NASA TECHNICAL NOTE

NASA TN D-7602

NASA TN D-7602

COMPUTER PROGRAM FOR CALCULATING THE FLOW FIELD OF SUPERSONIC EJECTOR NOZZLES

by Bernhard H. Anderson

Lewis Research Center

Cleveland, Ohio 44135

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1974

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TN D-7602		
4. Title and Subtitle COMPUTER PROGRAM FOR CALCULATING THE FLOW FIELD OF SUPERSONIC EJECTOR NOZZLES		5. Report Date APRIL 1974 6. Performing Organization Code
7. Author(s) Bernhard H. Anderson		8. Performing Organization Report No. E-7476 10. Work Unit No. 501-24
9. Performing Organization Name and Address Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135		11. Contract or Grant No. 13. Type of Report and Period Covered Technical Note
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

An analytical procedure for computing the performance of supersonic ejector nozzles is presented. This procedure includes real sonic "line" effects and an interaction analysis for the mixing process between the two streams. The procedure is programed in FORTRAN IV and has operated successfully on IBM 7094, IBM 360, CDC 6600, and UNIVAC 1108.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Propulsion Aerodynamics	Unclassified - unlimited Cat. 28

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	87	$4.00

* For sale by the National Technical Information Service, Springfield, Virginia 22151

COMPUTER PROGRAM FOR CALCULATING THE FLOW FIELD OF SUPERSONIC EJECTOR NOZZLES

by Bernhard H. Anderson

Lewis Research Center

SUMMARY

An analytical procedure for computing the performance and flow-field characteristics of supersonic ejector nozzles is presented. This procedure includes real sonic "line" effects and an interaction analysis for the mixing process between the primary and secondary flows of the ejector nozzle. The procedure, which has been programmed in FORTRAN IV, is designated REJECT and has operated on IBM 7094, IBM 360, CDC 6600, and UNIVAC 1108.

INTRODUCTION

The potential of secondary flow to improve the performance of exhaust nozzles has received widespread attention. Numerous experimental as well as theoretical investigations have been conducted. The early efforts to analyze supersonic ejector nozzles were based on one-dimensional concepts (refs. 1 to 3). In these studies of ejector systems, one-dimensional isentropic relations were applied to both the primary and secondary flows, which were considered to coexist within a cylindrical shroud and allowed to have different average total pressures. Because such a treatment has limitations, in later analyses the primary flow was constructed by using the method of characteristics, while one-dimensional isentropic flow was assumed for the secondary stream (refs. 4 to 6). In these analyses, the dissipative effects of the mixing process were not treated as an interaction problem, but rather superimposed on the inviscid jet boundary in the classical boundary-layer approach. The phenomenological description of this transport mechanism was based on a quasi-constant-pressure two-stream turbulent mixing process (ref. 5) obtained by an extension of analyses for mixing between a single stream and a quiescent fluid (ref. 7). In application, the effects of mixing were viewed as a simple change in the secondary weight flow ratio from that given by the inviscid solution.

In general, this method yielded good agreement with data, provided the shroud (or shoulder) diameter ratio was small and the primary stream entrance Mach number was greater than 1.0 so that the flow followed a Prandtl-Meyer relation for the expansion process around the primary nozzle lip.

It became apparent that an interaction approach had to be used for the mixing process for many ejectors (ref. 8) because a correction of the inviscid solution underestimated the effects of the turbulent mixing process. A second and very pragmatic problem also occurred when extensive calculations were made with these analyses. The inviscid analysis would not yield a choked solution in the stream for many ejector conditions of interest. However, it was realized (ref. 8) that if the second-order displacement effects were included in an interaction analysis, these solutions could be obtained. The assumption that the flow follows a Prandtl-Meyer expansion around the primary nozzle lip from a uniform sonic flow was very limiting because practical ejector systems usually have choked conical primary nozzles. It was not until the solution of compressible flow through choked conical nozzles was accomplished (ref. 9) and incorporated into an ejector nozzle analysis (refs. 10 and 11) that the influence of the sonic line assumptions could be resolved. The computer program reported herein is designated REJECT and includes both the choked conical nozzle analysis reported in reference 9 and an interaction analysis for the mixing process between the primary and secondary flows within the ejector nozzle.

The theoretical analysis for the solution of an ejector nozzle flow field of arbitrary geometry is presented in this report. A comparison of the theory for a large number of test cases is presented in references 10 and 11. The symbols used in the analysis are defined in appendix A. To facilitate the use of the computer program, a detailed description of the input which is required to operate REJECT and an interpretation of the printed output are given in appendixes B and C. In addition, a sample output listing and the computer program listing are presented in appendixes D and E.

ANALYSIS

The flow in an ejector nozzle involves the mutual interaction between a high-energy, high-velocity primary stream and a low-energy, low-velocity secondary stream, as shown in figure 1. These two streams begin to interact at the primary nozzle lip. For the ejector operating in the supersonic flow regime, the secondary flow is effectively "sealed off" from ambient conditions. It is this nozzle operating condition that is considered in the theoretical analysis presented in this report.

The primary flow field is determined by the method of characteristics, starting from an initial datum line, called the sonic line. The viscous interaction between the two streams occurs along the interface (dash-dot line in fig. 1) and results in a transfer of energy from the primary stream to the secondary stream. In the present analysis the two-stream mixing is computed on the basis of a quasi-constant-pressure mixing process; that is, the velocity profile is assumed to be developed from the local two streams through a constant-pressure turbulent mixing process. Two conditions were applied at each point along the jet boundary or interface: (1) the local static pressure must be equal for both streams at their boundary and (2) continuity between the two streams must be preserved. The later condition required that the amount of secondary flow entrained by the mixing process plus the unmixed flow be equal to the secondary flow supplied to the ejector. These conditions were used to determine the jet boundary and consequently the local flow conditions in the neighborhood of the shroud wall. Thus, the mixing process was treated as an interaction analysis rather than simply superimposing the mixing region on the inviscid flow field at the minimum secondary flow area as had been done previously.

Sonic Line Solutions

Real sonic line. - The axisymmetric transonic flow analysis through conical nozzles presented in reference 9 forms the basis for the sonic line solution in program REJECT. To obtain the flow angle distribution in the throat region of the primary nozzle, the flow is assumed to be steady, irrotational, isentropic, and two dimensional. The motion of such a fluid can be described in terms of the stream function ψ by the equation

$$\left[1 - \left(\frac{\rho_0}{\rho}\right)^2 \frac{\psi_y^2}{a^2}\right]\psi_{xx} + \left[2\left(\frac{\rho_0}{\rho}\right)^2 \frac{\psi_x\psi_y}{a^2}\right]\psi_{xy} + \left[1 - \left(\frac{\rho_0}{\rho}\right)^2 \frac{\psi_x^2}{a^2}\right]\psi_{yy} = 0 \tag{1}$$

where the subscripts refer to partial differentiation. Equation (1) is a nonlinear partial differential equation of second order. Within the transonic region, this equation is of the mixed type, elliptic for subsonic flow and hyperbolic for supersonic flow. Upon introducing the hodograph variables defined by

$$q = \left(u^2 + v^2\right)^{1/2} \tag{2}$$

$$\vartheta = \tan^{-1}\left(\frac{v}{u}\right) \tag{3}$$

equation (1) becomes

$$\frac{\partial}{\partial q}\left(\frac{\rho_0}{\rho} q \frac{\partial \psi}{\partial q}\right) + \frac{\rho_0}{\rho}\frac{1}{q}\left(1 - \frac{q^2}{a^2}\right)\frac{\partial^2 \psi}{\partial \vartheta^2} = 0 \tag{4}$$

Further simplification can be made by introducing the transformed velocity defined by

$$d\omega = \frac{\rho}{\rho_0}\frac{dq}{q} \tag{5}$$

Thus, equation (4) becomes

$$\frac{\partial^2 \psi}{\partial \omega^2} + \kappa(M)\frac{\partial^2 \psi}{\partial \vartheta^2} = 0 \tag{6}$$

where

$$\kappa(M) = \left(\frac{\rho_0}{\rho}\right)^2 (1 - M^2) \tag{7}$$

The introduction of the tangent gas approximation $\kappa(M) = 1$ greatly simplifies the solution of equation (6) because the hodograph relations reduce to the Cauchy-Riemann equations. Thus, the flow field may be solved by the method of complex variables. The incompressible and compressible flows are thus related by equation (5), which becomes upon integration

$$\omega = \ln \frac{2\left(\frac{q}{q_m}\right)\frac{M_m}{\sqrt{1 - M_m^2}}}{1 + \sqrt{1 + \frac{\left(\frac{q}{q_m}\right)^2 M_m^2}{1 - M_m^2}}} \tag{8}$$

where q_m and M_m represent the velocity and Mach number at the match state where this analysis is applied. Because the Cauchy-Riemann conditions are satisfied, the method of singularities can be used in such a way as to satisfy the boundary conditions of the present problem. The complex potential function of an arrangement of sources and sinks which satisfies the necessary boundary conditions is given by

$$F(\omega - i\vartheta) = \ln \frac{\cosh \frac{\pi}{\alpha} (\omega - i\vartheta - \omega_j) - \cosh\left(\frac{\pi}{\alpha} \Delta\omega\right)}{\cosh \frac{\pi}{\alpha} (\omega - i - \omega_j) - 1} \tag{9}$$

where $\Delta\omega = \omega_j - \omega_a$. The complex velocity can thus be found by differentiating equation (9)

$$\frac{dF}{d(\omega - i\vartheta)} = \frac{\frac{\pi}{\alpha} \sinh \frac{\pi}{\alpha} (\omega - i\vartheta - \omega_j)}{\cosh \frac{\pi}{\alpha} (\omega - i\vartheta - \omega_j) - \cosh\left(\frac{\pi}{\alpha} \Delta\omega\right)} - \frac{\frac{\pi}{\alpha} \sinh \frac{\pi}{\alpha} (\omega - i\vartheta - \omega_j)}{\cosh \frac{\pi}{\alpha} (\omega - i\vartheta - \omega_j) - 1} \tag{10}$$

Use of this solution limits configurations to sharp-edged conical nozzles. The hodograph solution can be transformed to the physical plane by introducing the complex variable $z = x + iy$. Thus,

$$dz = \frac{1}{e^{\omega - i\vartheta}} \times \frac{dF}{d(\omega - i\vartheta)} \times d(\omega - i\vartheta) - \overline{\frac{e^{\omega - i\vartheta}}{4} \times \frac{dF}{d(\omega - i\vartheta)} \times d(\omega - i\vartheta)} \tag{11}$$

where the bar over the last term indicates the complex conjugate. Thus, the location of any point (ω, ϑ) in the hodograph plane can be transferred to the point (x, y) in the physical plane by the integration of equation (11). The numerical constants have been adjusted in equation (11) such that the location of the nozzle lip in the hodograph plane (ω_j, α) corresponds to the point $(0, 1)$ in the physical plane. Of particular interest in this analysis is the distribution of flow angle within the throat region. These lines of constant flow angle are called isoclines. To obtain the location of these isoclines, equation (11) is numerically integrated from ω_a to ω_j at selected values of ω for flow angles between 0 and α.

The sonic line is determined by finding the points of intersection of the established isoclines with the Mach lines originating at the nozzle lip, as shown in figure 2. To determine these Mach lines, the method of characteristics is introduced. Thus, the primary flow field is constructed by using the set of equations

$$\frac{dy}{dx} = \tan(\vartheta \pm \mu) \qquad (12)$$

$$\frac{dp}{\rho q^2 \tan \mu} \pm d\vartheta + \frac{j \sin \vartheta \sin \mu}{\sin(\vartheta \pm \mu)} \frac{dy}{y} = 0 \qquad (13)$$

$$dS = 0 \qquad \text{on } \frac{dy}{dx} = \tan \vartheta \qquad (14)$$

where μ is the local Mach angle. The static pressure p and flow angle ϑ were chosen as the basic variables because these quantities must be matched along the primary jet boundary, or slipstream. At the nozzle lip, the expansion is a centered-wave type and thus satisfies the Prandtl-Meyer function. The flow at the nozzle lip turns through an angle which is determined by the back pressure, or the pressure in the vicinity of the primary nozzle. The expansion is divided into a finite number of discrete steps. The first point on the sonic line is the point of intersection of the first isocline (fig. 2) and the Mach line from the nozzle lip which yields Mach 1.0 at the isocline flow angle. This intersection is determined through an iterative process which involves satisfying the known properties of pressure and flow angle along the sonic line and the characteristic equations. Subsequent points along the sonic line are constructed in the same manner as the characteristic net is developed. Thus, the sonic line is constructed as the primary flow field is developed.

While the analysis presented is for the choked flow through a wedge nozzle, the solution of flow through conical nozzles is obtained by using the simplifying assumption that the isoclines for axisymmetric flow are the same as those for the corresponding two-dimensional flow. Once the sonic line is constructed, the inviscid discharge and velocity coefficients can be obtained from the expressions

$$C_{D_i} = \int (\cos \vartheta y \, dy - \sin \vartheta y \, dx) \qquad (15)$$

$$C_{V_i} = \frac{1}{C_{D_i}} \int \cos \vartheta(\cos \vartheta y \, dy - \sin \vartheta y \, dx) \qquad (16)$$

integrated along the sonic line.

Conical sonic line. - The present analysis constructs a primary inlet flow sonic line for an ejector with a centerbody by assuming that flow within the throat region satisfies the Taylor-Maccoll flow represented by the equation

$$vv_{uu} = 1 + v_u^2 - \frac{(u + vv_u)^2}{1 - \frac{\gamma - 1}{\gamma + 1}(u^2 + v^2)} \tag{17}$$

where the subscripts refer to partial differentiation. Equation (17) is integrated from the centerbody surface to the primary nozzle lip. The boundary conditions on the plug surface are given by

$$\left.\begin{aligned} u &= q_m \cos \vartheta \\ v &= q_m \sin \vartheta \\ v_u &= -\frac{u}{v} \end{aligned}\right\} \tag{18}$$

where q_m is the velocity of the match condition, which is usually chosen to be 1.003, and ϑ is the half-angle of the plug surface in the vicinity of the primary nozzle exit. The sonic line would therefore be represented as a Mach wave passing through the primary nozzle lip and intersecting the ejector centerbody. The conditions along that starting line are obtained from a solution of equation (17). Construction of the sonic line by using the Taylor-Maccoll equation provides a flow that will not compress to subsonic conditions on the plug surface downstream of the throat. It also satisfies the condition that the velocity at the plug surface be parallel to that surface, but the condition of a specified primary lip angle cannot be satisfied.

Plane sonic line. - The plane sonic line starting datum is constructed by using equations (12) to (14) at the match Mach number M_m. Along this datum line, the flow angle distribution does not vary. Thus, the only physically real case it can represent is that of flow discharging axially from a nozzle with a zero lip angle.

Primary Flow Field Solution

Because the method of characteristics represents the solution of a hyperbolic differential equation, downstream boundary conditions are not required. The solution marches downstream using only the known upstream values and boundary conditions. Disturbances in the flow field are not propagated upstream and do not affect regions of the flow which have already been calculated. The actual computational procedure varies somewhat depending on whether the new point to be calculated is an interior or boundary point. However, the set of equations (12) to (14) must be satisfied. The numerical technique used in constructing the primary flow field was structured as a marching technique. Only the conditions on two C+ characteristics are saved (fig. 2), and the computations always proceed in the downstream direction. This technique was chosen over a reference plane method (ref. 12) because the flow under many ejector operating conditions goes subsonic in the region near the centerline of the nozzle exit. Therefore, the complete pressure distribution on the shroud surface can be obtained without dealing with a subsonic region.

In the method-of-characteristics solution, a shock wave is formed when two or more members of the same characteristic family intersect. This phenomenon always occurs in an ejector exhaust nozzle at some point in the flow field as a direct result of the recompression process which follows the overexpansion of the flow at the primary nozzle lip. In this computer program, when two or more characteristics of the same family do intersect, the conditions are set equal to the average conditions at the intersection point. These average conditions are projected downstream to the next solution plane and thus form a weak shock wave. No attempt is made to account for entropy losses under these conditions.

When the static pressure at the free boundary is constant, the Mach number at this boundary is only a function of pressure. The numerical procedure for solving the free boundary under this condition for the method of characteristics is well established and will not be discussed. For flow in ejector nozzles the primary jet boundary is not at constant pressure. Thus, the conditions along this interface depend on the ejector mass flow ratio, the total pressure ratio, the primary nozzle geometry, the shroud geometry, and the mixing process along the jet boundary. A detailed discussion of this dependence appears in the section Secondary Flow Field Solutions. It suffices to say that once the pressure on the jet boundary is determined, the iteration for the location of the primary jet boundary proceeds as if it were at constant pressure.

Centerbody boundary layer. - The iteraction between the primary flow field and the boundary layer along the centerbody surface is taken into account in program REJECT by the classical method of patching the boundary layer and inviscid flow field solutions. Both the inviscid primary field and the boundary layer are computed simultaneously so that the "marching" procedure is maintained. The boundary layer parameters are calculated by the method presented in reference 13. This method was chosen because it provides working formulas which are simple to program and which reflect boundary layer behavior to a remarkably good degree, provided the Reynolds number is large and the flow remains attached. The technique for including the boundary layer in the characteristic solution proceeds in the same way as the standard boundary solution except the physical surface is displaced an amount δ^* and the surface tangent is increased an amount $d\delta^*/dx$. With the method presented in reference 13, the boundary layer properties are computed on the basis of an equivalent length defined by

$$X = \left(Gy^{\beta}\right)^{-1} \int_0^x \left(Gy^{\beta}\right) dx \tag{19}$$

where

$$G = \left(\frac{M}{1 + 0.2M^2}\right)^4 \tag{20}$$

and β takes on values of either 1.20 or 1.25 depending on the Reynolds number. The local Reynolds number R_X based on the equivalent length defined by equation (19) is computed by using the stagnation pressure and temperature. With this local Reynolds number R_X, the displacement thickness δ^* and momentum thickness θ are computed for R_X of order 10^6 by using the expressions

$$\delta^* = 0.046 \times \left(1.0 + 0.8M^2\right)^{0.44} R_X^{-0.20} \tag{21}$$

$$\theta = 0.036 \times \left(1.0 + 0.10M^2\right)^{-0.70} R_X^{-0.20} \tag{22}$$

and for R_X of order 10^7 by using the expressions

$$\delta^* = 0.028 \times \left(1.0 + 0.8M^2\right)^{0.44} R_X^{-0.167} \tag{23}$$

$$\theta = 0.022 \times \left(1.0 + 0.10M^2\right)^{-0.70} R_X^{-0.167} \tag{24}$$

These equations are valid for $\gamma = 1.4$.

Since the static pressure is considered constant through the boundary layer, the pressure that is computed a distance δ^* from the surface is transferred to the surface. This pressure forms the basis of the iteration loop so that the calculations can be cycled in the standard manner for conditions on the surface.

Reynolds number effect. - It is well known that Reynolds number affects the measured flow coefficient of choked nozzles. If the loss in mass flow is considered as a blockage effect caused by boundary layer displacement, the discharge and velocity coefficients associated with only blockage can be approximated by the following expressions:

$$C_D = 1.0 - \frac{4}{D_p} \frac{\delta^*}{\cos\ \alpha} \tag{25}$$

$$C_v = 1.0 - \frac{4}{D_p} \theta \tag{26}$$

For a 1/7-power velocity distribution law the boundary layer displacement thickness varies as $x^{4/5}$ according to the expression obtained from reference 14. When the primary nozzle exit diameter is identified with the characteristic length, equations (25) and (26) become

$$C_D = 1.0 - K_1 R_D^{-0.2} \tag{27}$$

$$C_v = 1.0 - K_2 R_D^{-0.2} \tag{28}$$

where

$$K_1 = \frac{0.185}{\cos\ \alpha} \tag{29}$$

$$K_2 = 0.144 \tag{30}$$

for a 1/7-power velocity profile. The correlations expressed by equations (27) and (28) are compared with unpublished data obtained on a ASME nozzle in figure 3. The nozzle thrust coefficient is presented as the ratio of actual thrust to ideal thrust of a choked nozzle based on the measured weight flow. In general, agreement appears to be very good for the Reynolds range 1×10^6 to 1×10^7. Equations (27) and (28) can be combined with equations (15) and (16) to form a more general expression for the flow and velocity coefficients.

$$C_D = \left(1.0 - K_1 R_D^{-0.2}\right) \int (\cos\ \vartheta y\ dy - \sin\ \vartheta y\ dx) \tag{31}$$

$$C_v = \left(1.0 - K_2 R_D^{-0.2}\right) \frac{1}{C_{D_i}} \int \cos\ \vartheta (\cos\ \vartheta y\ dy - \sin\ \vartheta y\ dx) \tag{32}$$

where C_{D_i} is the inviscid flow coefficient represented by equation (15) and the integration is performed along the sonic line. Equations (31) and (32) were used to compute the flow coefficients for the choked conical nozzles with primary nozzle lip angles α of 8^{o} and 27^{o} that were presented in reference 11. Agreement between calculations and measurements for both cases was very good. Thus, REJECT computes the performance of the primary nozzle acting in conjunction with the ejector.

Secondary Flow Field Solutions

The purpose of program REJECT is to establish, for a given geometry, the performance characteristics of the ejector system when the secondary flow becomes independent of ambient conditions. The flow regimes occurring under this condition can be categorized on the basis of the predominant flow mechanisms.

When the amount of secondary flow supplied to the ejector nozzle is small, the primary flow plumes out and impinges on the shroud wall, as shown in figure 4(a). This causes an oblique shock to form which effectively "seals off" the secondary flow from ambient conditions. The secondary flow is "dragged" through the oblique-shock pressure rise by its mixing action with the higher velocity primary jet flow. Equilibrium conditions are thus established in the "low" secondary mass flow regime when the amount of secondary flow supplied to the ejector is equal to the flow which is dragged past the recompression zone associated with the oblique shock. For the ejector operating in the "high" secondary mass flow regime shown in figure 4(b), the interaction between the two streams is such that the secondary flow accelerates to critical conditions somewhere downstream of the primary nozzle. The viscous interaction between the two streams occurs along their interface. As a result, energy (shear work) is transferred from the primary to the secondary stream and the pumping characteristics are modified by the displacement effects of the mixing region.

Interaction between inviscid flow fields. - The primary stream internal flow field is analyzed by the method of characteristics for irrotational axisymmetric flow. The general conditions that must be satisfied along the jet boundary between the primary and secondary streams are (1) the local static pressure must be equal for both streams at their interface and (2) continuity between the two streams must be preserved. For flows in which the mixing is neglected, the later condition requires that

$$\frac{W_s}{W_p} = \frac{f(\gamma_s)}{f(\gamma_p)} \frac{A_s^*}{A_p^*} \frac{P_s}{P_p} \tag{33}$$

where

$$f(\gamma) = \sqrt{\gamma}\left(\frac{\gamma + 1}{2}\right)^{-(\gamma+1)/2(\gamma-1)} \tag{34}$$

and

$$\frac{W_s}{W_p} = \frac{w_s}{w_p}\sqrt{\frac{T_s}{T_p}} \tag{35}$$

Thus, for a given secondary total-pressure ratio and corrected weight flow ratio, the parameter A_s^*/A_p^*, defined by equation (33), relates the conditions on both sides of the jet boundary. Consequently, the Mach number in the secondary passage can be determined from the expression

$$\frac{A_s}{A_s^*} = \frac{A_s}{A_p} \times \frac{A_p/A_p^*}{A_s^*/A_p^*} \tag{36}$$

where A_s is the local secondary flow area, A_p is the primary nozzle exit area, A_p/A_p^* is a function of the primary nozzle exit Mach number, and A_s^*/A_p^* is related to the ejector operating conditions through equation (33). The local secondary flow conditions can thus be established from equation (36), and this will determine the jet boundary.

