA VECTOR TO LOCATE THE INTERVAL IN ***
```

```
C        *** WHICH X LIES. ***
C
      DO 10 J=1,NF
         IF(J.EQ.NF) GOTO 20
         IF((XP(J)-EPS).LE.X.AND.X.LT.XP(J+1)) GOTO 20
10    CONTINUE
C
C        *** COMPUTE INPERPOLATED VALUES ***
C
20    D=X-XP(J)
      P=YP(J)+D*SLOPE(J)
      PD=SLOPE(J)
      RETURN
      END
```

```
      SUBROUTINE LNSPLN(X,Y,N,SLOPE,IER)
C*****************************************************************************
C                                                                              *
C    THIS SUBROUTINE WAS WRITTEN FOR THE JOINT INSTITUTE FOR AERONAUT-         *
C ICS AND ACOUSTICS, STANFORD UNIVERSTY BY THOMAS LUND.  LATEST REVIS-         *
C ION 13 SEPTEMBER 1984.                                                       *
C                                                                              *
C    SUBROUTINE LNSPLN (LINEAR SPLINE FIT) IS USED TO GENERATE THE             *
C SLOPE OF A DESCRETE FUNCTION THROUGH THE USE OF LINEAR SEGMENTS. THE         *
C SLOPF AT THE MIDPOINT OF EACH INTERVAL IS COMPUTED USING FIRST               *
C ORDER ACCURATE BACKWARD DIFFERENCING.  SUBROUTINE LINTRP IS CALLED           *
C TO DO THE ACTUAL INTERPOLATING.                                              *
C                                                                              *
C  **PARAMETER DESCRIPTION**                                                   *
C                                                                              *
C    INPUT:                                                                    *
C    X -  VECTOR OF LENGTH N CONTAINING THE ABSCISSI.  THE ELEMENTS OF         *
C         X MUST BE ORDERED SUCH THAT X(I+1)>X(I).                             *
C    Y  - VECTOR OF LENGTH N CONTAINIGN THE ORDINATES.                         *
C    N  - LENGTH OF THE INPUT VECTORS.  N MUST BE GREATER THAN ONE.            *
C                                                                              *
C    OUTPUT:                                                                   *
C SLOPE - VECTOR OF LENGTH N-1 CONTAINING THE SLOPE OF EACH INTERVAL           *
C   IER - ERROR PARAMETER.  ON NORMAL EXIT IER IS SET TO ZERO.  IER=1          *
C         INDICATES THAT N WAS LESS THAN 2.  IER=2 INDICATES THAT              *
C         X(I+1)<=X(I).                                                        *
C                                                                              *
C  **LINKING** - NO EXTERNAL SUBROUTINES TO LINK.                              *
C                                                                              *
C  **PRECISION** - ALL PARAMETERS AND INTERNAL VARIABLES ARE DOUBLE            *
C                  PRECISION.                                                  *
C                                                                              *
C*****************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N),Y(N),SLOPE(N-1)
      NF=N-1
C     CHECK FOR ERROR CONDITIONS
      IER=0
      IF(N.LT.2) THEN
         IER=1
         GOTO 200
      END IF
      DO 10 I=1,NF
         IF(X(I+1).LE.X(I)) THEN
            IER=2
            GOTO 200
         END IF
10    CONTINUE
C     COMPUTE FIRST ORDER ACCURATE SLOPES
      DO 20 I=1,NF
         SLOPE(I)=(Y(I+1)-Y(I))/(X(I+1)-X(I))
20    CONTINUE
```

```
200   RETURN
      END
```

```
      SUBROUTINE MATRIX(H,HD,B,UO,U1,A,T)
C
C*************************************************************************
C                                                                        *
C                                                                        *
C  THIS ROUTINE COMPUTES THE MATRIX ELEMENTS R.H.S. OF THE SYSTEM:       *
C                                                                        *
C          ! A(1,1) A(1,2) A(1,3) A(1,4) ! ! DUO/DX !       ! T(1) !    *
C          ! A(2,1) . . . . . A(2,4) ! ! DU1/DX !       ! T(2) !    *
C          ! A(3,1) . . . . . A(3,4) ! !  DB/DX !   =   ! T(3) !    *
C          ! A(4,1) . . . . . A(4,4) ! !  DP/DX !       ! T(4) !    *
C                                                                        *
C  THE FIRST THREE EQNS ARE THE FIRST, SECOND AND THIRD MOMENT OF THE    *
C  MOMENTUM EQUATION.  THE LAST EQN IS CONSERVATION OF MASS.  THE UNKNOWNS*
C  ARE THE X-DERIVATIVES OF UO, U1, B AND P, WHERE THE FIRST THEREE ARE  *
C  DEFINED BELOW AND P IS THE PRESSURE DEVIDED BY THE DENSITY.  NO       *
C  NONDIMENSIONALISATIONS ARE ASSUMED.                                   *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  H  - UPPER LIMIT OF INTEGRATION IN THE MOMENTS OF THE MOMENTUM EQN.  FOR *
C       A NON-CONFINED JET H IS A MULTIPLE OF B S.T. THE STRESS IS NEGLIGIBLE *
C       AT THAT DISTANCE. FOR A CONFINED JET, H IS THE HALF WIDTH OF THE  *
C       CONFINING CHANNEL.                                               *
C  HD - REPRESENTS DH/DX AND IS ONLY NEEDED IN CONFINED JETS, SINCE IT ENTERS *
C       ONLY IN LAST EQN. FOR NON-CONFINED JETS IT CAN BE LEFT UNDEFINED.  *
C  B  - CHARACTERISTIC HALF WIDTH OF JET.                                *
C  UO - SEE BELOW                                                        *
C  U1 - PARAMETERS DEFINING VELOCITY PROFILE IN EXPRESSION:              *
C               U = UO + U1*EXP(-0.693 * Y**2/B**2)                      *
C  DEL- FRACTION OF B (USUALLY DEL=0.78*B), WHERE THE LINEAR DECREASE OF EDDY *
C       VISCOSITY BEGINS.                                                *
C  ETA- MULTIPLE OF B WHERE EDDY VISCOSITY VANISHES (USUALLY ETA = 4.8*B). *
C  EPS- SCALING CONSTANT IN EDDY VISCOSITY (USUALLY EPS = 0.0283)        *
C                                                                        *
C    OUTPUT:                                                           *
C  A(I,J) - MATRIX ELEMENTS                                              *
C  T(I)   - RIGHT HAND SIDE                                              *
C                                                                        *
C  CAUTION ON DEL AND ETA:                                               *
C                   IF ETA > H, SET ETA = H.                             *
C                   IF ALSO DEL > H, SET DEL = H - (SMALL AMOUNT)        *
C  THE SMALL AMOUNT IS A VERY SMALL FRACTION OF H.  THIS CHANGE IS NEEDED TO *
C  AVOID THE DIVISIONS BY ZERO THAT WOULD ARISE IF BOTH DEL AND ETA WERE  *
C  EQUAL TO H.  ALL THESE CHANGES ARE TO BE DONE EXTERNALLY.             *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(4,4),T(4)
      PI=3.141592D0
```

```
      AL=0.693D0
      DEL=.78*B
      ETA=4.8*B
      EPS=.0283
      IF(ETA.GT.H) ETA=H
      IF(DEL.GT.H) DEL=H-.001
      VIS=EPS*B*U1
      H2=H**2
      H3=H**3
      B2=B**2
      B3=B**3
      U12=U1**2
      AL2=2*AL
      AL4=4*AL
      ER1=DSQRT(PI/AL)*DERF(DSQRT(AL)*H/B)
      ER2=DSQRT(PI/AL2)*DERF(DSQRT(AL2)*H/B)
      ERDEL=DSQRT(PI/AL)*DERF(DSQRT(AL)*DEL/B)
      ERETA=DSQRT(PI/AL)*DERF(DSQRT(AL)*ETA/B)
      EX1=DEXP(-AL*H2/B2)
      EX2=DEXP(-AL2*H2/B2)
      EXDEL=DEXP(-AL*DEL**2/B2)
      EXETA=DEXP(-AL*ETA**2/B2)
      UH=UO+U1*EX1
      AUX1=(1-EX1)/AL2
      AUX2=(1-EX2)/AL2
      AUX3=EX1*(1+AL*H2/B2)
      AUX4=EX1*(H/AL2/B+H3/3/B3)
      AUX5=EXDEL*(1+AL*DEL**2/B2)
      AUX6=EXETA*(1+AL*ETA**2/B2)
C
C        *** COMPUTE MATRIX ELEMENTS ***
C
      A(1,1)=2*H*UO+U1*B*ER1-UH*H
      A(1,2)=(UO-UH/2)*B*ER1+U1*B*ER2
      A(1,3)=(2*U1*UO-UH*U1)*(ER1/2-H*EX1/B)+U12*(ER2/2-H*EX2/B)
      A(1,4)=H
      A(2,1)=UO*H2/2+U1*B2*AUX1+U1*B2/AL*(1-AUX3)
      A(2,2)=UO*B2*AUX1+U1*B2*AUX2+U1/2*B*ER1*(B*ER1/4-H*EX1)
      A(2,3)=U1*UO*B/AL*(1-AUX3)+U12/2*(B*ER1**2/4-H*EX1*ER1+B*AUX2)
      A(2,4)=H2/2
      A(3,1)=UO*H3/3+U1*B2/AL2*(B*ER1/2-H*EX1)
     1       +U1*3*B3*(ER1/AL4-AUX4)
      A(3,2)=UO*B2/AL2*(B*ER1/2-H*EX1)+U1*B2/AL2
     1       *(B*ER2/2-H*EX2)+U1*B3/AL2*(ER2-ER1*AUX3)
      A(3,3)=U1*UO*3*B2*(ER1/AL4-AUX4)
     1       +U12*B2*(-ER1/AL2*AUX3+5*ER2/(8*AL)-H*EX2/AL4/B)
      A(3,4)=H3/3
      A(4,1)=H
      A(4,2)=B*ER1/2
      A(4,3)=U1*ER1/2-U1*H*EX1/B
      A(4,4)=0
C
```