Relations within mixing region. - The quasi-constant-pressure two-stream mixing processes presented in references 5 and 7 was used as the mixing model in program REJECT. In most cases of ejector operation, the characteristic Reynolds number is sufficiently large so that the jet mixing process occurring within the flow field is likely to be turbulent. The assumption of quasi-constant-pressure two-stream mixing means that the velocity profile, at the flow station under consideration, is considered to be developed from the local two streams through a constant-pressure turbulent jet mixing process. The velocity profile within such a region will be given by

$$\varphi = \frac{u}{u_p} = \frac{1}{2}(1 + \varphi_s) + \frac{1}{2}(1 - \varphi_s)\mathrm{erf}(\eta) \tag{37}$$

where

$$\varphi_s = \frac{u_s}{u_p} \tag{38}$$

$$\text{erf}(\eta) = \frac{2}{\sqrt{\pi}} \int_0^{\eta} e^{-\beta^2}\, d\beta \tag{39}$$

$$\eta = \frac{\sigma y}{x} \tag{40}$$

and σ is the similarity parameter for the homogeneous coordinate system within the jet mixing region. The jet boundary separating the two streams is located at η_j within the mixing region so that

$$I_1(\eta_j) = \frac{I_1(\eta_p) - I_2(\eta_p)}{1 - \varphi_s} \tag{41}$$

where $I_1(\eta)$ and $I_2(\eta)$ represent the integrals

$$I_1(\eta) = \frac{\left(1 - C_p^2\right)\varphi_s\eta_s}{\frac{T_s}{T_p} - C_p^2\varphi_s^2} + \int_{\eta_s}^{\eta} \frac{\left(1 - C_p^2\right)\varphi}{\lambda - C_p^2\varphi^2}\, d\eta \tag{42}$$

$$I_2(\eta) = \frac{\left(1 - C_p^2\right)\varphi_s^2\eta_s}{\frac{T_s}{T_p} - C_p^2\varphi_s^2} + \int_{\eta_s}^{\eta} \frac{\left(1 - C_p^2\right)\varphi^2}{\lambda - C_p^2\varphi^2}\, d\eta \tag{43}$$

where the Crocco number C_p is defined by the equation

$$C_p^2 = \frac{2}{(\gamma - 1) + M_p^2} \tag{44}$$

and

$$\lambda = \frac{1}{1 - \varphi_s}\left[\left(\frac{T_s}{T_p} - \varphi_s\right) + \left(1 - \frac{T_s}{T_p}\right)\varphi\right] \tag{45}$$

where λ is the stagnation temperature ratio throughout the mixing region for fluids with a Prandtl number of 1. The velocity and stagnation temperature profiles described are defined within an intrinsic coordinate system which is located by a shift of y_n with respect to the reference system and which is given by

$$\eta_n = \frac{\sigma y_n}{x_n} = \eta_p - \frac{1}{1 - \varphi_s}\left[I_2(\eta_p) - \varphi_s I_1(\eta_p)\right] \tag{46}$$

The effects of the entrainment of the secondary flow by the primary stream are interpreted in the boundary space as equivalent to a displacement of the secondary stream boundary δ^*. Thus, it can be shown that

$$-\sigma \frac{\delta^*}{x} = \sigma \frac{v_s}{u_s} = \frac{\left(\frac{T_s}{T_p}\right) - C_p^2 \varphi_s^2}{\left(1 - C_p^2\right)\varphi_s} I_1(\eta_j) - \eta_n \tag{47}$$

The numerical value of the similarity parameter σ has been well established to be 12 for the mixing of an incompressible fluid and later extended to $12 + 2.758\ M_p$ for the compressible flow regime. To extend the concept of a similarity parameter to the two-stream mixing problem, a rationale was proposed in reference 7 which resulted in the following expressions:

$$\sigma_{II} = \left(\frac{\sigma_{II}}{\sigma_I}\right)\sigma_I \tag{48}$$

where

$$\frac{\sigma_{II}}{\sigma_I} = \frac{1 + \varphi_s}{1 - \varphi_s} \tag{49}$$

and the equivalent one-stream Crocco number C_{p_I} is given by the relation

$$C_{p_I}^2 = \frac{C_{p_{II}}^2 (1 - \varphi_s)^2}{C_{p_{II}}^2 (1 - \varphi_s)^2 + \left(1 - C_{p_{II}}^2\right)} \tag{50}$$

By using equation (47), the equivalent one-stream similarity parameter σ_I can be defined in terms of the Crocco number C_{p_I} by the relation

$$\sigma_I = (12 + 2.758)\left[\frac{C_{p_I}}{\sqrt{\left(1 - C_{p_I}^2\right)^{(\gamma-1)/2}}}\right] \tag{51}$$

Mixing process in high secondary flow regime. - The conservation of mass within the ejector nozzle flow field (fig. 4) would require that at each streamwise station, the following expression would be valid:

$$\frac{W_s}{W_p} = \left(\frac{W_{s,i}}{W_p} + \frac{\Delta W_s}{W_p}\right) \tag{52}$$

where W_s/W_p is the secondary corrected weight flow ratio supplied to the ejector, $W_{s,i}/W_p$ is the secondary weight flow ratio which is unaffected by the mixing process, and $\Delta W_s/W_p$ represents the amount of flow entrained by the mixing process to the point of consideration. The ratio $\Delta W_s/W_p$ can be expressed by the following relation:

$$\frac{\Delta W_s}{W_p} = \frac{2}{\sigma_{II}}\frac{\left(\frac{A}{A^*}\right)_p}{\left(\frac{A}{A^*}\right)_n} \times x_n y_n \frac{\left(1 - C_p^2\right)\varphi_s}{1 - C_p^2 \varphi_s^2}\left(-\sigma_{II} \times \frac{\delta^*}{x}\right) \tag{53}$$

Since the secondary corrected weight flow supplied to the ejector remains constant, the term $W_{s,i}/W_p$ can be computed from equation (52). Since the secondary total-pressure ratio P_s/P_p remains constant, the local secondary critical area ratio is thus affected by jet mixing through the expression

$$\frac{A^*_{s,i}}{A_p} = \frac{f(\gamma_p)}{f(\gamma_s)} \times \frac{P_p}{P_s}\left(\frac{W_s}{W_p} - \frac{\Delta W_s}{W_p}\right) \tag{54}$$

The parameter $A^*_{s,i}/A^*_p$ defined by equation (54) is evaluated at each point along the jet boundary, so that conservation of mass is always preserved.

The local secondary flow conditions are determined from the expression

$$\frac{A_{s,i}}{A^*_{s,i}} = \frac{A_{s,i}}{A_p} \frac{A_p}{A^*_p} \left[\frac{f(\gamma_p)}{f(\gamma_s)} \frac{P_p}{P_s} \left(\frac{W_s}{W_p} - \frac{\Delta W_s}{W_p}\right)\right]^{-1} \tag{55}$$

which is used in the iterative procedure to define the local flow conditions along the jet boundary. This procedure takes into account the effects of mixing on the local secondary flow Mach number. The local secondary flow area is obtained from the expression

$$A_{s,i} = A_s - \Delta A_s \tag{56}$$

where A_s is the secondary flow area defined by the local jet boundary and ΔA_s is the area associated with the mixing region. This term can be obtained from the relation

$$\Delta A_s = \frac{2}{\sigma_{II}} \times x_n y_n \left(-\sigma_{II} \times \frac{\delta^*}{x}\right) \tag{57}$$

Equation (57) represents the change in the secondary flow area caused by the effects of entrainment of the secondary stream by the primary stream (fig. 4) as interpreted in the boundary layer sense by the displacement thickness $-\delta^*$.

The isolation condition for the ejector operating in the high secondary flow regime requires that the critical area ratio defined by equation (55) be greater than, but in the neighborhood of, 1.0. This is accomplished in REJECT through an iterative process by fixing the corrected secondary weight flow ratio W_s/W_p and varying the secondary total pressure ratio P_s/P_p until this condition is satisfied.

Impingement solution. - As previously described, equilibrium conditions are established in the low secondary weight flow regime when the amount of flow that is dragged past the recompression shock (fig. 4(a)) is equal to the amount of secondary flow supplied to the ejector system. The amount of secondary flow for this case is given by the expression

$$\frac{W_s}{W_p} = \frac{2}{\sigma_{II}} x_w y_w \frac{\left(\frac{A}{A^*}\right)_p}{\left(\frac{A}{A^*}\right)_w} \left[I_1(\eta_j) - I_1(\eta_d)\right] \tag{58}$$

where x_w and y_w give the location of the primary jet impingement on the shroud wall and $(A/A^*)_w$ is the area ratio associated with the primary flow upstream of the recompression shock. Application of the escape criterion

$$\left(\frac{p_2}{p_1}\right)_w = \left(\frac{P}{p}\right)_d = \left(1 - \frac{\gamma - 1}{\gamma + 1} q_d^2\right)^{-\gamma/(\gamma-1)} \tag{59}$$

would allow an identification of a discriminating streamline η_d within the jet mixing region which distinguishes that part of the flow which is "turned" back and that part which has sufficient energy to traverse the static-pressure rise $(p_2/p_1)_w$ associated with the impingement shock (fig. 4(a)).

Shroud boundary layer. - The boundary layer along the shroud wall is treated in a manner similar to that used for the boundary layer along the centerbody surface, which was previously discussed. The marching procedure is preserved, and equations (19) to (29) provide the working formulas for the computation of the shroud boundary layer.

Performance parameters. - The two most important parameters which describe the thrust performance of ejector systems are the nozzle efficiency

$$C_T = \frac{\dfrac{F}{P_pA_p} - \dfrac{p_e}{P_p}\left(\dfrac{A_e}{A_p}\right)}{\dfrac{F_{ip}}{P_pA_p} + \dfrac{F_{is}}{P_pA_p}} \tag{60}$$

and the gross thrust coefficient

$$C_{F_g} = \frac{\dfrac{F}{P_pA_p} - \dfrac{p_e}{P_p}\left(\dfrac{A_e}{A_p}\right)}{\dfrac{F_{ip}}{P_pA_p}} \tag{61}$$

where $F_{ip}/(P_pA_p)$ and $F_{is}/(P_pA_p)$ are the ideal thrust of the primary and secondary streams based on the measured weight flows and the nozzle pressure ratio P_p/p_e. The ideal thrust of the primary and secondary streams can be computed from the relations

$$\frac{F_{ip}}{P_pA_p} = C_D \times f_{id}\left(\gamma_p, \frac{p_e}{P_p}\right) \tag{62}$$

$$\frac{F_{is}}{P_p A_p} = \frac{P_s}{P_p} \times \frac{A_s^*}{A_p^*} \times \frac{A_p^*}{A_p} f_{id}\left(\gamma_s, \frac{p_e}{P_s}\right) \tag{63}$$

where

$$f_{id}\left(\gamma, \frac{p}{P}\right) = \gamma \left(\frac{2}{\gamma + 1}\right)^{\gamma/(\gamma-1)} \left(\frac{\gamma + 1}{\gamma - 1}\right)^{1/2} \left[1 - \left(\frac{p}{P}\right)^{(\gamma-1)/2}\right]^{1/2} \tag{64}$$

The total stream or vacuum thrust of the ejector system is defined by the following expression:

$$\frac{F}{P_p A_p} = \frac{F_p}{P_p A_p} + \frac{F_s}{P_p A_p} + \frac{F_{sh}}{P_p A_p} + \frac{F_b}{P_p A_p} - \frac{F_f}{P_p A_p} \tag{65}$$

where

$$\frac{F_p}{P_p A_p} = \left(\frac{p}{P}\right)_p \left(1.0 + \gamma_p C_D C_v M_p^2\right) \tag{66}$$

$$\frac{F_s}{P_p A_p} = \frac{p_s}{P_p} \left(1.0 + \gamma_s M_s^2\right) \frac{A_s}{A_p} \tag{67}$$

$$\frac{F_{sh}}{P_p A_p} = \frac{1}{P_p A_p} \int_{A_{sh}} p \, dA \tag{68}$$

$$\frac{F_b}{P_p A_p} = \frac{1}{P_p A_p} \int_{A_b} p \, dA \tag{69}$$

$$\frac{F_f}{P_p A_p} = \frac{1}{P_p A_p} \int_{S_{sh}} \gamma_s p_s M_s^2 C_f \, d\tau + \frac{1}{P_p A_p} \int_{S_b} \gamma_p p_p M_p^2 C_f \, d\tau \tag{70}$$

OUTLINE OF GENERAL SOLUTION

The actual computation of the pumping characteristics, whether in the low or high secondary flow regime, requires an iterative procedure which involves the calculation of the entire flow field to some point where the isolation condition can be verified. This overall iteration is accomplished by holding the secondary corrected weight flow ratio W_s/W_p constant and choosing different values of secondary total-pressure ratio P_s/P_p. The set of equations discussed were programmed in a downstream forward-marching procedure starting from the sonic line solution, which depends on both W_s/W_p and P_s/P_p. Once initiated, the procedure is advanced to a new station, where the previous station's solution is used as input for the new solution. At this point, the secondary flow field is examined and a decision is made whether to continue in the downstream direction or to seek a new value of total-pressure ratio P_s/P_p and repeat the cycle. When a solution has been found, the entire ejector flow field is computed, along with the standard ejector performance parameters. Because of the many iterations that can occur in arriving at the pumping characteristics, program REJECT has been structured to avoid excessive iterations within each flow field calculation, although such computations as the sonic line, two-stream mixing interaction and boundary layer effects are repeated for each iteration. One complete flow field computation requires about 40 seconds of central processing unit time on the CDC 6600 computer.

CONCLUDING REMARKS

A theoretical analysis for the solution of flow within ejector nozzles of arbitrary geometry is presented by using standard numerical techniques. A large number of cases have been calculated with this analysis and compared with appropriate experimental data (refs. 10 and 11). The procedure presented in this report provides an accurate and economical method of designing a wide variety of ejector nozzles over the range of flow conditions of interest.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, November 28, 1973,
501-24.

APPENDIX A

SYMBOLS

A	area
A_p	primary nozzle exit area
A_s	local secondary flow area
a	speed of sound
C	Crocco number
C_D	discharge coefficient
C_f	skin friction coefficient
C_{F_g}	gross thrust coefficient
C_T	nozzle efficiency
C_v	velocity coefficient
D_p	primary nozzle exit diameter
F	stream thrust
$F(\omega - i\vartheta)$	complex stream function, eq. (9)
F_f	integrated skin friction force
F_{ip}	ideal primary thrust based on actual weight flow
F_{is}	ideal secondary thrust based on actual weight flow
F_p	primary stream thrust
F_s	secondary stream thrust
$f(\gamma)$	parameter defined by eq. (34)
$f_{id}(\gamma, p/P)$	parameter defined by eq. (64)
G	parameter defined by eq. (20)
$I_1(\eta)$	parameter defined by eq. (42)
$I_2(\eta)$	parameter defined by eq. (43)
j	dimensionality
K_1	constant defined by eq. (29)
K_2	constant defined by eq. (30)

M Mach number

P total pressure

p static pressure

q velocity ratioed to critical speed

R_D Reynolds number based on primary nozzle diameter

R_X Reynolds number based on equivalent length

S entropy

S_b surface area of body

S_{sh} surface area of shroud

T total temperature

u axial component of velocity ratioed to critical speed

v radial component of velocity ratioed to critical speed

W corrected weight flow, $w\sqrt{T}$

w weight flow

X equivalent length, eq. (19)

x axial distance ratioed to primary nozzle radius

y radial distance ratioed to primary nozzle radius

z complex coordinate, $x + iy$

α primary nozzle lip angle

β constant, 1.20 or 1.25

γ ratio of specific heats

δ^* displacement thickness

η parameter defined by eq. (40)

θ momentum thickness

ϑ flow angle

$\kappa(M)$ parameter defined by eq. (7)

λ temperature ratio, eq. (45)

μ Mach angle

ρ density

σ similarity parameter

φ velocity ratio

ψ stream function

ω transformed velocity, eq. (8)

Subscripts:

a approach conditions

b body conditions

d discriminating streamline

e exit conditions

i inviscid

id ideal conditions

j jet boundary

m match conditions

n local conditions

p primary conditions

s secondary flow conditions

sh shroud conditions

w wall conditions

x conditions upstream of shock

z conditions downstream of shock

0 reference conditions

I jet mixing of one stream with a quiescent stream

II two-stream mixing

Superscript:

* critical conditions

APPENDIX B

DESCRIPTION OF INPUT

This section describes the loading of input data cards for running the computer program. Care should be taken in loading because the input changes depending on the options chosen. Multiple cases can be run simply by stacking the cases in order. See figures 5 and 6 for further explanation of geometric input quantities.

Card 1 - Title Card

Name	Column	Format	Comment
TITLE	1-72	18A4	Any alphanumeric characteristics

Card 2 - Data Variables

Name	Column	Format	Comment
WTFL	1-12	6E12.0	Secondary corrected weight flow ratio
HSHP	13-24		Initial estimate of secondary total-pressure ratio
TOS	25-36		Secondary stream total temperature, K (^{o}R)
GAMS	37-48		Ratio of specific heats for secondary stream
TOP	49-60		Primary stream total temperature, K (^{o}R)
GAMP	61-72		Ratio of specific heats for primary stream

Card 3 - Data Variables

Name	Column	Format	Comment
AMR	1-12	6E12.0	Match Mach number for starting the primary flow field calculation: set AMR = 1.003 for plug nozzle configurations; otherwise, set AMR = 1.001

Name	Column	Format	Comment
ANGR	13-24		Primary nozzle conical lip angle, deg: set ANGR = 0 for plane sonic line solution; otherwise, ANGR < 0
RATIO	25-36		Primary nozzle radius ratio, fig. 5
XPRIM	37-48		Location of primary nozzle exit relative to coordinate system of shroud contour points, cm (in.)
DPRIM	49-60		Diameter of primary nozzle exit, cm (in.)
DSHD	61-72		Set NSHD = 1 to calculate performance of cylindrical shroud ejector with diameter DSHD, cm (in.)

Card 4 - Data Variables

Name	Column	Format	Comment
DBDY	1-12	6E12.0	Set CONA > 0 to calculate conical plug nozzle geometry with half-cone angle of CONA and diameter of DBDY in plane of primary nozzle exit, cm (in.)
CONA	13-24		Half-angle of conical plug nozzle, deg
END	25-36		Location of nozzle exit relative to coordinate system of shroud contour points, cm (in.)
REYPRM	37-48		Reynolds number based on primary nozzle exit diameter
DELSHD(1)	49-60		Initial boundary layer displacement thickness ratioed to primary nozzle exit radius, δ^*/R_p
FDIM	61-72		Set FDIM = 0.0 to calculate two-dimensional flow; set FDIM = 1.0 to calculate axisymmetric flow

Card 5 - Diagram Constants and Options

Name	Column	Format	Comment
K1	1-12	6E12.0	Constant controlling initial expansion fan: set K1 = 0.0005
K2	13-24		Constant controlling initial expansion fan: set K2 = 0.1000
K3	25-36		Constant controlling initial expansion fan: set K3 = 1.000
K4	37-48		Constant controlling insertion of additional field points: set K4 = 0.0500
SOLVE	49-60		Set SOLVE = 0.0 to calculate nonmixing solution; set SOLVE = 1.0 to calculate mixing solution recommended for WTFL $>$ 0.04; set SOLVE = 2.0 to calculate impingement solution for 0 $\leq$ WTFL $<$ 0.04
PRINT	61-72		Set PRINT = 0 for no printout of primary flow field; set PRINT = 1.0 for printout of primary flow field for final solution; set PRINT = 2.0 for printout primary flow field for every iteration

Card 6 - Program Constants and Options

Name	Column	Format	Comment
NDATA	1-6	8I6	Number of field points along sonic line: for ANGR $<$ 0 set NDATA = 21; for ANGR = 0 set NDATA = 8
NSHD	7-12		Number of shroud contour points read as input data
NBDY	13-18		Number of centerbody contour points read as input data
NITER	19-24		Number of iterations previously completed for restart option

Name	Column	Format	Comment
IPLOT[1]	25-30		Parameter which controls plotting routines
IPNCH	31-36		Set IPNCH = 0
IPRNT	37-42		Set IPRNT = 0
ICOMP	43-48		Set ICOMP = 0

Card 7 - NITER > 0

Name	Column	Format	Comment
PTS(I)	1-12	6E12.0	Secondary total-pressure ratio (HSHP) at the I^{th} iteration, where I = 1, NITER
AREA(I)	13-24		Minimum computed secondary flow area ratio A_s/A_s^* at the I^{th} iteration, for SOLVE = 0.0, 1.0, where I = 1, NITER
WLEAK(I)	25-36		Computed leakage secondary weight flow ratio at the I^{th} iteration for SOLVE = 2.0, where I = 1, NITER

A restart option is provided in REJECT to iterate for the solution between the values read on card 7. The values of PTS(I), AREA(I), and WLEAK(I) are printed out after each iteration for the pumping characteristics.

Card 8+ - Shroud Geometry (for NSHD > 1)

Name	Column	Format	Comment
XSHD(I)		6E12.0	Axial location of shroud coordinate point, cm (in.)
YSHD(I)			Radial location of shroud coordinate point, where I = 1, NSHD, cm (in).

[1]The plotting routines have been deleted from the computer FORTRAN listings published in this report.

Card 9+ - Plug Geometry (for NBDY > 0)

Name	Column	Format	Comment
XBDY(I)		6E12.0	Axial location of plug coordinate point, cm (in.)
YBDY(I)			Radial location of plug coordinate point, where I = 1, NSHD, cm (in.)

APPENDIX C

DESCRIPTION OF OUTPUT

A sample output listing is presented in appendix D for the solution of the flow in a convergent-divergent conical flap ejector nozzle operating at a secondary weight flow ratio of 0.10. A detailed comparison between the calculated and measured performance of this ejector is presented in reference 11. The output was obtained by using the print option which gives the minimum amount of output (PRINT = 0.0) because a complete output would be both lengthy and repetitious.

The output presented on the title page summarizes the important input variables and is self-explanatory. The second and third pages of the output listing present a table of the dimensionless shroud coordinate points relative to the primary nozzle exit station, along with the surface tangent and the surface angle. These variables are labeled as follows: XSHD(I), YSHD(I), DYSDX(I), ANGLE(I). Since the calculations are always performed in a coordinate system relative to the primary nozzle exit (i.e., XSHD(I) = 0.0 always represents this station), the spacing ratio can easily be changed by varying XPRIM in the input data. The fourth page summarizes the condition calculated along the jet boundary when the ejector is operating at the listed secondary corrected weight flow ratio (WTFL) and secondary total pressure ratio (PTS/PTP). The parameters presented on this page of the output are defined as follows:

XSLP	axial position
YSLP	radial location of jet boundary
AMP	primary stream Mach number
THETA	flow angle of jet boundary
P/PTP	ratio of static to primary total pressure
AMS	secondary stream Mach number
P/PTS	ratio of static to secondary total pressure
AS/AS*	ratio of secondary flow area to secondary critical area

In using the restart option in REJECT, the values of PTS(I), AREA(I), and NLEAK(I) which are read in as input data correspond to the total-pressure ratio PTS/PTP, minimum secondary area ratio AS/AS*, and leakage weight flow ratio WLEAK(I) which are printed on this page of the output. The fifth page of the output listing presents the results of the boundary layer calculation performed in REJECT when the ejector is operating at the listed secondary weight flow ratio (WTFL), secondary total-pressure ratio (PTS/PTP), primary nozzle Reynolds number (REYPRM), and corresponding secondary Reynolds number (REYSEC). The boundary layer calculations along the shroud wall are based on the secondary flow conditions.

The displacement thickness (DELSHD) and momentum thickness (THETAS) are normalized with respect to the primary nozzle exit radius, while the skin friction coefficient (CFSHD) is based on secondary dynamic pressure.