```
C        *** RIGHT HAND SIDE ELEMENTS, ASSUMES THAT STRESS AT ***
C        *** DISTANCE H IS ZERO.                               ***
C
      T(1)=0
      T(2)=VIS*U1*((1-EXDEL)+ETA/(ETA-DEL)*(EXDEL-EXETA))
     1  -VIS*U1*(B*ERETA/2-B*ERDEL/2+DEL*EXDEL-ETA*EXETA)/(ETA-DEL)
      T(3)=VIS*U1*(B*ERDEL-2*DEL*EXDEL)+VIS*U1*ETA/(ETA-DEL)
     1    *(B*ERETA-B*ERDEL+2*DEL*EXDEL-2*ETA*EXETA)
     2    +2*VIS*U1*B2/AL/(ETA-DEL)*(AUX6-AUX5)
      T(4)=-HD*UH
      RETURN
      END
```

```
      SUBROUTINE PANVLC(XI,ALPHA,D,Q,N)
C
C*************************************************************************
C                                                                        *
C     SUBROUTINE PANVLC COMPUTES VALUES OF THE VELOCITY COMPONENTS AT THE JET *
C BOUNDARY (CONDITION EXTERNAL DENOTED BY APPENDED E).  THIS SUBROUTINE MAKES *
C REPEATED CALLS TO SUBROUTINE FLDVEL WHERE THE VELOCITY COMPONENTS ARE      *
C CALCULATED.  THE VELOCITY COMPONENTS ARE SENT TO SUBROUTINE JET VIA COMMON  *
C IN AREA1.                                                                  *
C                                                                            *
C   *** PARAMETER DESCRIPTION ***                                            *
C                                                                            *
C     INPUT:                                                                 *
C  XI     - COORDINATES OF THE CONTROL POINT LOCATIONS STRORED AS X,Y PAIRS   *
C  ALPHA  - VECTOR CONTAINING THE SURFACE SLOPES                             *
C  D      - VECTOR CONTAINING THE PANEL LENGHTS                              *
C  Q      - VECTOR CONTAINING THE SOURCE STRENGTHS                           *
C  N      - NUMBER OF PANELS                                                 *
C                                                                            *
C     OUTPUT:                                                                *
C  XE     - VECTOR CONTAINING THE ABSCISSA OF THE STATIONS AT WHICH THE      *
C           VELOCITIES ARE CALCULATED                                        *
C  YE     - VECTOR CONTAINING THE ORDINATE OF THE STATIONS AT WHICH THE      *
C           VELOCITIES ARE CLACULATED                                        *
C  UE     - VECTOR CONTAINING THE HORIZONTAL COMPONENT OF VELOCITY           *
C  VE     - VECTOR CONTAINING THE VERTICAL COMPONENT OF VELOCITY             *
C  ALL OUTPUTS ARE PASSED TO SUBROUTINE JET VIA COMMON IN AREA1             *
C                                                                           *
C*************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XI(N,2),ALPHA(N),D(N),Q(N)
      COMMON /AREA1/ XE(40),YE(40),UE(40),VE(40),SPLN(150),NE
      COMMON /AREA4/ PATM
      COMMON /AREA8/ NJS,NJF
      COMMON /AREA11/ XO
C
C        *** CALCULATE AND STORE VELOCITY COMPONENTS ***
C
      NS=NJS
      NF=NJF
      NE=NF-NS+1
      DO 10 I=NS,NF
         X=XI(I,1)
         Y=XI(I,2)
         CALL FLDVEL(XI,ALPHA,D,Q,N,X,Y,U,V)
         XE(I-NS+1)=X
         YE(I-NS+1)=Y
         UE(I-NS+1)=U
         VE(I-NS+1)=V
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PARMIN(U1O,VO,BETA)
C
C*************************************************************************
C                                                                        *
C     THIS SUBROUTINE READS INPUTS FROM DATA FILE PARAM.DAT.  THE INFORMATION *
C AQUIRED PORTAINS TO THE DETAILS OF THE SHROUD BODY AS WELL AS THE FLOW  *
C CONDITIONS.                                                            *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C     OUTPUT:                                                            *
C  U1O  - JET INITIAL CENTERLINE VELOCITY                                *
C  VO   - FREE-STREAM VELOCITY                                           *
C  BETA - ANGLE OF ATTACK                                                *
C                                                                        *
C  *** PASSED IN COMMON ***                                              *
C  XO   - X COORDINATE OF THE JET NOZZLE POSITION                        *
C  XC   - X COORDINATE OF THE CONTROL STATION                            *
C  XEXIT- X COORDINATE OF THE SHROUD END                                 *
C  NJS  - PANEL NUMBER AT WHICH THE JET BEGINS                           *
C  NJF  - PANEL NUMBER AT CHICH THE JET ENDS                             *
C  NLS  - PANEL NUMBER AT WHICH THE NOSE LIP STARTS                      *
C  NLF  - PANEL NUMBER AT WHICH THE NOSE LIP ENDS                        *
C                                                                        *
C*************************************************************************
C
      COMMON /AREA8/ NJS,NJF
      COMMON /AREA9/ NLS,NLF
      COMMON /AREA10/ XC
      COMMON /AREA11/ XO
      COMMON /AREA12/ XEXIT
      READ(2,10) XO,XC,XEXIT,NJS,NJF,NLS,NLF,VO,BETA,U1O
10    FORMAT(3F10.5,4I4,/,2F10.4,/,F10.4)
      RETURN
      END
```

```
      SUBROUTINE PERFRM(XI,ALPHA,D,Q,N,U1O,PHIS)
C
C*************************************************************************
C                                                                        *
C     THIS SUBROUTINE COMPUTES THE THRUST AUGMENTATION RATIO IN TWO      *
C INDEPENDENT CALCULATIONS; BY INTEGRATION OF THE SURFACE PRESSURES, AND BY A *
C CONTROL VOLUME ANALYSIS USING THE BLASIUS MOMENTUM THEOREM.  A SUMMARY OF  *
C THE PERFORMANCE PARAMETERS ARE WRITTEN TO THE OUTPUT FILE PARAM.DAT.      *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  XI     - COCRDINATES OF THE CONTROL POINTS STORED AS X,Y PAIRS        *
C  ALPHA  - VECTOR CONTAINING THE SURFACE SLOPES FOR EACH PANEL          *
C  D      - VECTOR CONTAINING THE PANEL LENGTHS                          *
C  Q      - VECTOR CONTAINING THE SOURCE STRENGTHS                       *
C  N      - NUMBER OF PANELS                                             *
C  U1O    - JET INITIAL CENTERLINE VELOCITY                              *
C                                                                        *
C     OUTPUT:                                                            *
C  PHIS   - THRUST AUGMENTATION AS COMPUTED THROUGH INTEGF ?ON OF THE SURFACE *
C           PRESSURE                                                     *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XI(N,2),ALPHA(N),D(N),Q(N)
      COMMON /UNIF/ VO
      COMMON /AREA4/ PATM
      COMMON /AREA7/ S(4)
      COMMON /AREA9/ NLS,NLF
      COMMON /AREA10/ XC
      COMMON /AREA11/ XO
      ALP=-DLOG(.5D0)
      BO=0.01
      PI=3.1415926
C
C        *** COMPUTE INVISCID FLUID SPEED AT THE JET NOZZLE     ***
C        *** USING THE BERNOULLI RELATION.  THE STATIC PRESSURE ***
C        *** AT THE NOZZLE VANISHES BY CONSTRUCTION             ***
C
C
      UO=DSQRT(2.0D0*PATM)
C
C        *** COMPUTE THE MOMENTUM FLUXES.  XMJ IS THE MOMENTUM FLUX ***
C        *** OF THE PRIMARY JET, XMI IS THE MOMENTUM FLUX ACROSS THE***
C        *** INLET BOUNDARY OF THE CONTROL VOLUME, AND XME IS THE   ***
C        *** MOMENTUM FLUX ACROSS THE EXIT BOUNDARY OF THE CONTROL  ***
C        *** VOLUME                                                 ***
C
      XMJ=U1O*U1O*BO*DSQRT(PI/2./ALP)
      XMI=2.0D0*BO*U1O*(2.1289D0*UO+0.7527D0*U1O)+VO*VO
```