The sixth page of the output listing presents the sonic line solution computed in REJECT, while the seventh page summarizes the important performance parameters. The eighth page of the output listing presents the relation between the nozzle pressure ratio (PTP/P0) and the following nozzle parameters:

FGROSS	gross stream thrust, $(F - P_e A_e)/(P_p A_p)$
FIP	ideal thrust of primary nozzle
FIS	ideal thrust of secondary stream
CVP	gross thrust coefficient
CV	nozzle efficiency

APPENDIX D

SAMPLE OUTPUT LISTING

```
*****BRITISH EJECTOR NOZZLE, DSHD/DPRIM = 1.225, LSHD/DPRIM = 0.220*****

     PRIMARY NOZZLE EXIT DIAMETER, DPRIM =  2.00000

     PRIMARY NOZZLE EXIT MACH NO., AMR =  1.00100

     PRIMARY NOZZLE LIP ANGLE, ANGR = -16.00000

     PRIMARY NOZZLE RADIUS RATIO, YRATIO = 1.40000

     LOCATION OF PRIMARY NOZZLE, XPRIM =     .95860

     EJECTOR LENGTH MEASURED FROM PRIMARY NOZZLE, END =  3.19592

     PRIMARY NOZZLE REYNOLDS NUMBER, REYPRM = 4.000E+06

     TOTAL TEMPERATURE OF SECONDARY FLOW, TOS =  560.000

     RATIO OF SPECIFIC HEATS FOR SECONDARY FLOW, GAMS = 1.40000

     TOTAL TEMPERATURE OF PRIMARY FLOW, TOP =  560.000

     RATIO OF SPECIFIC HEATS FOR PRIMARY FLOW, GAMP = 1.40000

     FLOW IN EJECTOR IS AXISYMMETRIC, FDIM = 1.0

     NUMBER OF POINTS SPECIFING SHROUD CONTOUR, NSHD = 86

     NUMBER OF POINTS SPECIFING PLUG CONTOUR, NBDY =  0

     NUMBER OF POINTS SPECIFING ENTERENCE CONDITIONS, NDATA = 21
```

SHROUD CONTOUR SPECIFICATIONS

I = 1	XSHD(I) = -3.95860	YSHD(I) = 1.65800	DYSDX(I) = .00009	ANGLE(I) = .0053
I = 2	XSHD(I) = -3.19980	YSHD(I) = 1.65800	DYSDX(I) = -.00012	ANGLE(I) = -.0067
I = 3	XSHD(I) = -2.44100	YSHD(I) = 1.65800	DYSDX(I) = .00037	ANGLE(I) = .0214
I = 4	XSHD(I) = -1.68223	YSHD(I) = 1.65800	DYSDX(I) = -.00138	ANGLE(I) = -.0789
I = 5	XSHD(I) = -.92344	YSHD(I) = 1.65800	DYSDX(I) = .00513	ANGLE(I) = .2941
I = 6	XSHD(I) = -.16465	YSHD(I) = 1.65800	DYSDX(I) = -.01916	ANGLE(I) = -1.0975
I = 7	XSHD(I) = -.15131	YSHD(I) = 1.65750	DYSDX(I) = -.07355	ANGLE(I) = -4.2063
I = 8	XSHD(I) = -.13806	YSHD(I) = 1.65590	DYSDX(I) = -.16173	ANGLE(I) = -9.1868
I = 9	XSHD(I) = -.12497	YSHD(I) = 1.65330	DYSDX(I) = -.23810	ANGLE(I) = -13.3927
I = 10	XSHD(I) = -.11212	YSHD(I) = 1.64970	DYSDX(I) = -.32299	ANGLE(I) = -17.8999
I = 11	XSHD(I) = -.09959	YSHD(I) = 1.64510	DYSDX(I) = -.41285	ANGLE(I) = -22.4334
I = 12	XSHD(I) = -.08747	YSHD(I) = 1.63950	DYSDX(I) = -.51464	ANGLE(I) = -27.2323
I = 13	XSHD(I) = -.07583	YSHD(I) = 1.63290	DYSDX(I) = -.61796	ANGLE(I) = -31.7146
I = 14	XSHD(I) = -.06473	YSHD(I) = 1.62550	DYSDX(I) = -.71683	ANGLE(I) = -35.6340
I = 15	XSHD(I) = -.05424	YSHD(I) = 1.61730	DYSDX(I) = -.86265	ANGLE(I) = -40.7828
I = 16	XSHD(I) = -.04444	YSHD(I) = 1.60820	DYSDX(I) = -.96975	ANGLE(I) = -44.1201
I = 17	XSHD(I) = -.01344	YSHD(I) = 1.57720	DYSDX(I) = -1.00851	ANGLE(I) = -45.2428
I = 18	XSHD(I) = .01755	YSHD(I) = 1.54620	DYSDX(I) = -.99717	ANGLE(I) = -44.9187
I = 19	XSHD(I) = .04860	YSHD(I) = 1.51520	DYSDX(I) = -.99896	ANGLE(I) = -44.9701
I = 20	XSHD(I) = .07960	YSHD(I) = 1.48420	DYSDX(I) = -1.00218	ANGLE(I) = -45.0623
I = 21	XSHD(I) = .11050	YSHD(I) = 1.45320	DYSDX(I) = -1.00206	ANGLE(I) = -45.0590
I = 22	XSHD(I) = .14150	YSHD(I) = 1.42220	DYSDX(I) = -.99930	ANGLE(I) = -44.9800
I = 23	XSHD(I) = .17250	YSHD(I) = 1.39120	DYSDX(I) = -1.00073	ANGLE(I) = -45.0210
I = 24	XSHD(I) = .20350	YSHD(I) = 1.36020	DYSDX(I) = -.99777	ANGLE(I) = -44.9359
I = 25	XSHD(I) = .23450	YSHD(I) = 1.32920	DYSDX(I) = -1.00820	ANGLE(I) = -45.2340
I = 26	XSHD(I) = .26550	YSHD(I) = 1.29820	DYSDX(I) = -.96942	ANGLE(I) = -44.1105
I = 27	XSHD(I) = .27560	YSHD(I) = 1.28870	DYSDX(I) = -.90019	ANGLE(I) = -41.9931
I = 28	XSHD(I) = .28620	YSHD(I) = 1.27970	DYSDX(I) = -.80136	ANGLE(I) = -38.7072
I = 29	XSHD(I) = .29720	YSHD(I) = 1.27140	DYSDX(I) = -.70686	ANGLE(I) = -35.2550
I = 30	XSHD(I) = .30880	YSHD(I) = 1.26370	DYSDX(I) = -.62887	ANGLE(I) = -32.1645
I = 31	XSHD(I) = .32070	YSHD(I) = 1.25660	DYSDX(I) = -.55966	ANGLE(I) = -29.2338
I = 32	XSHD(I) = .33300	YSHD(I) = 1.25020	DYSDX(I) = -.48481	ANGLE(I) = -25.8643
I = 33	XSHD(I) = .34560	YSHD(I) = 1.24450	DYSDX(I) = -.42002	ANGLE(I) = -22.7832
I = 34	XSHD(I) = .35850	YSHD(I) = 1.23950	DYSDX(I) = -.35583	ANGLE(I) = -19.5872
I = 35	XSHD(I) = .37170	YSHD(I) = 1.23520	DYSDX(I) = -.29744	ANGLE(I) = -16.5646
I = 36	XSHD(I) = .38510	YSHD(I) = 1.23160	DYSDX(I) = -.23805	ANGLE(I) = -13.3902
I = 37	XSHD(I) = .39870	YSHD(I) = 1.22880	DYSDX(I) = -.17445	ANGLE(I) = -9.8955
I = 38	XSHD(I) = .41240	YSHD(I) = 1.22680	DYSDX(I) = -.11998	ANGLE(I) = -6.8419
I = 39	XSHD(I) = .42620	YSHD(I) = 1.22550	DYSDX(I) = -.06635	ANGLE(I) = -3.7961
I = 40	XSHD(I) = .44000	YSHD(I) = 1.22500	DYSDX(I) = -.00591	ANGLE(I) = -.3387
I = 41	XSHD(I) = .45390	YSHD(I) = 1.22530	DYSDX(I) = .04583	ANGLE(I) = 2.6240
I = 42	XSHD(I) = .46770	YSHD(I) = 1.22630	DYSDX(I) = .10488	ANGLE(I) = 5.9874
I = 43	XSHD(I) = .48140	YSHD(I) = 1.22810	DYSDX(I) = .14647	ANGLE(I) = 8.3331
I = 44	XSHD(I) = .54930	YSHD(I) = 1.23880	DYSDX(I) = .16163	ANGLE(I) = 9.1813
I = 45	XSHD(I) = .61710	YSHD(I) = 1.24960	DYSDX(I) = .15763	ANGLE(I) = 8.9580

SHROUD CONTOUR SPECIFICATIONS

I = 46	XSHD(I) = .68500	YSHD(I) = 1.26030	DYSDX(I) = .15847	ANGLE(I) = 9.0048
I = 47	XSHD(I) = .75290	YSHD(I) = 1.27110	DYSDX(I) = .15841	ANGLE(I) = 9.0017
I = 48	XSHD(I) = .82070	YSHD(I) = 1.28180	DYSDX(I) = .15849	ANGLE(I) = 9.0061
I = 49	XSHD(I) = .88860	YSHD(I) = 1.29260	DYSDX(I) = .15824	ANGLE(I) = 8.9917
I = 50	XSHD(I) = .95650	YSHD(I) = 1.30330	DYSDX(I) = .15849	ANGLE(I) = 9.0060
I = 51	XSHD(I) = 1.02430	YSHD(I) = 1.31410	DYSDX(I) = .15843	ANGLE(I) = 9.0025
I = 52	XSHD(I) = 1.09220	YSHD(I) = 1.32480	DYSDX(I) = .15843	ANGLE(I) = 9.0024
I = 53	XSHD(I) = 1.16000	YSHD(I) = 1.33560	DYSDX(I) = .15849	ANGLE(I) = 9.0061
I = 54	XSHD(I) = 1.22790	YSHD(I) = 1.34630	DYSDX(I) = .15823	ANGLE(I) = 8.9913
I = 55	XSHD(I) = 1.29580	YSHD(I) = 1.35710	DYSDX(I) = .15851	ANGLE(I) = 9.0073
I = 56	XSHD(I) = 1.36360	YSHD(I) = 1.36780	DYSDX(I) = .15833	ANGLE(I) = 8.9972
I = 57	XSHD(I) = 1.43150	YSHD(I) = 1.37860	DYSDX(I) = .15877	ANGLE(I) = 9.0214
I = 58	XSHD(I) = 1.49930	YSHD(I) = 1.38930	DYSDX(I) = .15721	ANGLE(I) = 8.9346
I = 59	XSHD(I) = 1.56720	YSHD(I) = 1.40000	DYSDX(I) = .15858	ANGLE(I) = 9.0108
I = 60	XSHD(I) = 1.63510	YSHD(I) = 1.41080	DYSDX(I) = .15840	ANGLE(I) = 9.0008
I = 61	XSHD(I) = 1.70290	YSHD(I) = 1.42150	DYSDX(I) = .15845	ANGLE(I) = 9.0037
I = 62	XSHD(I) = 1.77080	YSHD(I) = 1.43230	DYSDX(I) = .15842	ANGLE(I) = 9.0020
I = 63	XSHD(I) = 1.83860	YSHD(I) = 1.44300	DYSDX(I) = .15849	ANGLE(I) = 9.0060
I = 64	XSHD(I) = 1.90650	YSHD(I) = 1.45380	DYSDX(I) = .15824	ANGLE(I) = 8.9917
I = 65	XSHD(I) = 1.97440	YSHD(I) = 1.46450	DYSDX(I) = .15849	ANGLE(I) = 9.0060
I = 66	XSHD(I) = 2.04220	YSHD(I) = 1.47530	DYSDX(I) = .15843	ANGLE(I) = 9.0025
I = 67	XSHD(I) = 2.11010	YSHD(I) = 1.48600	DYSDX(I) = .15843	ANGLE(I) = 9.0025
I = 68	XSHD(I) = 2.17790	YSHD(I) = 1.49680	DYSDX(I) = .15849	ANGLE(I) = 9.0060
I = 69	XSHD(I) = 2.24580	YSHD(I) = 1.50750	DYSDX(I) = .15824	ANGLE(I) = 8.9917
I = 70	XSHD(I) = 2.31370	YSHD(I) = 1.51830	DYSDX(I) = .15849	ANGLE(I) = 9.0059
I = 71	XSHD(I) = 2.38150	YSHD(I) = 1.52900	DYSDX(I) = .15843	ANGLE(I) = 9.0023
I = 72	XSHD(I) = 2.44940	YSHD(I) = 1.53980	DYSDX(I) = .15843	ANGLE(I) = 9.0023
I = 73	XSHD(I) = 2.51720	YSHD(I) = 1.55050	DYSDX(I) = .15849	ANGLE(I) = 9.0059
I = 74	XSHD(I) = 2.58510	YSHD(I) = 1.56130	DYSDX(I) = .15824	ANGLE(I) = 8.9917
I = 75	XSHD(I) = 2.65300	YSHD(I) = 1.57200	DYSDX(I) = .15849	ANGLE(I) = 9.0060
I = 76	XSHD(I) = 2.72080	YSHD(I) = 1.58280	DYSDX(I) = .15843	ANGLE(I) = 9.0025
I = 77	XSHD(I) = 2.78870	YSHD(I) = 1.59350	DYSDX(I) = .15843	ANGLE(I) = 9.0024
I = 78	XSHD(I) = 2.85650	YSHD(I) = 1.60430	DYSDX(I) = .15849	ANGLE(I) = 9.0061
I = 79	XSHD(I) = 2.92440	YSHD(I) = 1.61500	DYSDX(I) = .15823	ANGLE(I) = 8.9914
I = 80	XSHD(I) = 2.99230	YSHD(I) = 1.62580	DYSDX(I) = .15851	ANGLE(I) = 9.0070
I = 81	XSHD(I) = 3.06010	YSHD(I) = 1.63650	DYSDX(I) = .15835	ANGLE(I) = 8.9983
I = 82	XSHD(I) = 3.12800	YSHD(I) = 1.64730	DYSDX(I) = .15870	ANGLE(I) = 9.0174
I = 83	XSHD(I) = 3.19580	YSHD(I) = 1.65800	DYSDX(I) = .15748	ANGLE(I) = 8.9497
I = 84	XSHD(I) = 3.69580	YSHD(I) = 1.73720	DYSDX(I) = .15865	ANGLE(I) = 9.0150
I = 85	XSHD(I) = 4.19580	YSHD(I) = 1.81640	DYSDX(I) = .15830	ANGLE(I) = 8.9954
I = 86	XSHD(I) = 5.19580	YSHD(I) = 1.97480	DYSDX(I) = .15848	ANGLE(I) = 9.0052

*****BRITISH EJECTOR NOZZLE, DSHD/DPRIM = 1.225, LSHD/DPRIM = 0.220*****

WTFL = .100000 WLEAK = 0.000000 PTS/PTP = .294203 AS/AS* = 1.000482

XSLP	YSLP	AMP	THETA	P/PTP	AMS	P/PTS	AS/AS*
0.00000	1.00000	1.46709	-5.06323	.28564	.20582	.97090	2.88378
.00157	.99988	1.46709	-3.97018	.28564	.20580	.97091	2.88400
.00594	.99962	1.46732	-2.84841	.28555	.20696	.97059	2.86869
.01245	.99936	1.46774	-1.71882	.28537	.20908	.96999	2.84112
.02167	.99917	1.46845	-.58918	.28509	.21253	.96901	2.79742
.03364	.99916	1.46949	.54414	.28466	.21756	.96756	2.73620
.04779	.99944	1.47089	1.66407	.28409	.22411	.96562	2.66084
.06523	1.00011	1.47287	2.78193	.28328	.23311	.96287	2.56434
.08658	1.00135	1.47570	3.89434	.28213	.24542	.95896	2.44424
.11287	1.00339	1.47976	5.00112	.28049	.26205	.95338	2.30059
.14203	1.00622	1.48518	6.11199	.27830	.28286	.94595	2.14569
.18647	1.01136	1.49284	7.15251	.27524	.30998	.93555	1.97663
.24357	1.01910	1.51113	8.38723	.26804	.36720	.91107	1.70698
.32327	1.03178	1.54623	9.87796	.25466	.45879	.86561	1.42748
.47936	1.06246	1.64037	12.35298	.22156	.64959	.75309	1.13600
.63347	1.09646	1.67834	12.53002	.20931	.71471	.71145	1.08409
.76842	1.11873	1.66036	10.95857	.21503	.68446	.73091	1.10611
.88019	1.13745	1.66837	10.23643	.21247	.69805	.72218	1.09579
.97449	1.16021	1.75557	11.91014	.18624	.83533	.63303	1.02519
1.04058	1.17925	1.85164	14.01403	.16079	.97061	.54652	1.00073
1.11093	1.19308	1.82770	12.57312	.16681	.93807	.56700	1.00332
1.23534	1.21410	1.82137	11.08902	.16844	.92936	.57252	1.00434
1.37102	1.23751	1.84840	10.44173	.16159	.96625	.54925	1.00097
1.51851	1.26314	1.89215	10.15028	.15106	1.02421	.51345	1.00048
1.67713	1.29010	1.94136	9.89824	.13999	1.08720	.47582	1.00606
1.85218	1.31893	1.99617	9.62528	.12857	1.15505	.43701	1.01861
2.09473	1.35774	2.07632	9.35794	.11347	1.25070	.38570	1.04700
2.36228	1.39906	2.16066	9.06918	.09946	1.34766	.33807	1.08790
2.66395	1.44446	2.25307	8.81416	.08607	1.45043	.29254	1.14422
2.96532	1.48861	2.34051	8.57415	.07506	1.54497	.25514	1.20786
3.19592	1.52158	2.39954	8.37376	.06845	1.60754	.23266	1.25635

*****BRITISH EJECTOR NOZZLE, DSHD/DPRIM = 1.225, LSHD/DPRIM = 0.220*****

WTFL = .100000 PTS/PTS = .294203 REYPRM = 4.000E+06 REYSEC = 3.562E+05

XSHD	YSHD	AMS	DELSHD	THETAS	CFSHD
0.00000	1.56376	.20582	.00125	.00096	.00771
.00157	1.56219	.20580	.00126	.00097	.00769
.00594	1.55781	.20696	.00127	.00098	.00767
.01245	1.55130	.20908	.00127	.00098	.00765
.02167	1.54209	.21253	.00127	.00097	.00763
.03364	1.53013	.21756	.00125	.00096	.00762
.04779	1.51600	.22411	.00122	.00093	.00761
.06523	1.49857	.23311	.00118	.00090	.00761
.08658	1.47720	.24542	.00113	.00086	.00760
.11287	1.45083	.26205	.00107	.00081	.00759
.14203	1.42167	.28286	.00100	.00076	.00757
.18647	1.37723	.30998	.00099	.00075	.00741
.24357	1.32013	.36720	.00095	.00071	.00724
.32327	1.25523	.45879	.00092	.00067	.00690
.47936	1.22781	.64959	.00112	.00078	.00603
.63347	1.25219	.71471	.00129	.00085	.00548
.76842	1.27355	.68446	.00172	.00113	.00504
.88019	1.29126	.69805	.00215	.00141	.00476
.97449	1.30616	.83533	.00201	.00127	.00472
1.04058	1.31667	.97061	.00166	.00099	.00477
1.11093	1.32778	.93807	.00177	.00102	.00464
1.23534	1.34748	.92936	.00230	.00134	.00434
1.37102	1.36898	.96625	.00270	.00157	.00415
1.51851	1.39233	1.02421	.00297	.00168	.00400
1.67713	1.41744	1.08720	.00322	.00177	.00387
1.85218	1.44516	1.15505	.00353	.00187	.00374
2.09473	1.48358	1.25070	.00403	.00205	.00357
2.36228	1.52597	1.34766	.00460	.00222	.00341
2.66395	1.57374	1.45043	.00533	.00243	.00326
2.96532	1.62150	1.54497	.00612	.00264	.00313
3.19592	1.65802	1.60754	.00676	.00280	.00304

*****BRITISH EJECTOR NOZZLE, DSHD/DPRIM = 1.225, LSHD/DPRIM = 0.220*****

XSONIC = 0.00000	YSONIC = 1.00000	TSONIC = -16.00000
XSONIC = -.00018	YSONIC = .99785	TSONIC = -14.93333
XSONIC = -.00023	YSONIC = .99251	TSONIC = -13.86667
XSONIC = .00036	YSONIC = .98506	TSONIC = -12.80000
XSONIC = .00168	YSONIC = .97531	TSONIC = -11.73333
XSONIC = .00402	YSONIC = .96361	TSONIC = -10.66667
XSONIC = .00753	YSONIC = .94949	TSONIC = -9.60000
XSONIC = .01240	YSONIC = .93241	TSONIC = -8.53333
XSONIC = .01899	YSONIC = .91210	TSONIC = -7.46667
XSONIC = .02776	YSONIC = .88783	TSONIC = -6.40000
XSONIC = .03938	YSONIC = .85860	TSONIC = -5.33333
XSONIC = .05403	YSONIC = .81616	TSONIC = -4.26667
XSONIC = .07491	YSONIC = .76684	TSONIC = -3.20000
XSONIC = .10385	YSONIC = .68708	TSONIC = -2.13333
XSONIC = .15166	YSONIC = .55429	TSONIC = -1.06667
XSONIC = .19081	YSONIC = .40921	TSONIC = -.53333
XSONIC = .22212	YSONIC = .28076	TSONIC = -.26667
XSONIC = .24372	YSONIC = .17938	TSONIC = -.13333
XSONIC = .25717	YSONIC = .10309	TSONIC = -.06667
XSONIC = .26472	YSONIC = .05507	TSONIC = -.03333
XSONIC = .27180	YSONIC = .00000	TSONIC = 0.00000

```
*****BRITISH EJECTOR NOZZLE, DSHD/DPRIM = 1.225, LSHD/DPRIM = 0.220*****

    SECONDARY CORRECTED WEIGHT FLOW RATIO, WTFL =  .100000

    SECONDARY TOTAL PRESSURE RATIO, PTS/PTP =  .294203

    SECONDARY CRITICAL AREA RATIO, AS*/AP* =  .339901

    PRIMARY NOZZLE FLOW COEFFICIENT, CFL =  .96844

    PRIMARY NOZZLE VELOCITY COEFFICIENT, CVL =  .98910

    NOZZLE PRESSURE RATIO, PTP/P0 =  14.60958

    PRIMARY STREAM THRUST, FP/(PTP*AP) = 1.236697

    SECONDARY STREAM THRUST, FS/(PTP*AP) =  .427614

    PRESSURE FORCE ON SHROUD, FSHD/(PTP*AP) =  -.096920

    PRESSURE FORCE ON BODY, FBDY/(PTP*AP) = 0.000000

    SKIN FRICTION DRAG, FDRAG/(PTP*AP) =  .003662

    TOTAL STREAM THRUST, FT/(PTP*AP) = 1.563728

    GROSS STREAM THRUST, FGROSS/(PTP*AP) = 1.375562
```

EJECTOR THRUST CHARACTERISTICS

****

PTP/PO	FGROSS	FIP	FIS	CVP	CV
2.00000	.18921	.74366	0.00000	.25444	.25444
3.00000	.64739	.91063	0.00000	.71092	.71092
4.00000	.87647	1.00334	.03862	.87355	.84117
5.00000	1.01392	1.06519	.05854	.95187	.90228
6.00000	1.10556	1.11053	.07013	.99552	.93639
7.00000	1.17101	1.14576	.07824	1.02204	.95671
8.00000	1.22010	1.17424	.08438	1.03905	.96939
9.00000	1.25828	1.19795	.08928	1.05036	.97751
10.00000	1.28883	1.21811	.09331	1.05805	.98276
11.00000	1.31382	1.23557	.09672	1.06333	.98613
12.00000	1.33464	1.25089	.09966	1.06695	.98822
13.00000	1.35226	1.26449	.10222	1.06941	.98943
14.00000	1.36737	1.27669	.10449	1.07103	.99000
15.00000	1.38046	1.28771	.10651	1.07203	.99013
16.00000	1.39191	1.29775	.10834	1.07256	.98992
17.00000	1.40202	1.30694	.11000	1.07275	.98948
18.00000	1.41100	1.31540	.11151	1.07268	.98885
19.00000	1.41904	1.32323	.11291	1.07241	.98810
20.00000	1.42628	1.33050	.11419	1.07198	.98725
21.00000	1.43282	1.33729	.11538	1.07144	.98633
22.00000	1.43877	1.34364	.11650	1.07080	.98537
23.00000	1.44421	1.34961	.11753	1.07009	.98437
24.00000	1.44919	1.35522	.11851	1.06933	.98335
25.00000	1.45377	1.36052	.11942	1.06854	.98231
26.00000	1.45800	1.36554	.12028	1.06771	.98127
27.00000	1.46191	1.37029	.12110	1.06686	.98023
28.00000	1.46555	1.37482	.12187	1.06600	.97920
29.00000	1.46893	1.37912	.12261	1.06513	.97817
30.00000	1.47209	1.38322	.12330	1.06425	.97715
31.00000	1.47505	1.38714	.12397	1.06337	.97614
32.00000	1.47782	1.39089	.12460	1.06250	.97514
33.00000	1.48042	1.39449	.12521	1.06163	.97416
34.00000	1.48287	1.39794	.12579	1.06076	.97319
35.00000	1.48518	1.40125	.12634	1.05990	.97224
36.00000	1.48737	1.40443	.12688	1.05905	.97130
37.00000	1.48943	1.40750	.12739	1.05821	.97038
38.00000	1.49139	1.41045	.12788	1.05738	.96948
39.00000	1.49324	1.41330	.12836	1.05656	.96859
40.00000	1.49500	1.41606	.12882	1.05575	.96772