```
      XME=2.*S(1)*S(1)+4.*S(1)*S(2)*(S(3)/2.*DSQRT(PI/ALP)
     &    *DERF(DSQRT(ALP)/S(3)))+2.*S(2)*S(2)*(S(3)/2.*DSQRT(PI/2./ALP)
     &    *DERF(DSQRT(2.*ALP)/S(3)))
      WRITE(4,35) XMJ,XMI,XME
35    FORMAT(' JET MOMENTUM FLUX = ',F10.4,/,' ENTERING MOMENTUM',
     &       'FLUX = ',F10.4,/,' EXITING MOMENTUM FLUX = ',F10.4)
      T=XME-XMI
C
C        *** COMPUTE THE THRUST AUGMENTATION RATION USING THE ***
C        *** MOMENTUM THEOREM                                 ***
C
      PHI=(XMJ+T)/XMJ
      WRITE(4,40) PHI
40    FORMAT(/,' AUGMENTATION RATIO COMPUTED USING THE MOMENTUM',
     &         ' THEOREM',F12.8)
      NS=NLS
      NF=NLF
      SUM=0.
C
C        *** INTEGRATE THE SURFACE PRESSURES ***
C
      DO 50 I=NS,NF
      X=XI(I,1)
      Y=XI(I,2)
      CALL FLDVEL(XI,ALPHA,D,Q,N,X,Y,U,V)
      SUM=SUM+(U*U+V*V)*D(I)*SIN(ALPHA(I))
50    CONTINUE
C
C        *** SUBTRACT THE BASE PRESSURE IN ORDER TO OBTAIN ***
C        *** THE CORRECT THRUST                           ***
C
      T=SUM-VO*VO
C
C        *** COMPUTE THE THRUST AUGMENTATION RATIO THROUGH ***
C        *** INTEGRATION OF THE SURFACE PRESSURES          ***
C
      PHIS=(T+XMJ)/XMJ
      WRITE(4,60) PHIS
60    FORMAT(' AUGMENTATION RATIO COMPUTED FROM SURFACE PRESSURES',
     &       F12.8,//)
      RETURN
      END
```

```
      SUBROUTINE RK2(N,FCN,X,Y,XEND)
C*****************************************************************************
C                                                                              *
C   THIS ROUTINE WAS WRITTEN FOR THE JOINT INSTITUTE FOR AERONAUTICS           *
C AND ACOUSTICS, STANFORD UNIVERSITY BY THOMAS LUND.  LATEST REVISION          *
C 20 JAN 1985.                                                                 *
C                                                                              *
C   SUBROUTINE RK2 INTEGRATES A FIRST ORDER SYSTEM OF ORDINARY DIFFER-         *
C ENTIAL EQUATIONS USING A SECOND ORDER ACCURATE RUNGE-KUITA SCHEME.           *
C EACH CALL TO RK2 ADVANCES THE SOLUTION FOREWARD IN TIME ONE INTERVAL.        *
C                                                                              *
C  ***PARAMETER DESCRIPTION***                                                 *
C                                                                              *
C  N  -  RANK OF THE FIRST ORDER SYSTEM.                                       *
C  FCN - N-DIMENSIONAL FUNCTION WHICH DEFINES THE SYSTEM DERIVATIVE.           *
C  X   - INDEPENDENT VARIABLE, INITIAL VALUE FOR INTEGRATION STEP.             *
C  Y   - VECTOR OF LENGTH N WHICH ON INPUT CONTAINS THE INITIAL VALUES         *
C        AND ON OUTPUT CONTAINS THE APPROXIMATE SOLUTION ADVANCED IN           *
C        TIME ONE INTERVAL.                                                    *
C XEND - VALUE OF THE INDEPENDENT VARIABLE AT THE END OF THE INTERVAL.         *
C        THE INTERVAL SIZE IS DEFINED AS XEND-X.                               *
C                                                                              *
C  ***PRECISION***                                                             *
C                                                                              *
C     ALL PARAMETERS AND VARIABLES ARE DEFINED AS DOUBLE PRECISION             *
C                                                                              *
C  ***ENVIRONMENT***                                                           *
C                                                                              *
C     VAX 11-780                                                               *
C                                                                              *
C*****************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(N),YP(10),YHAT(10),YHATP(10)
      H=XEND-X
      CALL FCN(N,X,Y,YP)
      DO 10 I=1,N
         YHAT(I)=Y(I)+H*YP(I)
10    CONTINUE
      CALL FCN(N,XEND,YHAT,YHATP)
      DO 20 I=1,N
         Y(I)=0.5D0*(Y(I)+YHAT(I)+H*YHATP(I))
20    CONTINUE
      X=XEND
      RETURN
      END
```

```
      SUBROUTINE SIMQ(AD,A,B,N,ND,KS)
C
C*****************************************************************************
C                                                                             *
C     SUBROUTINE SIMQ IS AN OLD IBM SYSTEM USED TO SOLVE A SYSTEM OF          *
C SIMULTANEOUS LINEAR EQUATIONS.  THE ALGORITHM IS GAUSSIAN ELIMINATION.      *
C                                                                             *
C   *** PARAMETER DESCRIPTION ***                                             *
C                                                                             *
C     INPUT:                                                                  *
C  AD - MATRIX OF COUPLING COEFFICIENTS                                       *
C  A  - WORK SPACE MATRIX OF DIMENSION IDENTICAL TO THAT OF AD                *
C  B  - RIGHT HAND SIDE VECTOR                                                *
C  N  - RANK OF THE SYSTEM                                                    *
C  ND - NUMBER OF EQUATIONS IN THE SYSTEM (USUALLY EQUAL TO N)                *
C  KS - ERROR PARAMETER, KS=1 FOR A SINGULAR MATRIX                           *
C                                                                             *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION B(ND),AD(ND,ND),A(1)
      IJ=0
      DO 130 K=1,N
      DO 130 L=1,N
      IJ = IJ+1
 130  A(IJ) = AD(L,K)
 132  TOL=0.0
       KS=0
       JJ=-N
       DO 65 J=1,N
       JY=J+1
       JJ=JJ+N+1
       BIGA=0
       IT=JJ-J
       DO 30 I=J,N
       IJ=IT+I
       IF(DABS(BIGA)-DABS(A(IJ))) 20,30,30
    20 BIGA=A(IJ)
       IMAX=I
    30 CONTINUE
       IF(DABS(BIGA)-TOL) 35,35,40
    35 KS=1
       GO TO 220
    40 I1=J+N*(J-2)
       IT=IMAX-J
       DO 50 K=J,N
       I1=I1+N
       I2=I1+IT
       SAVE=A(I1)
       A(I1)=A(I2)
       A(I2)=SAVE
    50 A(I1)=A(I1)/BIGA
```

```
      SAVE=B(IMAX)
      B(IMAX)=B(J)
      B(J)=SAVE/BIGA
      IF(J-N) 55,70,55
   55 IQS=N*(J-1)
      DO 65 IX=JY,N
      IXJ=IQS+IX
      IT=J-IX
      DO 60 JX=JY,N
      IXJX=N*(JX-1)+IX
      JJX=IXJX+IT
   60 A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))
   65 B(IX)=B(IX)-(B(J)*A(IXJ))
   70 NY=N-1
      IT=N*N
      DO 80 J=1,NY
      IA=IT-J
      IB=N-J
      IC=N
      DO 80 K=1,J
      B(IB)=B(IB)-A(IA)*B(IC)
      IA=IA-N
   80 IC=IC-1
  220  IF (N.EQ.ND) RETURN
      IJ = N*N+1
      DO 110 L=1,N
      DO 110 K=1,N
      IJ = IJ-1
  110 AD(N-L+1,N-K+1) = A(IJ)
      RETURN
      END
```

```
      SUBROUTINE SIZE(M)
C
C*************************************************************************
C                                                                        *
C     SUBROUTINE SIZE READS THE DATA SET BODY.DAT IN ORDER TO DETERMINE THE   *
C NUMBER OF ELEMENTSCONTAINED THERE.  THIS IS NECESSARY TO ALLOW RUN-TIME     *
C DIMENSIONING OF THE ARRAYS IN THE INPUTING SUBROUTINE DATIN.                *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     OUTPUT:                                                            *
C  M - NUMBER OF LINES IN DATA FILE BODY.DAT                             *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
10    FORMAT(3F10.4)
15    FORMAT(2F10.4)
      READ(1,15) DUM1,DUM2
      DO 20 I=1,110
         READ(1,10,END=60) DUM1,DUM2,DUM3
20    CONTINUE
60    M=I-1
      RETURN
      END
```

```
      SUBROUTINE SOLVE3(A,T,UOD,R)
C
C*****************************************************************************
C                                                                             *
C     SUBROUTINE SOLVE3 SOLVES THE SIMULTANEOUS LINEAR EQUATIONS NECESSARY TO *
C DETERMINE THE DERIVATIVES OF THE JET PARAMETERS FOR USE IN MARCHING THE     *
C JET SOLUTION IN THE VISCOUS-INVISCID REGION.                                *
C                                                                             *
C   *** PARAMETER DESCRIPTION ***                                             *
C                                                                             *
C     INPUT:                                                                  *
C  A  - COUPLING COEFFICIENT MATRIX                                           *
C  T  - RIGHT HAND SIDE VECTOR                                                *
C  UOD- DERIVATIVE OF THE EXTERNAL VELOCITY UO AS DETERMINED BY THE INVISCID  *
C       SOLUTION.                                                             *
C                                                                             *
C     OUTPUT:                                                                 *
C  R  - VECTOR CONTAINING THE DERIVATIVE VALUES FOR U1, B, AND P.             *
C                                                                             *
C*****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(4,4),T(4),B(3,3),P(3,3),R(3)
C
C        *** CREATE 3X3 SYSTEM FROM 4X4 INPUT BY USING EXTERNAL VELOCITY ***
C        *** DERIVATIVE UOD AS A FORCING TERM ***
C
      I=0
      M=4
      DO 10 L=1,4
      IF(L.EQ.M) GOTO 9
         I=I+1
         R(I)=T(L)-UOD*A(L,1)
         DO 5 J=1,3
            B(I,J)=A(L,(J+1))
5        CONTINUE
9        CONTINUE
10    CONTINUE
C
C        *** SOLVE SYSTEM USING SIMQ ***
C
      CALL SIMQ(B,P,R,3,3,IER)
      RETURN
      END
```