APPENDIX E

COMPUTER PROGRAM

```
                PROGRAM CAUCP (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=OUTPUT)
          C
          C                     EJECTOR NOZZLE COMPUTER PROGRAM
          C
          C     INPUT PARAMETERS
          C
          C     WTFL = RATIO OF SECONDARY TO PRIMARY CORRECTED WEIGHT FLOW
          C     HSHP = RATIO OF SECONDARY TO PRIMARY TOTAL PRESSURE
          C     TOS = TOTAL TEMPERATURE OF SECONDARY FLOW
          C     GAMS = RATIO OF CP/CV OF SECONDARY FLOW
          C     TOP = TOTAL TEMPERATURE OF PRIMARY FLOW
          C     GAMP = RATIO OF CP/CV OF PRIMARY FLOW
          C     AMR = INITIAL PRIMARY MACH NUMBER
          C     ANGR = ANGLE OF PRIMARY LIP (DEGREES)
          C     YRATIO - PRIMARY NOZZLE RADIUS RATIO
          C     XPRIM = PRIMARY FLOW STATION
          C     DPRIM = DIAMETER OF PRIMARY
          C     DSHD = DIAMETER OF SHROUD
          C     DBDY = DIAMETER OF BODY AT PRIMARY FLOW STATION
          C     CONA = CONE CENTERBODY ANGLE (DEGREES)
          C     END = EJECTOR LENGTH
          C
          C     COMMAND PARAMETERS
          C
          C     SET FDIM = 0.0 FOR TWO DIMENSIONAL FLOW
          C     SET FDIM = 1.0 FOR AXISYMMETRIC FLOW
          C     SET SOLVE = 0.0 FOR NON-MIXING SOLUTION
          C     SET SOLVE = 1.0 FOR MIXING SOLUTION
          C     SET SOLVE = 2.0 FOR IMPINGEMENT SOLUTION
          C     SET PRINT = 0.0 FOR NO PRINT-OUT OF PRIMARY FLOW FIELD
          C     SET PRINT = 1.0 FOR PRINT-OUT OF FINAL PRIMARY FLOW FIELD
          C     SET PRINT = 2.0 FOR PRINT-OUT OF EVERY PRIMARY FLOW FIELD
          C     SET IPLOT = 0 FOR NO CALCOMP PLOT
          C     SET IPLOT = 1 FOR CALCOMP PLOT ON EVERY SOLUTION
          C     SET IPLOT = 2 FOR CALCOMP PLOT ON FINAL SOLUTION
          C
000003          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1       PHS(100),ASASS(100),DASDX(100),ISLP
000003          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000003          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000003          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003          COMMON/BNLYR/XSUM(100),YSUM(100),DELSHD(100),THETAS(100),
               1       CFSHD(100),XCNE(100),YCNE(100),DELCNE(100),THETAC(100),
               2       CFCNE(100),REYPRM,POP,AOP,VOP,REYSEC,POS,AOS,VOS,PEX,XSCALE
000003          COMMON/PRFLR/XPRF(6),YPRF(6,25),QPRF(6,25),NPRF(6),MAX,NMAX
000003          COMMON/CPLOT/IPLOT,XSTART,YSTART,XSPAN,YSPAN,SCALE,SPAN,AXIS,
               1     XORGN,YORGN,XSHFT,YSHFT,KKK(14),PP(14),XDOWN(100),YACROS(100)
000003          COMMON/OUTPN/IPNCH,IPRNT,ICOMP
000003          REAL K1,K2,K3,K4,MACH
          C
          C     FUNCTION STATEMENTS
          C
000003          FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000023          FUNP(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-G/(G-1.0))
000037          FUNQ(G,AM)=SQRT((G-1.0)/2.0*AM*AM/(1.0+(G-1.0)/2.0*AM*AM))
000055          FUNW(G,VEL)=SQRT(2.0/(G-1.0)*VEL*VEL/(1.0-VEL*VEL))
          C
          C     PROGRAM EJECT
          C
000072        5 READ (5,500) (TITLE(I), I=1,18)
000104          IF (EOF,5) 7,9
000107        7 STOP
000111        9 CONTINUE
000111          READ (5,502) WTFL,HSHP,TOS,GAMS,TOP,GAMP
```

```
000131          READ (5,502) AMR,ANGR,YRATIO,XPRIM,YPRIM,DSHD
000151          READ (5,502) DBDY,CONA,END,REYPRM,DELSHD(1),FDIM
000171          READ (5,502) K1,K2,K3,K4,SOLVE,PRINT
000211          READ (5,504) NDATA,NSHD,NBDY,NITER,IPLOT,IPNCH,IPRNT,ICOMP
000235          IF (NITER .GT. 0) READ (5,502) (PTS(I),AREA(I),WLEAK(I),I=1,NITER)
000255          IF (IPLOT .GT. 0) READ (5,502) XSTART,YSTART,XSPAN,YSPAN,
               1                                AXIS,SCALE
000276          IF (ICOMP .EQ. 2) READ (5,506) XPRF(1),XPRF(6)
000310          IF (NSHD .GT. 1) READ (5,502) (XSHD(I),YSHD(I), I=1,NSHD)
000327          IF (NBDY .GT. 0) READ (5,502) (XBDY(I),YBDY(I), I=1,NBDY)
000345      500 FORMAT (18A4)
000345      502 FORMAT (6E12.0)
000345      504 FORMAT (8I6)
000345      506 FORMAT (2E12.0)
000345          NSTOP=6
000346          TRY=0.0
000347          CASE =0.0
000350          CHOKE=-1.0
000351          IF (ICOMP .EQ. 0 .AND. NBDY .GT. 0) ICOMP=1
000361          IF (ANGR .LT. 0.0) TRY=1.0
000364          CALL START
000365          IF (TRY .EQ. 0.0) CALL DATUM
000367          IF (IPLOT .GT. 0) PRINT=-1.0
000372       10 NITER=1+NITER
000374          ISLP=1
000375          SKIP=0.0
000376          TYPE=1.0
000377          STAG=0.0
000400          ICONE=1
000401          IF (WTFL .EQ. 0.0) STAG=-1.0
000403          CHANGE=0.0
000404          CHARGE=0.0
000405          IF (IPLOT .EQ. 1) CALL PLOTC
000407          IF (IPLOT .EQ. 1) SKIP=1.0
000413          IF (PRINT .EQ. 1.0 .AND. CHOKE .NE. -1.0) SKIP=1.0
000424          IF (PRINT .EQ. 2.0) SKIP=1.0
000430          PTS(NITER)=HSHP
000432          WLEAK(NITER)=0.0
000433          CALL CLEAR(0,1)
000435          ASSAPS=FUNG*WTFL/HSHP
000440          XSLP(ISLP)=XPRIM
000442          YSLP(ISLP)=YPRIM
000443          AMS(ISLP)=0.200
000445          AMP(ISLP)=1.500
000446          CALL FLOW(ISLP)
000450          AREA(NITER)=ASASS(ISLP)
000453          IF (POINT .EQ. -1.0) GO TO 10
000455          PAMB=PHP(ISLP)
000456          IF (TRY .EQ. 1.0) CALL SONICS
000461          VEL=FUNQ(GAMP,AMR)
000466          DO 12 J=1,100
000467          X(1,J)=XPRIM
000471          Y(1,J)=YPRIM
000473          IF (J .EQ. 1) DELV=0.0
000476          IF (J .EQ. 2 .OR. J .EQ. 3) DELV=K1*(1.0+K2)**2/3.0
000512          IF (J .GE. 4) DELV=K1*(1.0+K2)**(J-2)
000524          IF (DELV .GT. K3) DELV=K3
000527          VEL=VEL+DELV
000531          MACH=FUNW(GAMP,VEL)
000535          IF (MACH .GT. AMP(ISLP)) MACH=AMP(ISLP)
000541          P(1,J)=FUNP(GAMP,MACH)
000546          T(1,J)=PMER(AMR,ANGR,MACH,GAMP)
000554          IF (J .EQ. 3) VEL=FUNQ(GAMP,AMR)
000562          IF (MACH .EQ. AMP(ISLP)) GO TO 14
000565       12 CONTINUE
000567          CALL EXIT
000570       14 IF (SKIP .EQ. 1.0) CALL OUTFLD(1)
000574          DO 22 I=2,NDATA
000576          X(2,1)=XSONIC(I)
000600          Y(2,1)=YSONIC(I)
000601          P(2,1)=PSONIC(I)
000603          T(2,1)=TSONIC(I)
```

```
000604          IF (TRY .EQ. 1.0) CALL SONPT(I)
000610          DO 16 J=2,100
000612          IF (P(1,J) .EQ. 0.0) GO TO 18
000614          CALL FIELD(J)
000615       16 CONTINUE
000617       18 ISLP=1+ISLP
000621          CALL SLIP(J)
000622          CALL SLID(J)
000624          IF (SOLVE .EQ. 2.0 .AND. POINT .EQ. -1.0) CALL BREAK(J)
000635       20 IF (STAG .EQ. 2.0) GO TO 34
000637          AREA(NITER)=AMIN1(AREA(NITER),ASASS(ISLP))
000644          CALL CLEAR(0,J)
000646          RADIUS=SQRT((X(1,2)-X(1,1))**2+(Y(1,2)-Y(1,1))**2)
000655          NSERT=RADIUS/K4
000660          IF (I .EQ. NDATA .AND. NSERT .GE. 2) CALL INSERT(NSERT-1)
000673          IF (SKIP .EQ. 1.0) CALL OUTFLD(1)
000677          IF (ICOMP .EQ. 2) CALL PROFLE
000702          IF (POINT .NE. 0.0) GO TO 34
000703       22 CONTINUE
000706       24 LSLP=ISLP+1
000710          DO 32 JSLP=LSLP,100
000711          CALL BOUND(2)
000712          DO 26 J=3,100
000714          IF (P(1,J) .EQ. 0.0) GO TO 28
000716          CALL FIELD(J)
000717          CALL CHECK(J,SHOCK)
000721          IF (SHOCK .EQ. 1.0) GO TO 30
000723       26 CONTINUE
000725       28 ISLP=1+ISLP
000727          CALL SLIP(J)
000730          CALL SLID(J)
000732          IF (SOLVE .EQ. 2.0 .AND. POINT .EQ. -1.0) CALL BREAK(J)
000743       30 IF (STAG .EQ. 2.0) GO TO 34
000745          AREA(NITER)=AMIN1(AREA(NITER),ASASS(ISLP))
000752          CALL CLEAR(1,J)
000754          RADIUS=SQRT((X(1,2)-X(1,1))**2+(Y(1,2)-Y(1,1))**2)
000763          NSERT=RADIUS/K4
000766          IF (NSERT .GE. 2) CALL INSERT(NSERT-1)
000774          IF (SKIP .EQ. 1.0) CALL OUTFLD(1)
001000          IF (ICOMP .EQ. 2) CALL PROFLE
001003          IF (POINT .NE. 0.0) GO TO 34
001004       32 CONTINUE
001006       34 IF (CASE .EQ. 1.0) GO TO 38
001010          IF (SOLVE .LE. 1.0) CALL ESTMP
001013          IF (SOLVE .EQ. 2.0 .AND. TYPE .EQ. 0.0) CHOKE=0.0
001023          IF (SOLVE .EQ. 2.0 .AND. TYPE .EQ. 1.0) CALL ESTMW
001033       36 IF (ICOMP .GT. 0 .AND. CHOKE .EQ. 0.0) CALL COMP(K4,SKIP)
001044          CALL OUTSLP
001045          IF (IPLOT .EQ. 1) CALL PLOTL
001050          IF (IPLOT .EQ. 1 .AND. CHOKE .NE. -1.0) GO TO 40
001061          IF (PRINT .EQ. 0.0 .AND. CHOKE .NE. -1.0) GO TO 40
001070          IF (PRINT .EQ. 1.0 .AND. CASE .EQ. 1.0) GO TO 38
001077          IF (PRINT .EQ. 2.0 .AND. CHOKE .NE. -1.0) GO TO 40
001106          IF (CHOKE .NE. -1.0) CASE=1.0
001111          IF (CASE .EQ. 1.0 .AND. IPLOT .EQ. 2) IPLOT=1
001123          GO TO 10
001124       38 IF (ICOMP .GT. 0 .AND. PRINT .EQ. 0.0) CALL COMP(K4,SKIP)
001135          IF (CASE .EQ. 1.0 .AND. IPLOT .EQ. 1) CALL PLOTL
001147       40 GO TO 5
001150          END
```

```
                SUBROUTINE START
          C
          C     CALCULATION OF PERTINENT EJECTOR PARAMERTERS
          C
000002          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
```

```
                 1          PHS(100),ASASS(100),DASDX(100),ISEP
000002            COMMON XTS(21,26),YTS(21,26),W(21),TAU(26),NSONIC,NANGLE
000002            COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002            COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002            COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002            COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002            COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
                 1          XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
                 2          PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
                 3          SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002            COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002            COMMON/ANLYR/XSUM(100),YSUM(100),DELSHD(100),THETAS(100),
                 1          CFSHD(100),XCNE(100),YCNE(100),DELCNE(100),THETAC(100),
                 2          CFCNE(100),REYPRM,POP,AOP,VOP,REYSEC,POS,AOS,VOS,PEX,XSCALF
000002            REAL K1,K2
000002            FUNA(G,AM)=((G+1.0)/2.0)**(-(G+1.0)/(2.0*(G-1.0)))*1.0/AM*(1.0+
                 1              (G-1.0)/2.0*AM*AM)**((G+1.0)/(2.0*(G-1.0)))
000033            FGAM(G)=SQRT(G)*((G+1.0)/2.0)**(-(G+1.0)/(2.0*(G-1.0)))
           C
           C      FORMAT STATEMENTS
           C
000051       600 FORMAT (1H1,//27X,18A4)
000051       602 FORMAT (//35X,35HPRIMARY NOZZLE EXIT MACH NO., AMR =F9.5)
000051       607 FORMAT (//35X,37HPRIMARY NOZZLE EXIT DIAMETER, DPRIM =F9.5)
000051       608 FORMAT (//35X,32HPRIMARY NOZZLE LIP ANGLE, ANGR =F10.5)
000051       610 FORMAT (//35X,43HDIAMETER OF THE CYCLINDRICAL SHROUD, DSHD =F9.5)
000051       611 FORMAT (//35X,35HLOCATION OF PRIMARY NOZZLE, XPRIM =F10.5)
000051       612 FORMAT (//35X,50HEJECTOR LENGTH MEASURED FROM PRIMARY NOZZLE, END
                0:E8<4.
000051       613 FORMAT (//35X,37HPRIMARY NOZZLE RADIUS RATIO, YRATIO =F8.5)
000051       615 FORMAT (//35X,40HPRIMARY NOZZLE REYNOLDS NUMBER, REYPRM =E10.3)
000051       616 FORMAT (//35X,42HTOTAL TEMPERATURE OF SECONDARY FLOW, TOS =F9.3)
000051       617 FORMAT (//35X,50HRATIO OF SPECIFIC HEATS FOR SECONDARY FLOW, GAMS
                1=F8.5)
000051       618 FORMAT (//35X,40HTOTAL TEMPERATURE OF PRIMARY FLOW, TOP =F9.3)
000051       619 FORMAT (//35X,48HRATIO OF SPECIFIC HEATS FOR PRIMARY FLOW, GAMP =F
                18.5)
000051       620 FORMAT (//35X,46HFLOW IN EJECTOR IS TWO DIMENSIONAL, FDIM = 0.0)
000051       622 FORMAT (//35X,43HFLOW IN EJECTOR IS AXISYMMETRIC, FDIM = 1.0)
000051       624 FORMAT (//35X,49HNUMBER OF POINTS SPECIFING SHROUD CONTOUR, NSHD =
                1I3)
000051       626 FORMAT (//35X,47HNUMBER OF POINTS SPECIFING PLUG CONTOUR, NBDY =I3
                1)
000051       628 FORMAT (//35X,50HNUMBER OF POINTS SPECIFING PLUG CONTOUR, NBDY = 4
                15)
000051       630 FORMAT (//35X,56HNUMBER OF POINTS SPECIFING ENTERENCE CONDITIONS,
                1NDATA =I3)
000051       632 FORMAT (1H1,//51X,29HSHROUD CONTOUR SPECIFICATIONS//)
000051       634 FORMAT (19X,3HI =I3,4X,9HXSHD(I) =F9.5,4X,9HYSHD(I) =F8.5,4X,
                110HDYSDX(I) =F9.5,4X,10HANGLE(I) =F9.4)
000051       636 FORMAT (1H1,//52X,27HPLUG CONTOUR SPECIFICATIONS//)
000051       638 FORMAT (19X,3HI =I3,4X,9HXBDY(I) =F8.5,4X,9HYBDY(I) =F8.5,4X,
                110HDYBDX(I) =F9.5,4X,10HANGLE(I) =F9.4)
           C
           C      SUBROUTINE START
           C
000051            END=END-XPRIM
000053            IF (ANGR .GT. 0.0) ANGR=0.0
000055            WRITE (6,600) (TITLE(I),I=1,18)
000067            WRITE (6,607) DPRIM
000075            WRITE (6,602) AMR
000103            WRITE (6,608) ANGR
000111            WRITE (6,613) YRATIO
000117            WRITE (6,611) XPRIM
000125            IF (NSHD .EQ. 1) WRITE (6,610) DSHD
000135            WRITE (6,612) END
000143            WRITE (6,615) REYPRM
000151            WRITE (6,616) TOS
000157            WRITE (6,617) GAMS
000165            WRITE (6,618) TOP
000173            WRITE (6,619) GAMP
000201            IF (FDIM .EQ. 0.0) WRITE (6,620)
```

```
000206          IF (FDIM .EQ. 1.0) WRITE (6,622)
000214          WRITE (6,624) NSHD
000222          IF (ABS(CONA) .EQ. 0.0) WRITE (6,626) NBDY
000232          IF (ABS(CONA) .GT. 0.0) WRITE (6,628)
000240          WRITE (6,630) NDATA
000246          LCHK=45
000247          PI=3.1415927
000251          CONVR=0.01745329
000252          CONVA=1.0/CONVR
000254          ANGR=CONVR*ANGR
000255          DPREF=DPRIM
000257          YPRIM=1.0
000260          IF (NSHD .EQ. 1) XSHD(1)=0.0
000263          IF (NSHD .EQ. 1) YSHD(1)=DSHD/2.0
000266          DO 10 I=1,NSHD
000270          XSHD(I)=2.0*(XSHD(I)-XPRIM)/DPREF
000274          YSHD(I)=2.0*YSHD(I)/DPREF
000277       10 CONTINUE
000301          IF (NSHD .LE. 1) GO TO 14
000303          CALL SPLINE(XSHD,YSHD,NSHD,DYSDX,D2FDX2)
000307          LINE=0
000310          WRITE (6,632)
000314          DO 12 I=1,NSHD
000316          LINE=1+LINE
000320          ANGLE=CONVA*ATAN(DYSDX(I))
000323          WRITE (6,634) I,XSHD(I),YSHD(I),DYSDX(I),ANGLE
000340          IF (LINE .LT. LCHK) GO TO 12
000343          LINE=0
000343          IF (I .LT. NSHD) WRITE (6,632)
000351       12 CONTINUE
000354       14 IF (ABS(CONA) .GT. 0.0) CALL CONE
000360          IF (NBDY .EQ. 0) GO TO 20
000361          DO 16 I=1,NBDY
000363          XBDY(I)=2.0*(XBDY(I)-XPRIM)/DPREF
000367          YBDY(I)=2.0*YBDY(I)/DPREF
000372       16 CONTINUE
000374          CALL SPLINE(XBDY,YBDY,NBDY,DYBDX,D2FDX2)
000377          LINE=0
000400          WRITE (6,636)
000404          DO 18 I=1,NBDY
000406          LINE=1+LINE
000410          ANGLE=CONVA*ATAN(DYBDX(I))
000413          WRITE (6,638) I,XBDY(I),YBDY(I),DYBDX(I),ANGLE
000430          IF (LINE .LT. LCHK) GO TO 18
000433          LINE=0
000433          IF (I .LT. NBDY) WRITE (6,636)
000441       18 CONTINUE
000444       20 XPRIM=0.0
000445          CALL FIND(XPRIM,YBODY,SLOPE,1.0)
000450          APRIM=(YPRIM+YBODY)*(YPRIM-YBODY)**FDIM*COS(ATAN(SLOPE))
000465          END=2.0*END/DPREF
000467          APREF=FUNA(GAMP,AMR)
000472          FUNG=FGAM(GAMP)/FGAM(GAMS)
000501          XSCALE=DPRIM/24.0
000503          IF (DPRIM .EQ. 2.0) XSCALE=6.0/24.0
000506          CFL=1.0
000510          CVL=1.0
000511       22 RETURN
000512          END
```

```
                SUBROUTINE DATUM
          C
          C     CALCULATION OF ENTERING FLOW CONDITIONS
          C
000002          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
              1        PHS(100),ASASS(100),DASDX(100),ISLP
```

```
000002        COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000002        COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002        COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002        COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002        COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002        COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
             1        XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
             2        PAMB,YRATIO,P1,CONVA,CONVR,FDIM,NDATA,NSTOP,
             3        SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002        COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002        COMMON/BNLYR/XSUM(100),YSUM(100),DELSHD(100),THETAS(100),
             1        CFSHD(100),XCNE(100),YCNE(100),DELCNE(100),THETAC(100),
             2        CFCNE(100),REYPRM,POP,AOP,VOP,REYSEC,POS,AOS,VOS,PFX,XSCALE
000002        DIMENSION D(2),AMQ(21)
000002        REAL MUAVE
000002        FUNMU(AM)=ASIN(1.0/AM)
000012        FUNM(G,PR)=SQRT(2.0/(G-1.0)*(PR**(-(G-1.0)/G)-1.0))
000031        FUNP(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-G/(G-1.0))
000045        FUNQ(G,AM)=SQRT(AM*AM/(1.0+(G-1.0)/(G+1.0)*(AM*AM-1.0)))
000065        FUNR(G,Q)=SQRT(((1.0-(G-1.0)/(G+1.0))*Q*Q)/
             1              (1.0-(G-1.0)/(G+1.0)*Q*Q))
000106        FUNT(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-1.0)
000120        FUNU(TEMP)=2.27*32.17*TEMP**1.5/(198.6+TEMP)*1.E-8
000133        AVE(X1,X2)=(X1+X2)/2.0
000141    600 FORMAT (1H1,//28X,12A6,///)
         C
         C      SUBROUTINE DATUM
         C
000141        XP=0.0
000142        DIV=NDATA-1
000145        ISONIC=NDATA
000146        NANGLE=NDATA
000147        CALL FIND(XP,YP,DYPDX,1.0)
000152        DELY=(1.0-YP)/DIV
000155        ANGLE=ABS(ATAN(DYPDX))
000161        ICONE=1
000162        XREF=0.0
000163        TCNE=ABS(ATAN(DYPDX))
000167        DELCNE(ICONE)=0.0
000171        IF (REYPRM .EQ. 0.0) GO TO 6
000172        TP=TOP*FUNT(GAMP,AMR)
000175        AP=49.02*SQRT(TP)
000200        VP=2.0*AP/REYPRM*XSCALE*AMR
000204        PP=53.3*TP*FUNU(TP)/VP
000211        POP=PP/FUNP(GAMP,AMR)
000214        AOP=49.02*SQRT(TOP)
000217        VOP=53.3*TOP*FUNU(TOP)/POP
000224      6 IF (ANGLE .EQ. 0.0) GO TO 8
000225        YCONE(ICONE)=YP
000227        PCONE(ICONE)=FUNP(GAMP,AMR)
000233        CALL CNLYR(DDSDX)
000235        YP=YP+DELCNE(ICONE)
000240        DYPDX=DYPDX+DDSDX
000242        XREF=XP-YP/DYPDX
000244        ANGLE=ABS(ATAN(DYPDX))
000250      8 DO 10 I=1,NDATA
000252        XSONIC(I)=0.0
000253        YSONIC(I)=1.0
000255        PSONIC(I)=FUNP(GAMP,AMR)
000260        TSONIC(I)=ATAN(DYPDX)
000264        AMQ(I)=FUNM(GAMP,PSONIC(I))
000272     10 CONTINUE
000275        IF (FDIM .EQ. 0.0 .OR. NBDY .EQ. 0) GO TO 12
000304        QP=FUNQ(GAMP,AMR)
000310        UP=QP*COS(ANGLE)
000313        VP=QP*SIN(ANGLE)
000316        VUP=-1.0/TAN(ANGLE)
000321        CALL CONIC(XREF,YSONIC(1),UP,VP,VUP,GAMP)
000325        QP=SQRT(UP*UP+VP*VP)
000332        AMQ(1)=FUNR(GAMP,QP)
000336        PSONIC(1)=FUNP(GAMP,AMQ(1))
000340        TSONIC(1)=-ATAN(VP/UP)
```

```
000344           ANGR=TSONIC(1)
000345        12 DO 18 I=2,NDATA
000347           IF (I .EQ. NDATA) GO TO 16
000351           XSONIC(I)=XSONIC(I-1)
000352           YSONIC(I)=YSONIC(I-1)-DELY
000355           DYQDX=0.0
000355           IF (XREF .GT. 0.0) DYQDX=-YSONIC(I)/XREF
000360           IF (FDIM .EQ. 0.0 .OR. NBDY .EQ. 0) GO TO 14
000367           QP=FUNQ(GAMP,AMR)
000373           UP=QP*COS(ANGLE)
000376           VP=QP*SIN(ANGLE)
000401           VUP=-1.0/TAN(ANGLE)
000404           CALL CONIC(XREF,YSONIC(I),UP,VP,VUP,GAMP)
000410           QP=SQRT(UP*UP+VP*VP)
000415           AMQ(I)=FUNR(GAMP,QP)
000422           PSONIC(I)=FUNP(GAMP,AMQ(I))
000426           TSONIC(I)=-ATAN(VP/UP)
000434        14 TAVE=AVE(TSONIC(I),TSONIC(I-1))
000437           MUAVE=AVE(FUNMU(AMQ(I)),FUNMU(AMQ(I-1)))
000447           D(2)=TAN(TAVE-MUAVE)
000453           XSONIC(I)=(YSONIC(I-1)-YSONIC(I)-D(2)*XSONIC(I-1))/(DYQDX-D(2))
000462           YSONIC(I)=YSONIC(I-1)+D(2)*(XSONIC(I)-XSONIC(I-1))
000467           GO TO 18
000467        16 XSONIC(I)=XSONIC(I-1)
000471           CALL FIND(XSONIC(I),YSONIC(I),DYQDX,1.0)
000474           DYQDX=DYQDX+DDSDX
000476           YSONIC(I)=YSONIC(I)+DELCNE(ICONE)
000501           TAVE=AVE(TSONIC(I),TSONIC(I-1))
000504           MUAVE=AVE(FUNMU(AMQ(I)),FUNMU(AMQ(I-1)))
000514           D(2)=TAN(TAVE-MUAVE)
000520           XSONIC(I)=(YSONIC(I-1)-YSONIC(I)-D(2)*XSONIC(I-1)+DYQDX*XSONIC(I))
               1           /(DYQDX-D(2))
000530        18 CONTINUE
000533           XCONE(ICONE)=XSONIC(NDATA)-DELCNE(ICONE)*SIN(TCNE)
000541           CALL FIND(XCONE(ICONE),YCONE(ICONE),DYPDX,1.0)
000544           PCONE(ICONE)=PSONIC(NDATA)
000547           TCONE(ICONE)=ATAN(DYPDX)
000552           AMR=FUNM(GAMP,PSONIC(1))
000557           RETURN
000560           END
```

```
                 SUBROUTINE SONICS
          C
          C      CONSTRUCTION OF ISOCLOINES FOR PRIMARY FLOW FIELD
          C