```
      SUBROUTINE STREN(XI,V,VN,ALPHA,D,W,N,P,VO,BETA,L)
C
C*************************************************************************
C                                                                        *
C     SUBROUTINE STREN COMPUTES THE PANEL SOURCE STRENGTHS.              *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  XI     - COORDINATES OF THE CONTROL STATION LOCATIONS STORED AS X,Y PAIRS  *
C  VN     - VECTOR CONTAINING THE TRANSPIRATION VELOCITY FOR EACH PANEL      *
C  ALPHA  - VECTOR CONTAINING THE SURFACE SLOPE ANGLES FOR EACH PANEL        *
C  D      - VECTOR CONTAINING THE PANEL LENGTHS                              *
C  W      - MATRIX OF AERODYNAMIC INFLUENCE COEFFICIENTS                     *
C  N      - NUMBER OF PANELS                                                 *
C  P      - WORK SPACE MATRIX                                                *
C  VO     - FREE-STREAM VELOCITY                                             *
C  BETA   - ANGLE OF ATTACK                                                  *
C  L      - PARAMETER WHICH SPECIFIES WHETHER OR NOT THE AERODYNAMIC INFLUENCE *
C           COEFFICIENTS ARE CALCULATED.  WHEN L=1 THE AERODYNAMIC INFLUENCE  *
C           COEFFICIENTS ARE CALCULATED, FOR L OTHER THAN 1 PREVIOUS VALUES   *
C           OF THE AERODYNAMIC INFLUENCE COEFFICIENTS ARE USED                *
C                                                                        *
C     OUTPUT:                                                            *
C  V      - VECTOR CONTAINING THE SOURCE STRENGTHS                           *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XI(N,2),V(N),VN(N),ALPHA(N),D(N),W(N,N),P(N,N)
      COMMON /DUMP/ DUMP1
      LOGICAL DUMP1
C
C        *** GENERATE THE MATRIX AND RIGHT HAND SIDE
C
      DO 10 I=1,N
         V(I)=VO*DSIN(ALPHA(I)-BETA)-VN(I)
         X=XI(I,1)
         Y=XI(I,2)
         IF(L.NE.1) GOTO 9
         DO 5 J=1,N
            CALL COEF(XI,X,Y,J,ALPHA,D,N,A,B)
            W(I,J)=B*DCOS(ALPHA(I))-A*DSIN(ALPHA(I))
5        CONTINUE
9     CONTINUE
10    CONTINUE
C
C        *** SOLVE THE SYSTEM USING SIMQ ***
C
      CALL SIMQ(W,P,V,N,N,IER)
      IF(.NOT.DUMP1) GOTO 50
      WRITE(3,20)
```

```
20    FORMAT(//,'SOURCE STRENGTHS',/)
      DO 40 I=1,N
         WRITE(3,30) I,V(I)
30       FORMAT(I2,5X,F10.4)
40    CONTINUE
50    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SURFVEL(XI,ALPHA,D,Q,N,SC,UEXT,NEXT,XLEN,STAG)
C
C*************************************************************************
C                                                                        *
C     THIS SUBROUTINE COMPUTES THE SHROUD SURFACE VELOCITY FROM THE INVISCID  *
C SOLUTION FOR USE IN THE BOUNDARY LAYER CALCULATION.                    *
C                                                                        *
C   *** PARAMETER DESCRIPTION ***                                        *
C                                                                        *
C     INPUT:                                                             *
C  XI     - COORDINATES OF THE CONTROL STATION LOCATIONS STORED AS X,Y PAIRS   *
C  ALPHA  - VECTOR CONTAINING THE SURFACE SLOPE ANGLE FOR EACH PANEL      *
C  D      - VECTOR CONTAINING THE PANEL LENGTHS                           *
C  Q      - VECTOR CONTAINING THE SOURCE STRENGHTS                        *
C  N      - NUMBER OF PANELS                                              *
C  SC     - VECTOR OF SURFACE COORDINATES AT WHICH THE VELOCITIES ARE      *
C           CALCULATED.  THE SURFACE COORDINATES ARE NORMALIZED SUCH THAT THE  *
C           CONTROL STATION LOCATION IS 1.  THE ORIGIN IS THE STAGNATION POINT *
C           IF A FREE-STREAM IS PRESENT AND THE SHROUD TRAILING EDGE FOR    *
C           STATIC OPERATION                                               *
C  UEXT   - VECTOR CONTAINING THE SURFACE VELOCITIES                      *
C  NEXT   - NUMER OF STATIONS AT WHICH THE VELOCITY IS CALCULATED         *
C  XLEN   - LENGTH OF THE SURFACE OVER WHICH THE THE VELOCITIES ARE CALCULATED *
C  STAG   - LOGICAL VARIABLE SET TO TRUE WHEN A STAGNETION POINT IS PRESENT    *
C                                                                        *
C*************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL STAG
      DIMENSION XI(N,2),ALPHA(N),D(N),Q(N)
      DIMENSION SC(100),UEXT(100)
      COMMON /AREA10/ XC
      COMMON /AREA12/ XEXIT
      LOGICAL FLAG
C
C        *** FIND PANEL INDEX OF SHROUD TRAILING EDGE ***
C
      DO 10 I=N,1, -1
      IF(XI(I-1,1).LT.XEXIT) GOTO 20
10    CONTINUE
20    NS=I
      NSJ=NS
C
C        *** FIND THE PANEL INDEX OF THE CONTROL STATION ***
C
      FLAG=.FALSE.
      DO 30 I=NS,1, -1
         IF(XI(I-1,1).GT.XI(I,1)) FLAG=.TRUE.
         IF(FLAG.AND.XI(I,1).GT.XC) GOTO 40
30    CONTINUE
40    NF=I+1
      NFJ=I
```