000002           COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002           COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1        PHS(100),ASASS(100),DASDX(100),ISLP
000002           COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000002           COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002           COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002           COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002           COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002           COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1        XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2        PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3        SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002           COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002           REAL MATCH,MAMB,MAPRCH
          C
000002           FUNM(G,V)=SQRT(2.0/(G+1.0)*V*V/(1.0-(G-1.0)/(G+1.0)*V*V))
000023           FUNP(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-G/(G-1.0))
000037           FUNR(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-1.0/(G-1.0))
000053           FUNQ(G,P)=SQRT((G+1.0)/(G-1.0)*(1.0-P**((G-1.0)/G)))
000073           FUNV(G,AM)=SQRT((G+1.0)/2.0*AM*AM/(1.0+(G-1.0)/2.0*AM*AM))
000112           OMEGA(V,AM)=ALOG(2.0*V*AM/SQRT(1.0-AM*AM)/(1.0+SQRT(1.0+V*V*
               1                    AM*AM/(1.0-AM*AM))))
```

```
      AVE(X1,X2)=(X1+X2)/2.0
  600 FORMAT (1H1,//40X,48HUNABLE TO OBTAIN CONVERGENCE IN SUBROUTINE SO
     1NIC////)
C
C     SUBROUTINE SONICS
C
      NMAX=26
      INNER=0
      NSONIC=21
      WTFLW=0.80
      NANGLE=NDATA
      ANGLE=CONVA*ABS(ANGR)
      ERROR=0.0001
      MATCH=0.99-0.001*(ANGLE-10.0)
      VMATCH=FUNV(GAMP,MATCH)
      VAMB=FUNU(GAMP,PAMB)
      MAMB=FUNM(GAMP,VAMB)
      VMAX=VAMB/VMATCH
   10 INNER=1+INNER
      ITER=0
      VELAPR=0.20
      RHOVEL=WTFLW/(YRATIO*YRATIO**FDIM)
   12 ITER=1+ITER
      VAPRCH=VELAPR
      VELSTR=VAPRCH*VMATCH
      MAPRCH=FUNM(GAMP,VELSTR)
      VELAPR=RHOVEL*FUNR(GAMP,MATCH)/FUNR(GAMP,MAPRCH)
      TEST=ABS(VELAPR-VAPRCH)
      IF (ITER .LT. 25) GO TO 14
      WRITE (6,600)
      CALL EXIT
   14 IF (TEST .GT. ERROR*VELAPR) GO TO 12
      VMIN=0.750
      IF (NANGLE .GT. NMAX) NANGLE=NMAX
      IF (NANGLE .LE. NSTOP) NANGLE=1+NSTOP
      XNOA=NANGLE-NSTOP
      XNOW=NSONIC-1
      WJ=OMEGA(VMAX,MATCH)
      DELW=WJ-OMEGA(VELAPR,MATCH)
      DELV=(VMAX-VMIN)/XNOW
      XIS(1,1)=0.0
      YIS(1,1)=1.0
      W(1)=OMEGA(VMAX,MATCH)
      TAU(1)=ANGR
      ALPHA=ABS(ANGR)
      DT=ALPHA/XNOA
      DO 16 I=2,NSONIC
      VRATIO=VRATIO-DELV
      IF (VRATIO .LT. 1.0) VRATIO=1.0
      W(I)=OMEGA(VRATIO,MATCH)
      DW=W(I)-W(I-1)
      CALL DZDXDY(W(I),WJ,ALPHA,TAU(1),DELW,DW,0.0,DXS,DYS)
      XIS(I,1)=XIS(I-1,1)+DXS
      YIS(I,1)=YIS(I-1,1)+DYS
      IF (VRATIO .EQ. 1.0) GO TO 18
   16 CONTINUE
   18 WTFLW=0.0
      NCHNGE=NANGLE-NSTOP
      DO 20 J=2,NANGLE
      IF (J .LE. NCHNGE) DT=DT
      IF (J .GT. NCHNGE) DT=DT/2.0
      IF (J .EQ. NANGLE) DT=-TAU(J-1)
      TAU(J)=TAU(J-1)+DT
      AVETAU=AVE(TAU(J),TAU(J-1))
      CALL DZDXDY(W(I),WJ,ALPHA,TAU(J),DELW,0.0,DT,DXS,DYS)
      XIS(I,J)=XIS(I,J-1)+DXS
      YIS(I,J)=YIS(I,J-1)+DYS
      WTFLW=WTFLW+(DYS*COS(AVETAU)-DXS*SIN(AVETAU))
   20 CONTINUE
      SCALE=1.0-YIS(I,J)
      WTFLW=-WTFLW/SCALE
      TEST=ABS(RHOVEL-WTFLW/(YRATIO*YRATIO**FDIM))
```

```
000454           IF (INNER .LT. 25) GO TO 22
000455           WRITE (6,600)
000462           CALL EXIT
000463        22 IF (TEST .GT. ERROR*RHOVEL) GO TO 10
000470           VRATIO=VMAX
000471           DO 24 I=2,NSONIC
000472           VRATIO=VRATIO-DELV
000474           W(I)=OMEGA(VRATIO,MATCH)
000501           DW=W(I)-W(I-1)
000503           CALL DZDXDY(W(I),WJ,ALPHA,TAU(1),DELW,DW,0.0,DXS,DYS)
000514           XIS(I,1)=XIS(I-1,1)+DXS/SCALE
000520           YIS(I,1)=YIS(I-1,1)+DYS/SCALE
000523        24 CONTINUE
000525           DO 28 I=1,NSONIC
000527           DO 26 J=2,NANGLE
000530           DT=TAU(J)-TAU(J-1)
000532           IF (I .EQ. 1 .AND. J .EQ. NANGLE) GO TO 25
000541           CALL DZDXDY(W(I),WJ,ALPHA,TAU(J),DELW,0.0,DT,DXS,DYS)
000552        25 XIS(I,J)=XIS(I,J-1)+DXS/SCALE
000562           YIS(I,J)=YIS(I,J-1)+DYS/SCALE
000565           IF (J .EQ. NANGLE) YIS(I,J)=0.0
000571        26 CONTINUE
000574        28 CONTINUE
000576           ISONIC=1
000577           XSONIC(ISONIC)=XPRIM
000601           YSONIC(ISONIC)=YPRIM
000603           PSONIC(ISONIC)=FUNP(GAMP,AMR)
000606           TSONIC(ISONIC)=TAU(ISONIC)
000610           RETURN
000611           END
```

```
                 SUBROUTINE SONPT(K)
        C
        C        CONSTRUCTION OF SONIC POINT
        C
000003           COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003           COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
                1         PHS(100),ASASS(100),DASDX(100),ISLP
000003           COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003           COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003           COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003           COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003           COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000003           COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
                1         XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
                2         PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
                3         SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000003           COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003           DIMENSION C(16),D(6),ANG(2)
             600 FORMAT (1H1,//40X,48HUNABLE TO OBTAIN CONVERGENCE IN SUBROUTINE SO
                1NPT////)
000003       602 FORMAT (8X,6HITER =I3,4X,8HX(2,1) =F8.5,4X,8HY(2,1) =F8.5,4X,
                18HP(2,1) =F8.5,4X,8HT(2,1) =F8.5,4X,6HTEST =1PE12.5)
000003       604 FORMAT (1H1,//40X,47HUNABLE TO OBTAIN A SOLUTION IN SUBROUTINE SON
                1PT////)
        C
        C        SUBROUTINE SONPT
        C
000003           ITER=0
000004           ERROR=0.0001
000005           DO 10 J=1,100
000007           IF (P(1,J) .EQ. 0.0) GO TO 12
000011           CALL FONICS(K,J,ANG(2))
000013           IF (J .GT. 1 .AND. ANG(2) .GE. T(2,1)) GO TO 14
000027           ANG(1)=ANG(2)
000030        10 CONTINUE
000032        12 WRITE (6,604)
000036           CALL OUTSLP
```

```
000037            CALL OUTSNP
000040            CALL EXIT
000041         14 I=0
000042            JMIN=J-1
000044            DO 18 J=1,100
000047            IF (P(1,J) .EQ. 0.0) GO TO 20
000051            IF (J .LT. JMIN) GO TO 16
000053            I=1+I
000055            X(1,I)=X(1,J)
000060            Y(1,I)=Y(1,J)
000062            P(1,I)=P(1,J)
000065            T(1,I)=T(1,J)
000067            IF (I .EQ. J) GO TO 18
000070         16 X(1,J)=0.0
000072            Y(1,J)=0.0
000074            P(1,J)=0.0
000075            T(1,J)=0.0
000077         18 CONTINUE
000101         20 ITER=1+ITER
000103            CALL COAVE(XAVE,YAVE,PAVE,TAVE,1,1,1,2)
000112            CALL COEFF(XAVE,YAVE,PAVE,TAVE,C,GAMP)
000116            D(1)=C(1)
000120            D(3)=1.0/(C(3)*C(5)*C(7))
000123            D(5)=FDIM*C(9)*C(11)/(C(13)*C(15))
000130            DXDA=(X(1,1)-X(1,2))/(ANG(1)-ANG(2))
000134            DTDA=(T(1,1)-T(1,2))/(ANG(1)-ANG(2))
000140            X(1,1)=X(1,2)+DXDA*(T(2,1)-ANG(2))
000144            Y(1,1)=Y(1,2)+D(1)*(X(1,1)-X(1,2))
000147            T(1,1)=T(1,2)+DTDA*(T(2,1)-ANG(2))
000153            P(1,1)=(D(3)*P(1,2)-(T(1,1)-T(1,2))-D(5)*(Y(1,1)-Y(1,2)))/D(3)
000164            IF (P(1,1) .GT. PSONIC(ISONIC)) GO TO 12
000171            CALL FONICS(K,1,ANG(1))
000173            TEST=ABS(T(2,1)-ANG(1))
000176            IF (ITER .LT. 45) GO TO 22
000202            IF (ITER .EQ. 45) WRITE (6,600)
000207            WRITE (6,602) ITER,X(2,1),Y(2,1),P(2,1),ANG(1),TEST
000227            IF (ITER .EQ. 50) CALL EXIT
000234         22 IF (TEST .GT. ERROR) GO TO 20
000240            ISONIC=1+ISONIC
000241            XSONIC(ISONIC)=X(2,1)
000243            YSONIC(ISONIC)=Y(2,1)
000245            PSONIC(ISONIC)=P(2,1)
000246            TSONIC(ISONIC)=T(2,1)
000250         24 RETURN
000251            END
```

```
                  SUBROUTINE FONICS(K,J,ANGLE)
            C
            C     CONSTRUCTION OF FIRST POINT ALONG ISOCLINE
            C
000006            COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000006            COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
                 1        PHS(100),ASASS(100),DASDX(100),ISLP
000006            COMMON XTS(21,26),YTS(21,26),W(21),TAU(26),NSONIC,NANGLE
000006            COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000006            COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000006            COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000006            COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000006            COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
                 1        XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
                 2        PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
                 3        SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000006            COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000006            DIMENSION A(16),B(16),C(16),D(8)
000006            REAL MACH,MUAVE
000006            FUNMU(AM)=ASIN(1.0/AM)
000020            FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000041            FUNP(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-G/(G-1.0))
```

```
000055          AVE(X1,X2)=(X1+X2)/2.0
000063      500 FORMAT (1H1,//40X,48HSONIC POINT (XP) LIES OUTSIDE RANGE OF ISOCLI
               1NES//)
000063      502 FORMAT (//24X,15HXIS(NSONIC,K) =F9.5,15X,4HXP =F8.5,15X,10HXIS(1,K
               1) =F8.5//)
          C
          C       SUBROUTINE FONICS
          C
000063          T(2,1)=TAU(K)
000065          P(2,1)=FUNP(GAMP,AMR)
000067          MACH=FUNM(GAMP,P(1,J))
000074          MSONIC=NSONIC-1
000076          TAVE=AVE(T(2,1),T(1,J))
000101          MUAVE=AVE(FUNMU(AMR),FUNMU(MACH))
000111          D(2)=TAN(TAVE-MUAVE)
000115          DO 10 I=1,MSONIC
000120          DYDX=(YIS(I+1,K)-YIS(I,K))/(XIS(I+1,K)-XIS(I,K))
000131          X(2,1)=(Y(1,J)-YIS(I,K)-D(2)*X(1,J)+DYDX*XIS(I,K))/(DYDX-D(2))
000147          Y(2,1)=Y(1,J)+D(2)*(X(2,1)-X(1,J))
000154          IF (K .EQ. NANGLE) GO TO 12
000156          IF (X(2,1) .GE. XIS(I+1,K) .AND. X(2,1) .LE. XIS(I,K)) GO TO 12
000170       10 CONTINUE
000172          WRITE (6,500)
000176          WRITE (6,502) XIS(NSONIC,K),X(2,1),XIS(1,K)
000222          CALL OUTSNP
000223          CALL EXIT
000224       12 CALL COAVE(XAVE,YAVE,PAVE,TAVE,1,J,2,1)
000236          CALL COEFF(XAVE,YAVE,PAVE,TAVE,C,GAMP)
000242          D(2)=C(2)
000244          D(4)=1.0/(C(4)*C(6)*C(8))
000247          D(6)=FDIM*C(10)*C(12)/(C(14)*C(16))
000254          ANGLE=T(1,J)+D(4)*(P(2,1)-P(1,J))+D(6)*(Y(2,1)-Y(1,J))
000270       14 RETURN
000271          END



                SUBROUTINE BOUND(J)
          C
          C       CENTERLINE OR SOLID BOUNDARY CALCULATION
          C
000003          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1         PHS(100),ASASS(100),DASDX(100),ISLP
000003          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000003          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1         XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DHDY,DSHD,END,
               2         PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3         SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000003          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003          COMMON/BNLYR/XSUM(100),YSUM(100),DELSHD(100),THETAS(100),
               1        CFSHD(100),XCNE(100),YCNE(100),DELCNE(100),THETAC(100),
               2        CFCNE(100),REYPRM,POP,AOP,VOP,REYSEC,POS,AOS,VOS,PEX,XSCALF
000003          COMMON/CPLOT/IPLOT,XSTART,YSTART,XSPAN,YSPAN,SCALE,SPAN,AXIS,
               1      XORGN,YORGN,XSHFT,YSHFT,KKK(14),PP(14),XDOWN(100),YACROS(100)
000003          DIMENSION C(16),D(6)
000003          AVE(X1,X2)=(X1+X2)/2.0
000012      500 FORMAT (1H1,//40X,47HUNABLE TO OBTAIN CONVERGENCE IN SUBROUTINE LI
               1NE////)
000012      502 FORMAT (6X,6HITER =I3,4X,8HX(2,J) =F8.5,4X,8HY(2,J) =F8.5,4X,
               18HP(2,J) =F8.5,4X,8HT(2,J) =F8.5,4X,6HTEST =1PE12.5)
000012      504 FORMAT (19X,7F12.5)
000012      506 FORMAT (1HJ)
          C
          C       SUBROUTINE BOUND
          C
```

```
000012          ITER=0
000013          ERROR=0.0001
000015          X(2,J)=X(1,J)
000021          P(2,J)=P(1,J)
000023          IF (ICONE .EQ. 1 .AND. NBDY .GT. 0) CALL CNLYR(DDSDX)
000034          ICONE=ICONE+1
000036          DELCNE(ICONE)=DELCNE(ICONE-1)
000040          IF (NBDY .GT. 0) GO TO 10
000042          DDSDX=0.0
000042          DELCNE(ICONE)=0.0
000043       10 ITER=1+ITER
000045          PREF=P(2,J)
000047          CALL FIND(X(2,J),YCNE(ICONE),DYCDX,1.0)
000053          TCNE=ABS(ATAN(DYCDX))
000057          DYPDX=DYCDX+DDSDX
000061          Y(2,J)=YCNE(ICONE)+DELCNE(ICONE)/COS(TCNE)
000072          T(2,J)=ATAN(DYPDX)
000076          CALL COAVE(XAVE,YAVE,PAVE,TAVE,1,J,2,J)
000106          CALL COEFF(XAVE,YAVE,PAVE,TAVE,C,GAMP)
000112          D(2)=C(2)
000114          D(4)=1.0/(C(4)*C(6)*C(8))
000117          D(6)=FDIM*C(10)*C(12)/(C(14)*C(16))
000124          X(2,J)=(Y(1,J)-Y(2,J)-D(2)*X(1,J)+DYPDX*X(2,J))/(DYPDX-D(2))
000140          P(2,J)=(D(4)*P(1,J)+T(2,J)-T(1,J)-D(6)*(Y(2,J)-Y(1,J)))/D(4)
000154          XCONE(ICONE)=X(2,J)-DELCNE(ICONE)*SIN(TCNE)
000163          CALL FIND(XCONE(ICONE),YCONE(ICONE),DYCDX,1.0)
000167          PCONE(ICONE)=P(2,J)
000173          TCONE(ICONE)=ATAN(DYCDX)
000177          IF (NBDY .GT. 0 .AND. REYPRM .GT. 0.0) CALL CNLYR(DDSDX)
000212       12 TEST=ABS(P(2,J)-PREF)
000216          IF (ITER .LT. 45) GO TO 14
000220          IF (ITER .EQ. 45) WRITE (6,600)
000226          WRITE (6,602) ITER,X(2,J),Y(2,J),P(2,J),T(2,J),TEST
000256          IF (ITER .LT. 45) GO TO 14
000262          CALL OUTSLP
000263          IF (CHOKE .EQ. -1.0) CALL OUTLYR
000270          IF (IPLOT .EQ. 1) CALL PLOTL
000274          CALL EXIT
000275       14 IF (TEST .GT. ERROR*P(2,J)) GO TO 10
000304       16 RETURN
000305          END
```

```
                SUBROUTINE FIELD(J)
          C
          C     FIELD POINT CALCULATION
          C
000003          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1       PHS(100),ASASS(100),DASDX(100),ISLP
000003          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000003          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000003          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003          DIMENSION A(16),B(16),D(6)
000003          AVE(X1,X2)=(X1+X2)/2.0
000012      600 FORMAT (1H1,//40X,48HUNABLE TO OBTAIN CONVERGENCE IN SUBROUTINE FI
               1ELD////)
000012      602 FORMAT (8X,6HITER =I3,4X,8HX(2,J) =F8.5,4X,8HY(2,J) =F8.5,4X,
               18HP(2,J) =F8.5,4X,8HT(2,J) =F8.5,4X,6HTEST =1PE12.5)
          C
```

```
          C       SUBROUTINE FIELD
          C
000012            K=J-1
000014            ITER=0
000015            ERROR=0.0001
000016            X(2,J)=AVE(X(1,J),X(2,K))
000025            Y(2,J)=AVE(Y(1,J),Y(2,K))
000032            P(2,J)=AVE(P(1,J),P(2,K))
000037            T(2,J)=AVE(T(1,J),T(2,K))
000044         10 ITER=1+ITER
000046            PREF=P(2,J)
000050            CALL COAVE(XAVE,YAVE,PAVE,TAVE,2,K,2,J)
000060            CALL COEFF(XAVE,YAVE,PAVE,TAVE,A,GAMP)
000064            CALL COAVE(XAVE,YAVE,PAVE,TAVE,1,J,2,J)
000075            CALL COEFF(XAVE,YAVE,PAVE,TAVE,B,GAMP)
000101            D(1)=A(1)
000103            D(2)=B(2)
000104            D(3)=1.0/(A(3)*A(5)*A(7))
000110            D(4)=1.0/(B(4)*B(6)*B(8))
000113            D(5)=FDIM*A(9)*A(11)/(A(13)*A(15))
000117            D(6)=FDIM*B(10)*B(12)/(B(14)*B(16))
000124            X(2,J)=(Y(1,J)-Y(2,K)+D(1)*X(2,K)-D(2)*X(1,J))/(D(1)-D(2))
000143            Y(2,J)=Y(1,J)+D(2)*(X(2,J)-X(1,J))
000152            P(2,J)=(D(3)*P(2,K)+D(4)*P(1,J)+T(2,K)-T(1,J)-D(5)*(Y(2,J)-Y(2,K))
                 1       -D(6)*(Y(2,J)-Y(1,J)))/(D(3)+D(4))
000202            T(2,J)=T(1,J)+D(4)*(P(2,J)-P(1,J))+D(6)*(Y(2,J)-Y(1,J))
000214            TEST=ABS(PREF-P(2,J))
000220            IF (ITER .LT. 95) GO TO 12
000222            IF (ITER .EQ. 95) WRITE (6,600)
000230            WRITE (6,602) ITER,X(2,J),Y(2,J),P(2,J),T(2,J),TEST
000260            IF (ITER .EQ. 100) CALL EXIT
000265         12 IF (TEST .GT. ERROR*P(2,J)) GO TO 10
000273            RETURN
000273            END
```

```
                  SUBROUTINE SLIP(J)
          C
          C       SLIPLINE CALCULATION
          C
000003            COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003            COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
                 1       PHS(100),ASASS(100),DASDX(100),ISLP
000003            COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003            COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003            COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003            COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003            COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000003            COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
                 1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
                 2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
                 3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000003            COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003            DIMENSION C(16),D(6)
000003            FUNP(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-G/(G-1.0))
000020            AVE(X1,X2)=(X1+X2)/2.0
000026        600 FORMAT (1H1,//40X,47HUNABLE TO OBTAIN CONVERGENCE IN SUBROUTINE SL
                 1IP////)
000026        602 FORMAT (8X,6HITER =I3,4X,8HX(2,J) =F8.5,4X,8HY(2,J) =F8.5,4X,
                 18HP(2,J) =F8.5,4X,8HT(2,J) =F8.5,4X,6HTEST =1PE12.5)
000026        604 FORMAT (1HJ)
          C
          C       SUBROUTINE SLIP
          C
000026            K=J-1
000030            ITER=0
000031            ERROR=0.0001
```

```
000032          CALL COEFF(X(2,K),Y(2,K),P(2,K),T(2,K),C,GAMP)
000042          DYDX=TAN(T(1,K))
000046       10 X(2,J)=(Y(2,K)-Y(1,K)+DYDX*X(1,K)-C(1)*X(2,K))/(DYDX-C(1))
000064          Y(2,J)=Y(2,K)+C(1)*(X(2,J)-X(2,K))
000073          P(2,J)=AVE(P(1,K),P(2,K))
000101          T(2,J)=AVE(T(1,K),T(2,K))
000107          IF (P(2,J) .GE. HSHP) P(2,J)=PHP(ISLP-1)
000114          CALL STORE(J)
000115       12 ITER=1+ITER
000117          YREF=Y(2,J)
000122          PREF=P(2,J)
000124       14 CALL COAVE(XAVE,YAVE,PAVE,TAVE,2,J,2,K)
000134          CALL COEFF(XAVE,YAVE,PAVE,TAVE,C,GAMP)
000140          D(1)=C(1)
000142          D(3)=1.0/(C(3)*C(5)*C(7))
000145          D(5)=FDIM*C(9)*C(11)/(C(13)*C(15))
000152          IF (CHANGE .EQ. 1.0) GO TO 16
000155          X(2,J)=(Y(2,K)-Y(1,K)+DYDX*X(1,K)-D(1)*X(2,K))/(DYDX-D(1))
000172          Y(2,J)=Y(2,K)+D(1)*(X(2,J)-X(2,K))
000201          CALL FLOW(J)
000202          IF (POINT .EQ. -1.0) GO TO 24
000205          IF (CHANGE .EQ. 1.0) GO TO 18
000207          T(2,J)=T(2,K)-D(3)*(P(2,J)-P(2,K))-D(5)*(Y(2,J)-Y(2,K))
000225          DYDX=AVE(TAN(T(1,K)),TAN(T(2,J)))
000240          CALL STORE(J)
000241          GO TO 20
000243       16 CALL FLOW(J)
000244          IF (POINT .EQ. -1.0) GO TO 24
000247       18 Y(2,J)=YSLP(ISLP)
000252          IF (ABS(YREF-Y(2,J)) .LE. ERROR*Y(2,J)) Y(2,J)=YREF
000262          X(2,J)=X(2,K)+(Y(2,J)-Y(2,K))/D(1)
000273          DYDX=(Y(2,J)-Y(1,K))/(X(2,J)-X(1,K))
000302          DZDX=TAN(T(1,K))
000305          T(2,J)=ATAN(AVE(DYDX,DZDX))
000314          P(2,J)=(D(3)*P(2,K)-(T(2,J)-T(2,K))-D(5)*(Y(2,J)-Y(2,K)))/D(3)
000330          PEST=P(2,J)/HSHP
000333          IF (PEST .GT. 0.0) PEST=PEST**(-(GAMS-1.0)/GAMS)-1.0
000343          IF (PEST .GT. 0.0) CALL STORE(J)
000347          IF (PEST .GT. 0.0) GO TO 20
000352          POINT=-1.0
000353          ASASS(ISLP)=0.5
000355          GO TO 24
000355       20 TESTP=ABS(P(2,J)-PREF)
000361          TESTY=ABS(Y(2,J)-YREF)
000364          IF (ITER .LT. 95) GO TO 22
000367          IF (ITER .EQ. 95) WRITE (6,600)
000374          IF (WTFL .EQ. 0.0) WRITE (6,602) ITER,X(2,J),Y(2,J),P(2,J),
               1                                 T(2,J),TESTY
000426          IF (WTFL .GT. 0.0) WRITE (6,602) ITER,X(2,J),Y(2,J),P(2,J),
               1                                 T(2,J),TESTP
000460          IF (ITER .EQ. 100) GO TO 23
000462       22 IF (ITER .LT. 1) GO TO 12
000465          IF (TESTP .GT. ERROR*P(2,J)) GO TO 12