```
      K=0
C
C        *** STORE THE SURFACE COORDINATES AND COMPUTE THE ***
C        *** SURFACE VELOCITIES                             ***
C
      DO 100 I=NS,NF,-1
         IF(I.EQ.NS) THEN
            K=K+1
            S=XEXIT-XI(I,1)
            SC(1)=S
            X=XI(I,1)
            Y=XI(I,2)
            CALL FLDVEL(XI,ALPHA,D,Q,N,X,Y,U,V)
            UEXT(K)=DSQRT(U*U+V*V)
         ELSE
            S=S+D(I+1)/2.0D0+D(I)/2.0D0
C
C              *** FILTER THE VELOCITY DATA WHICH IS TAKEN IN A REGION ***
C              *** ADJACENT TO THE CONTROL STATION SINGULARITY.        ***
C
            X=XI(I,1)
            Y=XI(I,2)
            CALL FLDVEL(XI,ALPHA,D,Q,N,X,Y,U,V)
            UMOD=DSQRT(U*U+V*V)
            IF(S.LT.5.0) THEN
C
C                 *** INCLUDE THE LOCAL POINT ONLY IF THE ***
C                 *** VELOCITY IS INCREASING              ***
C
               IF(UMOD.GT.UEXT(K)) THEN
                  K=K+1
                  SC(K)=S
                  UEXT(K)=UMOD
               END IF
            ELSE
               K=K+1
               SC(K)=S
               UEXT(K)=UMOD
            END IF
         END IF
100   CONTINUE
C
C        *** SEARCH FOR THE STAGNATION POINT (MINIMUM VELOCITY MODULUS) ***
C
      UMIN=10.0D0
      DO 105, I=1,K
         IF(UEXT(I).LT.UMIN) THEN
            UMIN=UEXT(I)
            L=I
         END IF
105   CONTINUE
      IF(L.EQ.1) THEN
```

```
         STAG=.FALSE.
      ELSE
         STAG=.TRUE.
      END IF
C
C        *** CORRECT IF NOT ALL DATA IS FROM THE SAME SIDE OF THE ***
C        *** STAGNATION POINT                                     ***
C
      IF(STAG) THEN
         TEST=(UEXT(L+2)-UEXT(L+1))/(UEXT(L+1)-UEXT(L))
         IF(TEST.GT.10.0) L=L+1
      END IF
C
C        *** NORMALIZE SURFACE COORDINATES SKIPPING OVER POINTS SUFFERING   ***
C        *** FROM SINGULARITIES NEAR THE CONTROL STATION (LAST THREE POINTS) **
C
      NEND=(K-2)
      IF(STAG) THEN
         SO= SC(L) - (SC(L+1)-SC(L))*UEXT(L)/(UEXT(L+1)-UEXT(L))
         SC(1)=0.0D0
         UEXT(1)=0.0D0
         NEXT=K-L
         K=1
      ELSE
         SO= 0.0D0
         NEXT=NEND
         K=0
      END IF
      XLEN=SC(NEND)-SO
      DO 110 I=L,NEND
         K=K+1
         SC(K)=(SC(I)-SO)/XLEN
         UEXT(K)=UEXT(I)
110   CONTINUE
      RETURN
      END
```

End of Document