000472          IF (TESTY .GT. ERROR*Y(2,J)) GO TO 12
000476       23 IF (STAG .GE. 0.0) GO TO 24
000500          CALL FIND(XSLP(ISLP),YP,DYPDX,2.0)
000504          DELTA=YP-YSLP(ISLP)
000507          IF (DELTA .LE. 0.0) POINT=-1.0
000513       24 RETURN
000514          END


                SUBROUTINE SLID(J)
          C
          C     CALCULATION OF WAKE EFFECT ON FLOW FIELD
          C
000003          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
```

```
                1         PHS(100),ASASS(100),DASDX(100),ISLP
000003            COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003            COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003            COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003            COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003            COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000003            COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
                1         XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
                2         PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
                3         SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000003            COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003            FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000024            FUNP(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-G/(G-1.0))
000040            AVE(X1,X2)=(X1+X2)/2.0
000046        600 FORMAT (1H1,//43X,47HUNABLE TO OBTAIN CONVERGENCE IN SUBROUTINE SL
               1IP////)
000046        602 FORMAT (22X,6HITER =I6,4X,7HASASS =F8.5,4X,6HWSEC =F8.5,4X,
                1         6HDELW =F8.5,4X,6HTEST =1PE12.5)
000046        604 FORMAT (1H1,//28X,12A6,///)
000046        606 FORMAT (22X,6HISLP =I6,4X,6HXSLP =F8.5,4X,6HWSEC =F8.5,4X,
                1         6HWMIX =F8.5,4X,8HASSAPS =F8.5)
000046        608 FORMAT (1H0)
000046            ITER=0
000047            ERROR=0.0001
000051            IF (ISLP .EQ. 2) NPOINT=0
000054            IF (SOLVE .EQ. 0.0 .OR. STAG .EQ. -1.0) GO TO 18
000063            IF (CHARGE .EQ. 1.0 .OR. CHANGE .EQ. 1.0) GO TO 18
000072            AMSAVE=AMP(ISLP)
000074            PSAVE=PHP(ISLP)
000075            TSAVE=THETA(ISLP)
000077            AMS(ISLP)=AMS(ISLP-1)
000100            CALL AJAX(XP,YP,ALPHA,ASEC,DADX)
000104         10 ITER=1+ITER
000106            ASAVE=ASASS(ISLP)
000110            WSAVE=WSEC
000111            IF (ITER .EQ. 2) ASASS(ISLP)=WSEC/WTFL*ASASS(ISLP)
000116            IF (ITER .GT. 2) ASASS(ISLP)=ASASS(ISLP)+DADW*(WTFL-WSEC)
000125            IF (ASASS(ISLP) .LT. 1.0) GO TO 16
000130            CALL ASTAR(AMS(ISLP),ASASS(ISLP),GAMS)
000133            PHS(ISLP)=FUNP(GAMS,AMS(ISLP))
000140            PHP(ISLP)=PHS(ISLP)*HSHP
000142            AMP(ISLP)=FUNM(GAMP,PHP(ISLP))
000150            CALL WAKE(DELW,DELA)
000152            ASPREF=(ASEC-DELA)/APRIM*APREF/ASASS(ISLP)
000157            WS=1.0/FUNG*HSHP*ASPREF
000163            WSEC=WS+DELW
000165            WRATIO=WS/WTFL
000167            WMIX=DELW/WTFL
000170            TEST=ABS(WSEC-WTFL)
000172            IF (DELA/ASEC .GE. 0.50) GO TO 16
000177            IF (ITER .GT. 1) DADW=(ASAVE-ASASS(ISLP))/(WSAVE-WSEC)
000206            IF (ITER .LT. 45) GO TO 12
000211            IF (ITER .EQ. 45) WRITE (6,600)
000216            WRITE (6,602) ITER,ASASS(ISLP),WSEC,DELW,TEST
000234            IF (ITER .EQ. 50) CALL EXIT
000241         12 IF (TEST .GT. 0.10*ERROR) GO TO 10
000246            ASAVE=ASASS(ISLP)
000250            IF (AREA(1) .LT. 1.20) ASAVE=AREA(NITER)
000254            IF (ASPREF .GT. ASSAPS) GO TO 16
000260            IF (ASAVE .GT. AREA(NITER) .AND. AREA(NITER) .LT. 1.20) GO TO 16
000271            ASSAPS=ASPREF
000272            P(2,J)=PHP(ISLP)
000275            T(2,J)=PMER(AMSAVE,TSAVE,AMP(ISLP),GAMP)
000303         14 CALL STORE(J)
000304            GO TO 18
000306         16 CHARGE=1.0
000310            CALL SLIP(J)
000311         18 IF (POINT .NE. -1.0) GO TO 22
000314            IF (STAG .EQ. 1.0) GO TO 20
000316            IF (NPOINT .LE. 1) GO TO 24
000321         20 STAG=2.0
000323            POINT=1.0
```

```
000324          ISLP=ISLP-1
000326          HSHP=PTS(NITER)
000330          GO TO 24
000330       22 IF (ASASS(ISLP) .LE. 1.05) NPOINT=1+NPOINT
000335          CALL AJAX(XP,YP,ALPHA,ASEC,DASDX(ISLP))
000342          DADX=-1.0
000344          POINT=0.0
000345          DELA=0.0
000346          IF (WTFL .EQ. 0.0) GO TO 21
000350          IF (ISLP .LT. 3) GO TO 24
000352          DELP=ASASS(ISLP)-ASASS(ISLP-1)
000354          DELQ=ASASS(ISLP-1)-ASASS(ISLP-2)
000356          DELA=-ABS(DELP)
000360          IF (DELP .GT. 0.0 .AND. DELQ .GT. 0.0) DELA=DELP
000371       21 IF (DASDX(ISLP) .GT. 0.0 .AND. DASDX(ISLP-1) .GT. 0.0) DADX=1.0
000403          IF (DELA .GE. 0.0 .AND. DADX .GT. 0.0) POINT=1.0
000414          IF (TRY .EQ. 1.0 .AND. ISLP .LE. NDATA) POINT=0.0
000427          IF (XSLP(ISLP) .LT. END .AND. AREA(NITER) .LE. 1.05) POINT=0.0
000443          IF (XSLP(ISLP) .GE. END) POINT=1.0
000450          IF (SOLVE .LE. 1.0 .OR. STAG .EQ. 1.0) GO TO 24
000461          IF (DELA .GT. 0.0 .AND. DADX .GT. 0.0) POINT=1.0
000471          IF (TRY .EQ. 1.0 .AND. ISLP .LE. NDATA) POINT=0.0
000504       24 RETURN
000505          END
```

```
                SUBROUTINE FLOW(J)
          C
          C     CALCULATION OF FLOW CONDITIONS ALONG THE SLIPLINE
          C
000003          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1         PHS(100),ASASS(100),DASDX(100),ISLP
000003          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000003          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1         XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2         PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3         SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000003          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003          FUNA(G,AM)=((G+1.0)/2.0)**(-(G+1.0)/(2.0*(G-1.0)))*1.0/AM*(1.0+
               1              (G-1.0)/2.0*AM*AM)**((G+1.0)/(2.0*(G-1.0)))
000034          FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000054          FUNP(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-G/(G-1.0))
000070          AVE(X1,X2)=(X1+X2)/2.0
000076      600 FORMAT (1H1,//40X,47HUNABLE TO OBTAIN CONVERGENCE IN SUBROUTINE FL
               1OW////)
000076      602 FORMAT (1HJ)
000076          I=ISLP
000100          ITER=0
000101          ATOL=1.05
000102          ERROR=0.0001
000104          ASASS(I)=1.0
000106          IF (ISLP .EQ. 1) AMIN=10.0
000111          IF (STAG .GE. 0.0) GO TO 10
000113          P(2,J)=HSHP
000115          PHP(I)=HSHP
000117          AMP(I)=FUNM(GAMP,PHP(I))
000124          AMS(I)=0.0
000125          PHS(I)=1.0
000127          ASASS(I)=500.0
000130          CALL SHLYR(DDSDX)
000132          IF (I .EQ. 1) THETA(I)=PMER(AMR,ANGR,AMP(I),GAMP)
000144          GO TO 20
000145       10 ITER=1+ITER
```

```
000147          ASAVE=ASASS(I)
000151          IF (I .EQ. 1) THETA(I)=PMER(AMR,ANGR,AMP(I),GAMP)
000160          IF (CHANGE .EQ. 1.0) GO TO 12
000162          CALL AJAX(XP,YP,ALPHA,ASEC,DADX)
000166          ASASS(I)=ASEC/APRIM*APREF/ASSAPS
000173          IF (I .EQ. 1) GO TO 14
000175          IF (ASASS(I) .GT. 1.05 .AND. ASASS(I-1) .GT. 1.07) GO TO 14
000206          AMS(I)=AMS(I-1)
000210       12 CHANGE=1.0
000212          CALL AJAX(XP,YP,ALPHA,ASEC,DADX)
000215          ASASS(I)=AVE(ASASS(I-1),FUNA(GAMS,AMS(I)))
000225          ASEC=ASASS(I)*APRIM/APREF*ASSAPS
000231          YSLP(I)=YP*YP**FDIM-ASEC*COS(ALPHA)
000242          IF (YSLP(I) .LE. 0.0) ASASS(I)=0.5
000247          IF (YSLP(I) .GT. 0.0 .AND. FDIM .EQ. 1.0) YSLP(I)=SQRT(YSLP(I))
000264          Y(2,J)=YSLP(I)
000267          DZDX=TAN(T(1,J-1))
000274          DYDX=(Y(2,J)-Y(1,J-1))/(X(2,J)-X(1,J-1))
000302          THETA(I)=ATAN(AVE(DYDX,DZDX))
000310          IF (ASASS(I) .GT. 1.0) GO TO 16
000315       14 IF (ASASS(I) .LT. 1.0) GO TO 20
000320          CALL ASTAR(AMS(I),ASASS(I),GAMS)
000323          PHS(I)=FUNP(GAMS,AMS(I))
000330          PHP(I)=PHS(I)*HSHP
000332          AMP(I)=FUNM(GAMP,PHP(I))
000340          P(2,J)=PHP(I)
000343          IF (I .GT. 1) GO TO 20
000346       16 TEST=ABS(ASASS(I)-ASAVE)
000352          IF (ITER .LT. 25) GO TO 18
000354          WRITE (6,600)
000360          CALL EXIT
000361       18 IF (TEST .GT. ERROR*ASASS(I)) GO TO 10
000367       20 POINT=0.0
000370          IF (ASASS(I) .LT. 1.0) POINT=-1.0
000375          IF (ISLP .GT. 1 .OR. POINT .EQ. 0.0) GO TO 22
000406          HSHP=1.50*HSHP/ASASS(I)
000411          ASASS(I)=0.50
000412          IF (NITER .EQ. 50) CALL EXIT
000416       22 RETURN
000417          END
```

```
                SUBROUTINE SHLYR(DDSDX)
        C
        C       BOUNDARY LAYER ALONG SHROUD WALL
        C
000003          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
              1        PHS(100),ASASS(100),DASDX(100),ISLP
000003          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000003          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
              1        XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
              2        PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
              3        SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000003          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003          COMMON/BNLYR/XSUM(100),YSUM(100),DELSHD(100),THETAS(100),
              1        CFSHD(100),XCNE(100),YCNE(100),DELCNE(100),THETAC(100),
              2        CFCNE(100),REYPRM,POP,AOP,VOP,REYSEC,POS,AOS,VOS,PEX,XSCALF
000003      600 FORMAT (1H1,//28X,18A4,//)
000003      602 FORMAT (22X,6HWTFL =F9.6,5X,9HPTS/PTS =F9.6,5X,8HREYPRM =1PE10.3,
              1        5X,8HREYSEC =1PE10.3)
000003      604 FORMAT (//26X,4HXSHD,8X,4HYSHD,8X,3HAMS,8X,6HDELSHD,6X,6HTHETAS,
              1        6X,5HCFSHD,6X,5HDDSDX//)
000003      606 FORMAT (19X,7F12.5)
```

```
000003        610 FORMAT (1HJ)
000003            AVE(X1,X2)=(X1+X2)/2.0
000012            PR(AM)=(AM/(1.0+0.2*AM*AM))**4
000020            FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000040            FUNT(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-1.0)
000052            FUNP(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-G/(G-1.0))
000066            FUNU(TEMP)=2.27*32.17*TEMP**1.5/(198.6+TEMP)*1.E-8
          C
          C       SUBROUTINE LAYER
          C
000101            DDSDX=0.0
000102            IF (REYPRM .EQ. 0.0) GO TO 30
000103            I=ISLP
000104            S=0.702
000106            BETA=0.0
000107            START=0.0
000110            TWTO=0.950
000111            IF (FDIM .EQ. 1.0) BETA=1.25
000115            IF (WTFL .GT. 0.0 .AND. ISLP .EQ. 1) GO TO 12
000126            IF (WTFL .GT. 0.0 .AND. ISLP .GT. 1) GO TO 14
000137            IF (WTFL .EQ. 0.0 .AND. AMS(I-1) .GT. 0.0) GO TO 14
000147            IF (ISLP .EQ. 1) GO TO 12
000151         10 XSUM(I)=0.0
000153            DELSHD(I)=0.0
000154            THETAS(I)=0.0
000155            CFSHD(I)=0.0
000156            REYSEC=0.0
000157         12 TP=TOP*FUNT(GAMP,AMR)
000163            AP=49.02*SQRT(TP)
000166            VP=2.0*AP/REYPRM*XSCALE*AMR
000172            PP=53.3*TP*FUNU(TP)/VP
000200            POP=PP/FUNP(GAMP,AMR)
000203            POS=POP*PTS(NITER)
000205            AOP=49.02*SQRT(TOP)
000210            AOS=49.02*SQRT(TOS)
000213            VOP=53.3*TOP*FUNU(TOP)/POP
000221            VOS=53.3*TOS*FUNU(TOS)/POS
000226         14 CALL FIND(XSLP(I),YSUM(I),DYDX,2.0)
000232            IF (I .GE. 1 .AND. AMS(I) .EQ. 0.0) GO TO 30
000245            IF (I .GT. 1 .AND. AMS(I-1) .EQ. 0.0) GO TO 30
000256            IF (DELSHD(I) .EQ. 0.0) START=1.0
000261            AMSHD=AMS(I)
000263            IF (I .GT. 1) AMSHD=AVE(AMS(I),AMS(I-1))
000270            PS=POS*FUNP(GAMS,AMSHD)
000274            TS=TOS*FUNT(GAMS,AMSHD)
000277            AS=49.02*SQRT(TS)
000302            VS=53.3*TS*FUNU(TS)/PS
000310            TSTO=1.0/(1.0+0.20*AMSHD**2)
000314            TAVETO=0.50*TWTO+0.22*S**(1.0/3.0)+(0.50-0.22*S**(1.0/3.0))*TSTO
000331            TSTAVE=TSTO/TAVETO
000332            TAVE=TS/TSTAVE
000334            VAVE=53.3*TS*FUNU(TAVE)/PS
000341            IF (ISLP .GT. 1) GO TO 16
000344            REYSEC=2.0*AS/VS*XSCALE*AMS(I)
000350            XREX=XSCALE*DELSHD(I)/(0.046*(1.0+0.80*AMS(I)**2)**(0.44))
000362            REX=(AOS/VOS*AMS(I)*(1.0+0.20*AMS(I)**2)**(-2.25))**(-0.20)
000375            XDIM=(XREX/REX)**(5.0/4.0)
000404            XSUM(I)=XDIM*YSUM(I)**BETA*PR(AMS(I))
000415            GO TO 18
000415         16 YAVE=AVE(YSUM(I),YSUM(I-1))
000420            AMAVE=AVE(AMS(I),AMS(I-1))
000423            DELX=XSLP(I)-XSLP(I-1)
000425            XSUM(I)=XSUM(I-1)+YAVE**BETA*PR(AMAVE)*XSCALE*DELX
000437         18 XDIM=1.0/(YSUM(I)**BETA*PR(AMSHD))*XSUM(I)
000450            REX=AOS/VOS*XDIM*AMSHD*(1.0+0.20*AMSHD**2)**(-2.25)
000460            IF (REX .LE. 0.0) GO TO 22
000462            DELSHD(I)=0.046*XDIM/XSCALE*(1.0+0.80*AMSHD**2)**(0.44)
                 1                        *REX**(-1.0/5.0)
000500            THETAS(I)=0.036*XDIM/XSCALE*(1.0+0.10*AMSHD**2)**(-0.70)
                 1                        *REX**(-1.0/5.0)
000517            DDSDX=0.0368*(1.0+0.80*AMSHD**2)**(0.44)*REX**(-1.0/5.0)
000534            REY=AS/VAVE*THETAS(I)*XSCALE*AMSHD
```

```
000541          HC=DELSHD(I)/THETAS(I)
000543          HI=(HC/TWTO-0.20*AMSHD**2)*TSTO
000550       20 CFSHD(I)=0.246*EXP(-1.561*HI)*REY**(-0.268)*(TSTAVE)**1.268
000566          IF (ISLP .EQ. 1) GO TO 30
000571          IF (DELSHD(I-1) .EQ. 0.0) GO TO 30
000573          IF (START .EQ. 1.0) DELSHD(I)=DELSHD(I-1)
000576          IF (DELSHD(I) .GT. 5.0*DELSHD(I-1)) DELSHD(I)=DELSHD(I-1)
000604          GO TO 30
000605       22 ISLP=ISLP-1
000607          CALL OUTLYR
000610          CALL EXIT
000611       30 RETURN
000612          END
```

```
                SUBROUTINE CNLYR(DDSDX)
          C
          C     BOUNDARY LAYER ALONG CENTERBODY SURFACE
          C
000003          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1        PHS(100),ASASS(100),DASDX(100),ISLP
000003          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000003          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1        XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2        PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3        SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000003          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003          COMMON/BNLYR/XSUM(100),YSUM(100),DELSHD(100),THETAS(100),
               1        CFSHD(100),XCNE(100),YCNE(100),DELCNE(100),THETAC(100),
               2        CFCNE(100),REYPRM,POP,AOP,VOP,REYSEC,POS,AOS,VOS,PEX,XSCALE
000003      600 FORMAT (1H1,//28X,18A4,//)
000003      602 FORMAT (//32X,4HXCNE,8X,4HYCNE,8X,3HAMC,8X,6HDELCNE,6X,6HTHETAC,
               1        6X,5HCFCNE,7X,5HDDSDX//)
000003      604 FORMAT (1H0)
000003      606 FORMAT (25X,7F12.5)
000003          AVE(X1,X2)=(X1+X2)/2.0
000012          PR(AM)=(AM/(1.0+0.2*AM*AM))**4
000020          FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000040          FUNT(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-1.0)
000052          FUNP(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-G/(G-1.0))
000066          FUNU(TEMP)=2.27*32.17*TEMP**1.5/(198.6+TEMP)*1.E-8
000101          IF (REYPRM .EQ. 0.0) GO TO 20
000102          S=0.702
000104          I=ICONE
000105          BETA=0.0
000106          TWTO=0.950
000107          IF (FDIM .EQ. 1.0) BETA=1.25
000113          AMCNE=FUNM(GAMP,PCONE(I))
000120          IF (I .GT. 1) AMCNE=FUNM(GAMP,AVE(PCONE(I),PCONE(I-1)))
000131          PP=POP*FUNP(GAMP,AMCNE)
000135          TP=TOP*FUNT(GAMP,AMCNE)
000140          AP=49.02*SQRT(TP)
000143          TPTO=FUNT(GAMP,AMCNE)
000146          TAVETO=0.50*TWTO+0.22*S**(1.0/3.0)+(0.50-0.22*S**(1.0/3.0))*TPTO
000163          TPTAVE=TPTO/TAVETO
000164          TAVE=TP/TPTAVE
000166          VAVE=53.3*TP*FUNU(TAVE)/PP
000173          IF (ICONE .GT. 1) GO TO 10
000176          DELCNE(1)=DELSHD(1)
000177          XREX=XSCALE*DELCNE(I)/(0.046*(1.0+0.80*AMCNE**2)**(0.44))
000210          REX=(AOP/VOP*AMCNE*(1.0+0.20*AMCNE**2)**(-2.25))**(-0.20)
000223          XDIM=(XREX/REX)**(5.0/4.0)
000232          XCNE(1)=XDIM*YCONE(I)**BETA*PR(AMCNE)
```

```
000242          GO TO 12
000242       10 YAVE=AVE(YCONE(I),YCONE(I-1))
000245          PAVE=AVE(PCONE(I),PCONE(I-1))
000250          AMAVE=FUNM(GAMP,PAVE)
000254          DELX=XCONE(I)-XCONE(I-1)
000256          XCNE(I)=XCNE(I-1)+YAVE**BETA*PR(AMAVE)*XSCALE*DELX
000270       12 XDIM=1.0/(YCONE(ICONE)**BETA*PR(AMCNE))*XCNE(I)
000301          REX=AOP/VOP*XDIM*AMCNE*(1.0+0.20*AMCNE**2)**(-2.25)
000311          IF (REX .LE. 0.0) GO TO 18
000313          IF (REX .GE. 1.E7) GO TO 14
000315          DELCNE(I)=0.046*XDIM/XSCALE*(1.0+0.80*AMCNE**2)**(0.44)
               1                         *REX**(-1.0/5.0)
000334          THETAC(I)=0.036*XDIM/XSCALE*(1.0+0.10*AMCNE**2)**(-0.70)
               1                         *REX**(-1.0/5.0)
000353          DDSDX=0.0368*(1.0+0.80*AMCNE**2)**(0.44)*REX**(-1.0/5.0)
000370          GO TO 16
000370       14 DELCNE(I)=0.028*XDIM/XSCALE*(1.0+0.8*AMCNE**2)**(0.44)
               1                         *REX**(-1.0/6.0)
000407          THETAC(I)=0.022*XDIM/XSCALE*(1.0+0.10*AMCNE**2)**(-0.70)
               1                         *REX**(-1.0/6.0)
000426          DDSDX=0.0233*(1.0+0.80*AMCNE**2)**(0.44)*REX**(-1.0/6.0)
000443       16 REY=AP/VAVE*THETAC(I)*XSCALE*AMCNE
000450          HC=DELCNE(I)/THETAC(I)
000453          HI=(HC/TWTO-0.20*AMCNE**2)*TPTO
000457          CFCNE(I)=0.246*EXP(-1.561*HI)*REY**(-0.268)*(TPTAVE)**1.268
000474          GO TO 20
000476       18 ICONE=ICONE-1
000500          CALL OUTLYR
000501          CALL EXIT
000502       20 RETURN
000503          END
```

```
                SUBROUTINE WAKE(DELW,DELA)
          C
          C     CALCULATION OF VISCOUS MIXING REGION
          C
000005          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000005          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1        PHS(100),ASASS(100),DASDX(100),ISLP
000005          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000005          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000005          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000005          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000005          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000005          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1        XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2        PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3        SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000005          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000005          FUNA(G,AM)=((G+1.0)/2.0)**(-(G+1.0)/(2.0*(G-1.0)))*1.0/AM*(1.0+
               1            (G-1.0)/2.0*AM*AM)**((G+1.0)/(2.0*(G-1.0)))
000036          FUNQ(G,AM)=SQRT((G+1.0)/2.0*AM*AM/(1.0+(G-1.0)/2.0*AM*AM))
000057          CROCCO(G,AM)=AM*AM/(2.0/(G-1.0)+AM*AM)
000067          FUNCP(CA,PHI)=(1.0-CA)*PHI/(1.0-CA*PHI*PHI)
000077          AVE(X1,X2)=(X1+X2)/2.0
000105          PHIB=FUNQ(GAMS,AMS(ISLP))/FUNQ(GAMP,AMP(ISLP))*SQRT(TOS/TOP)
000127          GAMI=(GAMP+WTFL*GAMS)/(1.0+WTFL)
000135          SIISI=(1.0+PHIB)/(1.0-PHIB)
000141          CAIISQ=CROCCO(GAMP,AMP(ISLP))
000144          CAISQ=CAIISQ*(1.0-PHIB)**2/(CAIISQ*(1.0-PHIB)**2+(1.0-CAIISQ))
000155          SIGI=12.0+2.758*SQRT(CAISQ)/(SQRT((1.0-CAISQ)*(GAMI-1.0)/2.0))
000171          SIGII=SIISI*SIGI
000173          ARATIO=FUNA(GAMP,AMR)/FUNA(GAMP,AMP(ISLP))
000202          CALL MIX(TOS/TOP,CAIISQ,PHIB,0.0,SIGVB)
000207          DELA=2.0/SIGII*XSLP(ISLP)*YSLP(ISLP)*SIGVB
000215          DELW=2.0/SIGII*XSLP(ISLP)*YSLP(ISLP)*ARATIO*FUNCP(CAISQ,PHIB)*
               1      SIGVB
000225          RETURN
000226          END
```

```
                SUBROUTINE MIX(TOB,CA2,PHIB,PHID,SIGVB)
          C
          C     CALCULATION OF PERTINENT PARAMETERS FOR MIXING SOLUTION
          C
000010          DIMENSION ETA(410),PHI(410),TOR(410),EI1(410),EI2(410),EI3(410)
000010          PI=SQRT(3.1415927)
000011          ETARB=-5.0
000013          DEETA=0.020
000014          ETA(1)=ETARB
000016          PHI(1)=PHIB
000022          TOR(1)=TOB
000023          A1=PHIB*(1.0-CA2)/(TOB-CA2*PHIB**2)
000030          B1=PHIB*A1
000031          C1=TOB*A1
000033          EI1(1)=A1*ETARB
000034          EI2(1)=B1*ETARB
000035          EI3(1)=C1*ETARB
000037          DO 12 I=1,400
000041          ETA(I+1)=ETA(I)+DEETA
000043          AVETA=(ETA(I+1)+ETA(I))/2.0
000046          PHI(I+1)=PHI(I)+(1.0-PHIB)*EXP(-(AVETA**2))*DEETA/PI
000064          TOR(I+1)=(TOB*(1.0-PHI(I+1))+PHI(I+1)-PHIB)/(1.0-PHIB)
000074          A2=PHI(I+1)*(1.0-CA2)/(TOR(I+1)-CA2*PHI(I+1)**2)
000102          B2=PHI(I+1)*A2
000105          C2=TOR(I+1)*A2
000106          EI1(I+1)=EI1(I)+(A1+A2)*DEETA/2.0
000114          EI2(I+1)=EI2(I)+(B1+B2)*DEETA/2.0
000121          EI3(I+1)=EI3(I)+(C1+C2)*DEETA/2.0
000127          A1=A2
000130          B1=B2
000132          C1=C2
000133          J=I+1
000135          IF (ETA(I+1)) 12,12,10
000137       10 A=EI1(I+1)-EI2(I+1)
000142          B=EI1(I)-EI2(I)
000144          IF (ABS(A-B)-1.0E-05) 14,14,12
000151       12 CONTINUE
000153       14 EI1J=(EI1(J)-EI2(J))/(1.0-PHIB)
000157          ETAM=ETA(J)-(EI2(J)-PHIB*EI1(J))/(1.0-PHIB)
000167          DO 16 K=2,400
000170          IF (EI1(K)-EI1J) 16,18,18
000173       16 CONTINUE
000175       18 DETA=(EI1J-EI1(K-1))/(EI1(K)-EI1(K-1))*DEETA
000202          ETAJ=ETA(K-1)+DETA
000204          PHIJ=PHI(K-1)+DETA/DEETA*(PHI(K)-PHI(K-1))
000211          EI2J=EI2(K-1)+DETA/DEETA*(EI2(K)-EI2(K-1))
000216          EI3J=EI3(K-1)+DETA/DEETA*(EI3(K)-EI3(K-1))
000223          STNSG=(EI2J-PHIB*EI1J)/(1.0-PHIB)
000231          VBSIG=EI1J*(TOB-CA2*PHIB**2)/(1.0-CA2)-ETAM*PHIB
000240          IF (PHIB .EQ. 0.0) GO TO 20
000242          SIGVB=VBSIG/PHIB
000243          GO TO 26
000243       20 DO 22 I=1,400
000245          IF (PHI(I+1) .GT. PHID) GO TO 24
000251       22 CONTINUE
000252       24 DELP=PHID-PHI(I)
000254          DPHI=PHI(I+1)-PHI(I)
000257          DETA=ETA(I+1)-ETA(I)
000261          DEI1=EI1(I+1)-EI1(I)
000264          ETAD=ETA(I)+DETA/DPHI*DELP
000270          EI1D=EI1(I)+DEI1/DPHI*DELP
000273          SIGVB=EI1J-EI1D
000275       26 RETURN
000276          END
```

```
                 SUBROUTINE BREAK(J)
          C
          C      CALCULATION OF WAKE IMPINGMENT ON SHROUD WALL
          C
000003           COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003           COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
                1        PHS(100),ASASS(100),DASDX(100),ISLP
000003           COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003           COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003           COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003           COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003           COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000003           COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
                1        XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
                2        PA4B,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
                3        SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000003           COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003           REAL KAPPA
000003           FUNA(G,AM)=((G+1.0)/2.0)**(-(G+1.0)/(2.0*(G-1.0)))*1.0/AM*(1.0+
                1             (G-1.0)/2.0*AM*AM)**((G+1.0)/(2.0*(G-1.0)))
000034           FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000054           FUNQ(G,AM)=SQRT((G+1.0)/2.0*AM*AM/(1.0+(G-1.0)/2.0*AM*AM))
000074           RECV(G,P)=(((G+1.0)*P+(G-1.0))/((G-1.0)*P+(G+1.0)))**(G/(G-1.0))*
                1            P**(-1.0/(G-1.0))
000117           ESCAPE(G,PR)=SQRT((G+1.0)/(G-1.0)*(1.0-PR**(-(G-1.0)/G)))
000140           CROCCO(G,AM)=AM*AM/(2.0/(G-1.0)+AM*AM)
000150           TEST(G,PR)=1.0-PR**(-(G-1.0)/G)
000162           AVE(X1,X2)=(X1+X2)/2.0
000170       600 FORMAT (1H1,//35X,38HWAKE IMPINGES ON SHROUD WALL AT XSLP =F9.6,8H
                1, YSLP =F9.6//)
000170       602 FORMAT (24X,6HWTFL =F9.6,4X,7HWLEAK =F9.6,4X,8HPRATIO =F9.6,4X,
                19HTOS/TOP =F9.6)
000170       604 FORMAT (1H1)
000170           KAPPA=1.000
000172           WTOL=0.0005
000173           PTOL=0.0005
000174        10 I=ISLP-1
000176           DYDX=TAN(THETA(I))
000201           DTDX=(THETA(I)-THETA(I-1))/(XSLP(I)-XSLP(I-1))
000205           CALL FIND(XSLP(I),YP,DYPDX,2.0)
000211           XREF=XSLP(I)+(YP-YSLP(I))/(DYDX-DYPDX)
000217           CALL FIND(XREF,YREF,DYRDX,2.0)
000222           THETAP=THETA(I)+DTDX*(XREF-XSLP(I))
000227           PREF=HSHP
000230           AMREF=FUNM(GAMP,PREF)
000236           THETAR=PMER(AMP(I),THETAP,AMREF,GAMP)
000243        12 DELTA=THETAR-ATAN(DYRDX)
000247           IF (DELTA .LE. 0.0) GO TO 18
000251           IF (DELTA .GT. 0.0) PRATIO=PSI(GAMP,AMREF,DELTA)
000257           RECOMP=1.0+KAPPA*(PRATIO-1.0)
000263           PEST=TEST(GAMP,RECOMP)
000266           IF (PEST .LT. 0.0) CALL EXIT
000271           AMA=FUNQ(GAMP,AMREF)
000276           AMD=ESCAPE(GAMP,RECOMP)
000302           B=(1.0-TOS/TOP)*(AMD/AMA)**2/2.0
000310           C=TOS/TOP*(AMD/AMA)**2
000313           PHID=B+SQRT(B*B+C)
000321           TOSTOP=TOS/TOP+(1.0-TOS/TOP)*PHID
000326           PHIB=0.0
000327           GAMI=(GAMP+WTFL*GAMS)/(1.0+WTFL)
000335           SIISI=(1.0+PHIB)/(1.0-PHIB)
000341           CAIISQ=CROCCO(GAMP,AMREF)
000345           CAISQ=CAIISQ*(1.0-PHIB)**2/(CAIISQ*(1.0-PHIB)**2+(1.0-CAIISQ))
000356           SIGI=12.0+2.758*SQRT(CAISQ)/(SQRT((1.0-CAISQ)*(GAMI-1.0)/2.0))
000372           SIGII=SIISI*SIGI
000374           ARATIO=FUNA(GAMP,AMR)/FUNA(GAMP,AMREF)
000402           CALL MIX(TOS/TOP,CAIISQ,PHIB,PHID,SIGVB)
000407           WLEAK(NITER)=2.0/SIGII*XREF*YREF*ARATIO*SIGVB
000415           XSLP(ISLP)=XSLP(ISLP-1)
000417           YSLP(ISLP)=YSLP(ISLP-1)
000421           THETA(ISLP)=ATAN(DYPDX)
000424           PHS(ISLP)=1.0/RECOMP
```

```
000426          AMS(ISLP)=FUNM(GAMS,PHS(ISLP))
000435          ASASS(ISLP)=FUNA(GAMS,AMS(ISLP))
000441          PHP(ISLP)=RECOMP*PREF
000443          AMP(ISLP)=FUNM(GAMP,PHP(ISLP))
000451          AREA(NITER)=AMIN1(AREA(NITER),ASASS(ISLP))
000456          POINT=-1.0
000457          NHIGH=0
000460          DO 14 I=1,NITER
000461          IF (WLEAK(I) .EQ. 0.0) GO TO 14
000462          IF (WLEAK(I) .GT. WTFL) NHIGH=1+NHIGH
000466       14 CONTINUE
000471          WIFF=ABS(WTFL-WLEAK(NITER))
000474          IF (NITER .EQ. 1) PIFF=1.0
000477          IF (NITER .GT. 1) PIFF=ABS(PTS(NITER)-PTS(NITER-1))
000504          IF (NHIGH .GT. 0 .AND. PIFF .LE. PTOL*PTS(NITER)) GO TO 16
000517          IF (WIFF .GT. WTOL) GO TO 18
000522       16 POINT=0.0
000523          TYPE=0.0
000524          STAG=1.0
000525          CHANGE=1.0
000526          PTS(NITER+1)=HSHP
000530          WLEAK(NITER)=WTFL
000531          ASEC=YP*YP**FDIM-YSLP(ISLP)*YSLP(ISLP)**FDIM
000544          IF (ABS(DYPDX) .GT. 0.0) CALL AJAX(XP,YP,ALPHA,ASEC,DADX)
000553          ASPAPS=ASEC/APRIM*APREF/ASASS(ISLP)
000557          IF (WTFL .GT. 0.0) ASSAPS=AMIN1(ASSAPS,ASPAPS)
000564          IF (WTFL .EQ. 0.0) ASSAPS=ASPAPS
000567          HSHP=PHP(ISLP)/PHS(ISLP)
000572          P(1,J-1)=AVE(PHP(ISLP),PHP(ISLP-1))
000575          T(1,J-1)=AVE(THETA(ISLP),THETA(ISLP-1))
000601          CALL FIELD(J-1)
000604       17 ISLP=ISLP+1
000606          CALL SLIP(J)
000610          CALL SLID(J)
000612       18 RETURN
000613          END
```

```
                SUBROUTINE INSERT(NPT)
          C
          C     INSERTION OF FIELD POINTS
          C
000003          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1       PHS(100),ASASS(100),DASDX(100),ISLP
000003          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000003          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000003          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003          DIMENSION C(16),D(6)
          C
          C     SUBROUTINE INSERT
          C
000003          NPTS=1+NPT
000005          XNPTS=NPTS
000006          DX=(X(1,2)-X(1,1))/XNPTS
000011          DY=(Y(1,2)-Y(1,1))/XNPTS
000014          DT=(T(1,2)-T(1,1))/XNPTS
000017          DO 12 J=1,100
000020          X(2,J)=X(1,J)
000024          Y(2,J)=Y(1,J)
000026          P(2,J)=P(1,J)
```

```
000030          T(2,J)=T(1,J)
000032          CALL COAVE(XAVE,YAVE,PAVE,TAVE,1,1,1,2)
000041          CALL COEFF(XAVE,YAVE,PAVE,TAVE,C,GAMP)
000045          D(1)=C(1)
000047          D(3)=1.0/(C(3)*C(5)*C(7))
000052          D(5)=FDIM*C(9)*C(11)/(C(13)*C(15))
000057       12 CONTINUE
000062          DO 14 J=2,NPTS
000063          X(1,J)=X(1,J-1)+DX
000070          Y(1,J)=Y(1,J-1)+D(1)*(X(1,J)-X(1,J-1))
000075          T(1,J)=T(1,J-1)+DT
000101          P(1,J)=(D(3)*P(1,J-1)-(T(1,J)-T(1,J-1))-D(5)*(Y(1,J)-Y(1,J-1)))/
               1        D(3)
000114       14 CONTINUE
000117          I=NPTS
000120          DO 16 J=2,100
000121          IF (P(2,J) .EQ. 0.0) GO TO 16
000123          I=1+I
000124          X(1,I)=X(2,J)
000127          Y(1,I)=Y(2,J)
000132          P(1,I)=P(2,J)
000134          T(1,I)=T(2,J)
000137       16 CONTINUE
000141       18 DO 20 J=1,100
000143          X(2,J)=0.0
000145          Y(2,J)=0.0
000146          P(2,J)=0.0
000150          T(2,J)=0.0
000151       20 CONTINUE
000153          RETURN
000154          END
```

```
                SUBROUTINE CLEAR(KSHIFT,JREF)
          C
          C
          C
000005          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000005          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1        PHS(100),ASASS(100),DASDX(100),ISLP
000005          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000005          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000005          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000005          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000005          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000005          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1        XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2        PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3        SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000005          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000005          COMMON/BNLYR/XSUM(100),YSUM(100),DELSHD(100),THETAS(100),
               1        CFSHD(100),XCNE(100),YCNE(100),DELCNE(100),THETAC(100),
               2        CFCNE(100),REYPRM,POP,AOP,VOP,REYSEC,POS,AOS,VOS,PEX,XSCALE
000005          ERROR=0.0001
000006          XREF=X(2,JREF)
000010          IF (ISLP .GT. 1) GO TO 14
000014          DO 10 I=1,2
000015          DO  8 J=1,100
000016          X(I,J)=0.0
000020          Y(I,J)=0.0
000022          P(I,J)=0.0
000024          T(I,J)=0.0
000025        8 CONTINUE
000027       10 CONTINUE
000031          DO 12 I=1,100
000033          IF (I .GT. 1) DELSHD(I)=0.0
000036          THETAS(I)=0.0
000040          CFSHD(I)=0.0
```

```
000041          DELCNE(I)=0.0
000042          THETAC(I)=0.0
000043          CFCNE(I)=0.0
000044       12 CONTINUE
000046          GO TO 26
000046       14 J=0
000047          KMAX=100-KSHIFT
000051          DO 18 K=1,JREF
000052          I=K+KSHIFT
000054          TEST=ABS(X(2,I)-X(2,I-1))
000061          IF (X(2,I) .EQ. X(2,I+1)) GO TO 16
000063          IF (KSHIFT .GT. 0 .AND. TEST .LE. ERROR) GO TO 16
000074          J=1+J
000075          X(1,J)=X(2,I)
000101          Y(1,J)=Y(2,I)
000103          P(1,J)=P(2,I)
000105          T(1,J)=T(2,I)
000110       16 X(2,I)=0.0
000112          Y(2,I)=0.0
000114          P(2,I)=0.0
000115          T(2,I)=0.0
000117       18 CONTINUE
000121       20 KREF=1+J
000123          DO 24 K=KREF,100
000124          DO 22 I=1,2
000125          X(I,K)=0.0
000127          Y(I,K)=0.0
000131          P(I,K)=0.0
000132          T(I,K)=0.0
000134       22 CONTINUE
000135       24 CONTINUE
000137       26 RETURN
000140          END
```

```
                SUBROUTINE CHECK(J,SHOCK)
          C
          C     CHECK FOR COALESCENCE OF CHARACTERISTICS
          C
000005          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000005          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1       PHS(100),ASASS(100),DASDX(100),ISLP
000005          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000005          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000005          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000005          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000005          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000005          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000005          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000005          AVE(X1,X2)=(X1+X2)/2.0
000014      600 FORMAT (//35X,30HCOALESENCE HAS OCCURED AT XP =F8.5,6H, YP =F8.5)
000014          SHOCK=0.0
000015          LREF=J-1
000017          DO 12 L=1,LREF
000020          K=J-L
000021          IF (X(2,J) .GT. X(2,2)) GO TO 6
000025          IF (Y(2,J) .GT. Y(2,2)) GO TO 6
000031          X(2,J)=X(2,2)
000033          Y(2,J)=Y(2,2)
000034          P(2,J)=P(2,2)
000036          T(2,J)=T(2,2)
000040          GO TO 8
000041        6 IF (X(2,J) .GT. X(2,K)) GO TO 12
000047          IF (Y(2,J) .GT. Y(2,K)) GO TO 12
000053        8 DO 10 N=K,LREF
```

```
000055          X(2,N)=X(2,J)
000060          Y(2,N)=Y(2,J)
000063          P(2,N)=P(2,J)
000065          T(2,N)=T(2,J)
000070       10 CONTINUE
000072       12 CONTINUE
000075       14 IF (X(2,J) .GT. X(1,J)) GO TO 20
000103          IF (Y(2,J) .LT. Y(1,J)) GO TO 20
000106          DO 16 I=J,100
000107          IF (P(1,I) .EQ. 0.0) GO TO 18
000111          X(2,I)=X(1,I)
000114          Y(2,I)=Y(1,I)
000116          P(2,I)=P(1,I)
000120          T(2,I)=T(1,I)
000122       16 CONTINUE
000124       18 J=I-1
000126          SHOCK=1.0
000127       20 RETURN
000130          END
```

```
                SUBROUTINE ESTMP
          C
          C     ESTIMINATION OF TOTAL PRESSURE RATIO FOR CHOKED SECONDARY FLOW
          C
          C     IF CHOKE=-1.0 SOLUTION HAS NOT BEEN FOUND
          C     IF CHOKE= 0.0 SOLUTION HAS BEEN FOUND, SECONDARY FLOW CHOKED
          C     IF CHOKE= 1.0 SOLUTION HAS BEEN FOUND, SECONDARY FLOW UNCHOKED
          C
000002          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1        PHS(100),ASASS(100),DASDX(100),ISLP
000002          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000002          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1        XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2        PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3        SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002      600 FORMAT (1H1)
000002          PTOL=0.0001
000004          ATOL=1.05
000005          DELP=1.02
000007          AREF=1.001
000010          HSHP=PTS(NITER)
000012          AREA(NITER)=ASASS(ISLP)
000014          DO 8 I=1,ISLP
000016          AREA(NITER)=AMIN1(AREA(NITER),ASASS(I))
000023        8 CONTINUE
000025          IF (POINT .NE. 1.0) GO TO 10
000027          IF (AREA(NITER) .GE. 1.0 .AND. AREA(NITER) .LE. ATOL) GO TO 22
000041       10 IF (NITER .GT. 1) GO TO 12
000045          IF (POINT .EQ. -1.0) HSHP=HSHP*DELP
000050          IF (POINT .EQ.  1.0) HSHP=HSHP/DELP
000054          GO TO 26
000055       12 NLOW=0
000056          NHIGH=0
000057          PLOW=0.0
000060          PMIN=1.0
000061          DO 16 I=1,NITER
000063          IF (AREA(I) .GE. 1.0) GO TO 14
000066          NLOW=1+NLOW
000067          PLOW=AMAX1(PLOW,PTS(I))
000073          GO TO 16
000073       14 NHIGH=1+NHIGH
```

```
000075          PMIN=AMIN1(PMIN,PTS(I))
000101          IF (PMIN .EQ. PTS(I)) AMIN=AREA(I)
000104       16 CONTINUE
000107          IF (NHIGH .GT. 1) GO TO 18
000112          IF (ABS(PMIN-PLOW) .LE. PTOL*PMIN) GO TO 24
000117          IF (NHIGH .EQ. 0 .AND. NLOW .GE. 2) DELP=DELP*DELP
000130          IF (NHIGH .EQ. 0) HSHP=PLOW*DELP
000133          IF (NHIGH .EQ. 1) HSHP=PMIN/DELP
000137          IF (HSHP .LE. PLOW) HSHP=PMIN-(PMIN-PLOW)/4.0
000146          GO TO 26
000147       18 IF (ABS(PMIN-PLOW) .LE. PTOL*PMIN) GO TO 24
000154          PMAX=1.0
000156          DO 20 I=1,NITER
000157          IF (AREA(I) .LT. 1.0) GO TO 20
000162          IF (PTS(I) .EQ. PMIN) GO TO 20
000164          PMAX=AMIN1(PTS(I),PMAX)
000167          IF (PMAX .EQ. PTS(I)) AMAX=AREA(I)
000172       20 CONTINUE
000175          DMIN=AMIN*(1.0+AMIN/2.0)
000200          DMAX=AMAX*(1.0+AMAX/2.0)
000203          ALPHA=(PMIN*DMAX-PMAX*DMIN)/(DMAX-DMIN)
000211          GAMMA=(PMAX-PMIN)/(DMAX-DMIN)
000214          HSHP=ALPHA+GAMMA*AREF*(1.0+AREF/2.0)
000222          HLOW=(1.0+PTOL)*PLOW
000225          HMIN=(1.0-PTOL)*PMIN
000227          IF (HSHP .LE. HLOW .OR. HSHP .GE. HMIN) HSHP=PMIN-(PMIN-PLOW)/4.0
000245          GO TO 26
000246       22 HSHP=HSHP
000247          CHOKE=0.0
000250          PTS(NITER+1)=HSHP
000252          GO TO 26
000253       24 HSHP=PMIN
000255          CHOKE=1.0
000256          PTS(NITER+1)=HSHP
000260       26 RETURN
000261          END
```

```
                SUBROUTINE ESTMW
          C
          C     ESTIMATION OF TOTAL PRESSURE RATIO FOR WLEAK=WTFL
          C
000002          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
              1         PHS(100),ASASS(100),DASDX(100),ISLP
000002          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000002          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
              1         XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
              2         PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
              3         SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002          AVE(X1,X2)=(X1+X2)/2.0
000011          NLOW=0
000012          NMAX=0
000013          NHIGH=0
000014          PLOW=0.0
000015          PMAX=1.0
000016          PHIGH=1.0
000017          DELP=1.02
000020          DO 10 I=1,NITER
000022          IF (ABS(WLEAK(I)) .GT. 0.0) GO TO 10
000026          NMAX=1+NMAX
000027          PMAX=AMIN1(PTS(I),PMAX)
000033       10 CONTINUE
```

```
000036          DO 14 I=1,NITER
000037          IF (WLEAK(I) .EQ. 0.0) GO TO 14
000040          IF (WLEAK(I) .GT. WTFL) GO TO 12
000044          IF (PTS(I) .GT. PMAX) GO TO 14
000047          NLOW=1+NLOW
000050          PLOW=AMAX1(PTS(I),PLOW)
000054          IF (PLOW .EQ. PTS(I)) WLOW=WLEAK(I)
000057          GO TO 14
000060       12 IF (PTS(I) .GT. PMAX) GO TO 14
000064          NHIGH=1+NHIGH
000065          PHIGH=AMIN1(PTS(I),PHIGH)
000071          IF (PHIGH .EQ. PTS(I)) WHIGH=WLEAK(I)
000074       14 CONTINUE
000077       16 IF (NLOW .GT. 0 .AND. NHIGH .GT. 0) GO TO 18
000107          IF (NMAX .GE. 0 .AND. NHIGH .EQ. 0) HSHP=PTS(NITER)*DELP
000121          IF (NMAX .GE. 0 .AND. NLOW .EQ. 0) HSHP=PTS(NITER)/DELP
000132          IF (NMAX .GT. 0 .AND. NLOW .GT. 0) HSHP=PMAX-(PMAX-PLOW)/2.0
000146          GO TO 22
000147       18 IF (ABS(WHIGH-WLOW) .LE. 0.001) GO TO 20
000154          HSHP=PHIGH-(PHIGH-PLOW)/4.0
000160          GO TO 22
000160       20 DPDW=(PHIGH-PLOW)/(WHIGH-WLOW)
000164          HSHP=PLOW+DPDW*(WTFL-WLOW)
000170       22 RETURN
000171          END
```

```
                SUBROUTINE STORE(J)
          C
          C     STORAGE OF PERTINENT INFORMATION ALONG SLIPLINE
          C
000003          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1         PHS(100),ASASS(100),DASDX(100),ISLP
000003          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003          COMMON XBDY(100),YBDY(100),DYSDX(100),NBDY
000003          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1         XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2         PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3         SOLVF,CHOKF,CHANGE,CHARGE,TYPE,POINT,STAG
000003          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003          FUNA(G,AM)=((G+1.0)/2.0)**(-(G+1.0)/(2.0*(G-1.0)))*1.0/AM*(1.0+
               1           (G-1.0)/2.0*AM*AM)**((G+1.0)/(2.0*(G-1.0)))
000034          FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000054          XSLP(ISLP)=X(2,J)
000057          YSLP(ISLP)=Y(2,J)
000061          PHP(ISLP)=P(2,J)
000063          THETA(ISLP)=T(2,J)
000065          AMP(ISLP)=FUNM(GAMP,PHP(ISLP))
000073          IF (STAG .GE. 0.0) GO TO 10
000075          PHS(ISLP)=1.0
000076          AMS(ISLP)=0.0
000077          ASASS(ISLP)=500.0
000101          GO TO 12
000101       10 PHS(ISLP)=PHP(ISLP)/HSHP
000104          AMS(ISLP)=FUNM(GAMS,PHS(ISLP))
000111          ASASS(ISLP)=FUNA(GAMS,AMS(ISLP))
000115       12 RETURN
000116          END
```

```
                SUBROUTINE COAVE(XAVE,YAVE,PAVE,TAVE,I1,J1,I2,J2)
          C
          C     AVERAGING SUBROUTINE
          C
000013          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000013          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1        PHS(100),ASASS(100),DASDX(100),ISLP
000013          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000013          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000013          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000013          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000013          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000013          COMMON WTEL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1        XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2        PA43,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3        SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000013          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000013          COMMON/CPLOT/IPLOT,XSTART,YSTART,XSPAN,YSPAN,SCALE,SPAN,AXIS,
               1      XORGN,YORGN,XSHFT,YSHFT,KKK(14),PP(14),XDOWN(100),YACROS(100)
000013          COMMON/CLPLOT/XPEN,YPEN,NX,NY,IPEN,XLABEL(10),YLABEL(10)
000013          COMMON/SPECL/TEST,ORGSET,SPASET
000013          AVE(X1,X2)=(X1+X2)/2.0
000022          FUNP(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-G/(G-1.0))
000037      600 FORMAT (1H1,//33X,62HCONDITIONS OF SONIC FLOW HAVE BEEN REACHED IN
               1 SUBROUTINE COAVE///)
000037      602 FORMAT (29X,6HXAVE =F9.5,4X,6HYAVE =F8.5,4X,6HPAVE =F8.5,4X,6HTAVE
               1 =F9.5)
000037          PMAX=FUNP(GAMP,1.0)
000044          XAVE=AVE(X(I1,J1),X(I2,J2))
000054          YAVE=AVE(Y(I1,J1),Y(I2,J2))
000064          PAVE=AVE(P(I1,J1),P(I2,J2))
000074          TAVE=AVE(T(I1,J1),T(I2,J2))
000104          IF (PAVE .LT. PMAX) GO TO 10
000106          WRITE (6,600)
000111          WRITE (6,602) XAVE,YAVE,PAVE,TAVE
000130          CALL OUTSLP
000131          IF (CHOKE .EQ. -1.0) CALL OUTLYR
000144          IF (IPLOT .EQ. 1) CALL PLOTL
000153          CALL EXIT
000154       10 RETURN
000155          END
```

```
                SUBROUTINE COEFF(XP,YP,PPS,ANGP,CFS,GAM)
          C
          C     COEFFICIENTS TO THE CHARACTERISTIC EQUATIONS
          C
000011          DIMENSION CFS(16)
000011          FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000035          AMP=FUNM(GAM,PPS)
000040          AMUP=ASIN(1.0/AMP)
000044          ANGPU=ANGP+AMUP
000051          ANGMU=ANGP-AMUP
000053          CFS(1)=TAN(ANGPU)
000062          CFS(2)=TAN(ANGMU)
000070          CFS(3)=GAM*PPS
000072          CFS(4)=CFS(3)
000073          CFS(5)=AMP*AMP
000075          CFS(6)=CFS(5)
000076          CFS(7)=TAN(AMUP)
000105          CFS(8)=CFS(7)
000107          CFS(9)=SIN(ANGP)
000115          CFS(10)=CFS(9)
000117          CFS(11)=SIN(AMUP)
000125          CFS(12)=CFS(11)
000127          CFS(13)=YP
000130          CFS(14)=CFS(13)
000132          CFS(15)=SIN(ANGPU)
000140          CFS(16)=SIN(ANGMU)
000147          RETURN
000150          END
```

```
                 SUBROUTINE DZDXDY(W,WJ,ALPHA,THETA,DELW,DW,DT,X,Y)
          C
          C      TRANSFORMATION INTEGRATION
          C
000014           COMPLEX F,G,H,DF,DZ,DFDZ
000014           DIMENSION A(3),B(3),DX(3),DY(3)
000014           A(1)=W
000014           A(2)=W-DW/2.0
000017           A(3)=W-DW
000020           B(1)=THETA
000021           B(2)=THETA-DT/2.0
000025           B(3)=THETA-DT
000027           DO 10 I=1,3
000030           DZ=CMPLX(DW,-DT)
000033           CALL DFDQB(A(I),WJ,ALPHA,B(I),DELW,DFDZ)
000042           F=CMPLX(COS(B(I)),SIN(B(I)))
000054           G=EXP(-A(I))*F
000066           H=EXP(A(I))*CONJG(F)
000103           DF=G*DFDZ*DZ-CONJG(H*DFDZ*DZ/4.0)
000131           DX(I)=REAL(DF)
000134           DY(I)=AIMAG(DF)
000136        10 CONTINUE
000143           X=(DX(1)+4.0*DX(2)+DX(3))/6.0
000150           Y=(DY(1)+4.0*DY(2)+DY(3))/6.0
000157           RETURN
000160           END
```

```
                 SUBROUTINE DFDQB(W,WJ,ALPHA,THETA,DELW,DFDZ)
          C
          C      EVALUATION OF TRANSFORMATION DERIVATIVE
          C
000011           COMPLEX P,Q,R,DFDZ
000011           PI=3.1415927
000012           A=PI/ALPHA*(W-WJ)
000015           B=PI/ALPHA*THETA
000016           C=PI/ALPHA*DELW
000020           P=CMPLX(SINH(A)*COS(B),-COSH(A)*SIN(B))
000034           Q=CMPLX(COSH(A)*COS(B)-COSH(C),-SINH(A)*SIN(B))
000054           R=CMPLX(COSH(A)*COS(B)-1.0,-SINH(A)*SIN(B))
000071           IF (DELW .EQ. 0.0) DFDZ=-PI/ALPHA*P/R
000110           IF (DELW .NE. 0.0) DFDZ=PI/ALPHA*(P/Q-P/R)
000137           RETURN
000140           END
```

```
                 SUBROUTINE CONIC(XP,YP,UP,VP,VUP,GAM)
          C
          C      CALCULATION OF CONDITIONS IN A CONICAL FLOW FIELD
          C
000011           DELUP=0.0001
000012           ERROR=0.0001
000013           SIGSQ=(GAM-1.0)/(GAM+1.0)
000017           DELTA=YP/XP
000020           TEST=ABS(DELTA)-ABS(1.0/VUP)
000025           IF (ABS(TEST) .LE. ERROR*ABS(DELTA)) GO TO 12
000032           SIGNT=TEST/ABS(TEST)
000035           DELUP=SIGNT*DELUP
000036        10 CALL RUNGE(UP,VP,VUP,DELUP,SIGSQ)
000041           PEST=TEST
000042           SIGNP=PEST/ABS(PEST)
000044           TEST=ABS(DELTA)-ABS(1.0/VUP)
000054           IF (ABS(TEST) .LE. ERROR*ABS(DELTA)) GO TO 12
000061           SIGNT=TEST/ABS(TEST)
```

```
000064          IF (SIGNP .EQ. SIGNT) GO TO 10
000066          DELUP=-0.50*((1.0/DELTA)+VUP)/FUNV(UP,VP,VUP,SIGSQ)
000101          IF (DELUP .EQ. 0.0) GO TO 12
000103          GO TO 10
000103       12 RETURN
000104          END
```

```
                SUBROUTINE RUNGE(UP,VP,VUP,DU,SIGSQ)
          C
          C     RUNGE KUTTA INTEGRATION OF TAYLOR-MACCOLLI EQUATION
          C
000010          REAL K1,K2,K3,K4
000010          K1=DU*FUNV(UP,VP,VUP,SIGSQ)
000015          K2=DU*FUNV(UP+DU/2.0,VP+DU/2.0*VUP+K1*DU/8.0,VUP+K1/8.0,SIGSQ)
000037          K3=DU*FUNV(UP+DU/2.0,VP+DU/2.0*VUP+K1*DU/8.0,VUP+K2/8.0,SIGSQ)
000061          K4=DU*FUNV(UP+DU,VP+DU*VUP+K3*DU/2.0,VUP+K3,SIGSQ)
000100          UP=UP+DU
000101          VP=VP+DU*(VUP+(K1+K2+K3)/6.0)
000107          VUP=VUP+(K1+2.0*K2+2.0*K3+K4)/6.0
000117          RETURN
000117          END
```

```
                FUNCTION FUNV(UP,VP,VUP,SIGSQ)
          C
          C     TAYLOR-MACCOLLI EQUATION
          C
000007          FUN1=(1.0+VUP**2)
000010          FUN2=(1.0-SIGSQ)*(UP+VP*VUP)**2
000014          FUN3=(1.0-SIGSQ*(UP**2+VP**2))
000017          FUNV=(1.0/VP)*(FUN1-FUN2/FUN3)
000024          RETURN
000024          END
```

```
                FUNCTION PMER(AMP,ANGP,AMQ,GAM)
          C
          C     PRANDTL MEYER EXPANSION ANGLE
          C
000007          AK=SQRT((GAM-1.0)/(GAM+1.0))
000020          FAMP=SQRT(AMP*AMP-1.0)
000027          THETAP=ATAN(AK*FAMP)/AK-ATAN(FAMP)
000040          FAMQ=SQRT(AMQ*AMQ-1.0)
000051          THETAQ=ATAN(AK*FAMQ)/AK-ATAN(FAMQ)
000062          PMER=ANGP+(THETAQ-THETAP)
000070          RETURN
000070          END
```

```
                SUBROUTINE ASTAR(MACH,AREA,GAM)
          C
          C     CALCULATION OF MACH NUMBER AS A FUNCTION OF A/A*
          C
000006          REAL MACH
000006          FUNA(G,AM)=(2.0/(G+1.0)*(1.0+(G-1.0)/2.0*AM*AM))**
              1            ((G+1.0)/(2.0*(G-1.0)))
```

```
000027         FUNB(G,AM)=AM*(2.0/(G+1.0)*(1.0+(G-1.0)/2.0*AM*AM))**
              1             ((3.0-G)/(2.0*(G-1.0)))
000040         ITER=0
000041         ERROR=0.00001
000053      10 ITER=1+ITER
000055         F=MACH-FUNA(GAM,MACH)/AREA
000060         DFDM=1.0-FUNB(GAM,MACH)/AREA
000064         MACH=MACH-F/DFDM
000067         IF (ITER .LT. 100) GO TO 12
000072         CALL EXIT
000073      12 IF (ABS(F/DFDM/MACH) .GT. ERROR) GO TO 10
000103      14 RETURN
000104         END
```

```
               FUNCTION PSI(G,AM1,DEL)
         C
         C     SERIES EXPANSION FOR PRESSURE RATIO ACROSS AN OBLIQUE SHOCK
         C
000006         EXP=7.0/2.0
000007         AM2=AM1*AM1
000010         AM4=AM2*AM2
000011         AM6=AM2*AM4
000013         AM8=AM4*AM4
000015         BETA=AM2-1.0
000016         A=1.0
000017         B=G*AM2/SQRT(BETA)
000025         C=G*AM2/(4.0*BETA**2)*((G+1.0)*AM4-4.0*BETA)
000034         D=G*AM2/BETA**EXP*((G+1.0)**2/32.0*AM8-(7.0+12.0*G-3.0*G*G)/24.0*
              1   AM6+3.0/4.0*(G+1.0)*AM4-AM2+2.0/3.0)
000065         PSI=A+B*DEL+C*DEL**2+D*DEL**3
000076         RETURN
000077         END
```

```
               SUBROUTINE CONE
         C
         C     CALCULATION OF CONIC PLUG CONTOUR
         C
000002         COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002         COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
              1         PHS(100),ASASS(100),DASDX(100),ISLP
000002         COMMON XIS(21,26),YIS(21,26),N(21),TAU(26),NSONIC,NANGLE
000002         COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002         COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002         COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002         COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002         COMMON WTFL,HSHP,TUS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
              1         XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
              2         PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
              3         SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002         COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002         NBDY=45
000003         DIV=44.0
000005         B=-TAN(CONVR*CONA)
000011         A=DBDY/2.0
000013         X0=-DPRIM/4.0
000015         XR=-A/B
000017         DELX=(XR-X0)/DIV
000022         XBDY(1)=X0
000023         YBDY(1)=A+B*X0
000025         DO 10 I=2,NBDY
000027         XBDY(I)=XBDY(I-1)+DELX
000032         YBDY(I)=A+B*XBDY(I)
000035      10 CONTINUE
000037         RETURN
000040         END
```

```
              SUBROUTINE FIND(XP,YP,DYPDX,SURF)
        C
        C     LOCATION OF SHROUD CONTOUR POINT
        C
000007        COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000007        COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
             1       PHS(100),ASASS(100),DASDX(100),ISLP
000007        COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000007        COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000007        COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000007        COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000007        COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000007        COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
             1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
             2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
             3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000007        COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000007    600 FORMAT (1H1,//37X,53HSHROUD POINT (XP) LIES OUTSIDE RANGE OF INPUT
             1 CONTOUR//)
000007    602 FORMAT (//24X,9HXSHD(1) =F8.5,15X,4HXP =F8.5,15X,12HXSHD(NSHD) =F8
             1.5//)
000007    604 FORMAT (1H1,//38X,51HBODY POINT (XP) LIES OUTSIDE RANGE OF INPUT C
             1ONTOUR//)
000007    606 FORMAT (//24X,9HXBDY(1) =F8.5,15X,4HXP =F8.5,15X,12HXBDY(NBDY) =F8
             1.5//)
000007        IF (SURF .EQ. 1.0) GO TO 14
000011        IF (NSHD .GT. 1) GO TO 10
000014        YP=YSHD(1)
000015        DYPDX=0.0
000015        GO TO 20
000016     10 IF (XP .GE. XSHD(1) .AND. XP .LE. XSHD(NSHD)) GO TO 12
000027        WRITE (6,600)
000032        WRITE (6,602) XSHD(1),XP,XSHD(NSHD)
000054        CALL EXIT
000055     12 CALL SINTP(XSHD,YSHD,NSHD,XP,YP)
000064        CALL SINTP(XSHD,DYSDX,NSHD,XP,DYPDX)
000073        GO TO 20
000076     14 IF (NBDY .GT. 0) GO TO 16
000101        YP=0.0
000101        DYPDX=0.0
000102        GO TO 20
000102     16 IF (XP .GE. XBDY(1) .AND. XP .LE. XBDY(NBDY)) GO TO 18
000113        WRITE (6,604)
000116        WRITE (6,606) XBDY(1),XP,XBDY(NBDY)
000140        CALL EXIT
000141     18 CALL SINTP(XBDY,YBDY,NBDY,XP,YP)
000150        CALL SINTP(XBDY,DYBDX,NBDY,XP,DYPDX)
000157     20 RETURN
000160        END
```

```
              SUBROUTINE AJAX(XP,YP,ALPHA,ASEC,DADX)
        C
        C     CALCULATION OF FLOW AREA IN SECONDARY PASSAGE
        C
000010        COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000010        COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
             1       PHS(100),ASASS(100),DASDX(100),ISLP
000010        COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000010        COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000010        COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000010        COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000010        COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000010        COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
             1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
             2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
             3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000010        COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000010        COMMON/ANLYR/XSUM(100),YSUM(100),DELSHD(100),THETAS(100),
```

```
                1          CFSHD(100),XCNE(100),YCNE(100),DELCNE(100),THETAC(100),
                2          CFCNE(100),REYPRM,POP,AOP,VOP,REYSEC,POS,AOS,VOS,PEX,XSCALE
000010            AVE(X1,X2)=(X1+X2)/2.0
000017        600 FORMAT (1H1,//28X,18A4,//)
000017        602 FORMAT (25X,55HUNABLE TO OBTAIN CONVERGENCE IN SUBROUTINE AJAX, XS
                 1LP =F8.5,8H, YSLP =F8.5)
000017            I=ISLP
000021            ITER=0
000022            ALPHA=0.0
000022            ERROR=0.0005
000024            XP=XSLP(I)
000025            DYPDX=TAN(THETA(I))
000030            CALL FIND(XP,YP,A,2.0)
000035            IF (ISLP .EQ. 1) PEST=0.0
000043         10 ITER=1+ITER
000045            XREF=XP
000046            YREF=YP
000047            B=-1.0/AVE(A,DYPDX)
000053            XP=(YSLP(I)-YREF+A*XREF-B*XSLP(I))/(A-B)
000064            CALL FIND(XP,YP,A,2.0)
000066            TEST=ABS(YP-YREF)
000074            IF (ITER .LT. 100) GO TO 12
000076            IF (PEST .EQ. 0.0) WRITE (6,600) (TITLE(K), K=1,12)
000114            WRITE (6,602) XSLP(ISLP),YSLP(ISLP)
000124            PEST=1.0
000126            GO TO 14
000131         12 IF (TEST .GT. ERROR*YP) GO TO 10
000136         14 CALL SHLYR(DDSDX)
000140            YP=YP-DELSHD(ISLP)
000145            A=A-DDSDX
000147            ALPHA=ATAN(AVE(A,DYPDX))
000160            ASEC=(YP*YP**FDIM-YSLP(I)*YSLP(I)**FDIM)/COS(ALPHA)
000177            DADX=(1.0+FDIM)*(YP**FDIM*A-YSLP(I)**FDIM*DYPDX)/COS(ALPHA)
000221            RETURN
000221            END



                  SUBROUTINE SPLINE (X,Y,N,SLOPE,DUMMY)
           C
           C      CALCULATION OF FIRST AND SECOND DERIVATIVES
           C
000010            DIMENSION X(100),Y(100),S(100),A(100),B(100),C(100),F(100),W(100),
                 1SB(100),G(100),EM(100),SLOPE(100)
000010            DO 10 I=2,N
000011            S(I)= X(I)-X(I-1)
000014         10 CONTINUE
000016            NO= N-1
000020            DO 12 I=2,NO
000021            A(I)= S(I)/6.0
000023            B(I)= (S(I)+S(I+1))/3.0
000027            C(I)= S(I+1)/6.0
000031            F(I)= (Y(I+1)-Y(I))/S(I+1)-(Y(I)-Y(I-1))/S(I)
000040         12 CONTINUE
000044            A(N)= -.5
000046            B(1)= 1.0
000047            B(N)= 1.0
000050            C(1)= -.5
000051            F(1)= 0.0
000051            F(N)= 0.0
000053            W(1)= B(1)
000054            SB(1)= C(1)/W(1)
000056            G(1)= 0.0
000057            DO 14 I=2,N
000060            W(I)= B(I)-A(I)*SB(I-1)
000064            SB(I)= C(I)/W(I)
000066            G(I)= (F(I)-A(I)*G(I-1))/W(I)
000073         14 CONTINUE
000075            EM(N)= G(N)
```

```
000076          DO 16 I=2,N
000100          K= N+1-I
000102          EM(K)= G(K)-SB(K)*EM(K+1)
000107       16 CONTINUE
000112          SLOPE(1)= -S(2)/6.0*(2.0*EM(1)+EM(2))+(Y(2)-Y(1))/S(2)
000123          DO 18 I=2,N
000125          SLOPE(I)= S(I)/6.0*(2.0*EM(I)+EM(I-1))+(Y(I)-Y(I-1))/S(I)
000137       18 CONTINUE
000143          RETURN
000144          END
```

```
                SUBROUTINE SINTP(X,Y,N,X1,Y1)
          C
          C     INTERPOLATION SUBROUTINE
          C
000010          DIMENSION X(100),Y(100)
000010          ERROR=0.0001
000011          XMIN=100.0
000013          DO 10 I=1,N
000014          DELX=X1-X(I)
000016          IF (ABS(DELX) .LT. ERROR) DELX=0.0
000022          IF (DELX .LT. 0.0) GO TO 10
000024          XMIN=AMIN1(DELX,XMIN)
000027          IF (XMIN .EQ. DELX) K=I
000032       10 CONTINUE
000035          IF (XMIN .EQ. 0.0) GO TO 12
000036          IF (K .EQ. 1) K=K+1
000041          IF (K .EQ. N) K=K-1
000043          DELX=X1-X(K)
000045          A=Y(K)
000047          D=(Y(K)-Y(K-1))/(X(K)-X(K-1))
000054          B=(D+(Y(K+1)-Y(K))/(X(K+1)-X(K)))/2.0
000064          C=((Y(K+1)-Y(K))-B*(X(K+1)-X(K)))/((X(K+1)-X(K))*(X(K+1)-X(K-1)))
000100          IF (DELX .LT. 0.0) B=D
000103          Y1=A+B*(X1-X(K))+C*(X1-X(K))*(X1-X(K-1))
000115          GO TO 14
000115       12 Y1=Y(K)
000117       14 RETURN
000120          END
```

```
                SUBROUTINE COMP(K4,SKIP)
          C     COMPLETION OF FLOW FIELD FOR AN EJECTOR WITH A CENTERBODY
          C
000005          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000005          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1       PHS(100),ASASS(100),DASDX(100),ISLP
000005          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000005          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000005          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000005          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000005          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000005          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000005          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000005          COMMON/OUTPN/IPNCH,IPRNT,ICOMP
000005          REAL K4
000005          DIMENSION C(16),D(6)
          C
          C     SUBROUTINE COMP
          C
000005          DO 8 K=1,100
```

```
000006           X(2,K)=0.0
000010           Y(2,K)=0.0
000011           P(2,K)=0.0
000013           T(2,K)=0.0
000014           IF (P(1,K) .GT. 0.0) J=K
000017         8 CONTINUE
000021           IF (XSLP(ISLP) .LT. END) GO TO 10
000024           DELX=END-XSLP(ISLP-1)
000026           DX=XSLP(ISLP)-XSLP(ISLP-1)
000030           DY=YSLP(ISLP)-YSLP(ISLP-1)
000032           DP=PHP(ISLP)-PHP(ISLP-1)
000034           DT=THETA(ISLP)-THETA(ISLP-1)
000041           X(1,J)=XSLP(ISLP-1)+DELX
000044           Y(1,J)=YSLP(ISLP-1)+DY/DX*DELX
000051           P(1,J)=PHP(ISLP-1)+DP/DX*DELX
000057           T(1,J)=THETA(ISLP-1)+DT/DX*DELX
000065        10 XCOMP=XSLP(ISLP)
000067           IF (NBDY .GT. 0) XCOMP=XBDY(NBDY)
000074           DO 16 I=1,100
000076           CALL BOUND(2)
000077           DO 12 J=3,100
000102           IF (P(1,J) .EQ. 0.0) GO TO 14
000104           CALL FIELD(J)
000105           CALL CHECK(J,SHOCK)
000107           IF (X(2,J) .GT. XCOMP .OR. SHOCK .EQ. 1.0) GO TO 14
000122        12 CONTINUE
000124        14 CALL CLEAR(1,J)
000126           RADIUS=SQRT((X(1,2)-X(1,1))**2+(Y(1,2)-Y(1,1))**2)
000135           NSERT=RADIUS/K4
000140           IF (P(1,2) .EQ. 0.0) NSERT=0
000142           IF (NSERT .GE. 2) CALL INSERT(NSERT-1)
000152           IF (SKIP .EQ. 1.0) CALL OUTFLD(1)
000160           IF (ICOMP .EQ. 2) CALL PROFLE
000165           XTEST=X(1,1)+2.0*(X(1,2)-X(1,1))
000171           IF (X(1,1) .GT. XCOMP) GO TO 18
000175           IF (XTEST .GT. XCOMP) GO TO 18
000200           IF (X(1,2) .GE. XCOMP .OR. P(1,2) .EQ. 0.0) GO TO 18
000207        16 CONTINUE
000211        18 IF (NBDY .EQ. 0) GO TO 20
000212           ICONE=ICONE+1
000214           XCONE(ICONE)=XBDY(NBDY)
000216           YCONE(ICONE)=YBDY(NBDY)
000217           PCONE(ICONE)=PCONE(ICONE-1)+DP/DX*(XCONE(ICONE)-XCONE(ICONE-1))
000224           TCONE(ICONE)=ATAN(DYBDX(NBDY))
000231        20 RETURN
000232           END
```

```
                 SUBROUTINE PROFLE
          C
          C      COMPUTATION OF INITIAL VELOCITY PROFILE
          C
000002           COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002           COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
                1       PHS(100),ASASS(100),DASDX(100),ISLP
000002           COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000002           COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002           COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002           COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002           COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002           COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
                1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
                2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
                3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002           COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002           COMMON/BNLYR/XSUM(100),YSUM(100),DELSHD(100),THETAS(100),
                1       CFSHD(100),XCNE(100),YCNE(100),DELCNE(100),THETAC(100),
                2       CFCNE(100),REYPRM,POP,AOP,VOP,REYSEC,POS,AOS,VOS,PEX,XSCALE
```

```
000002        COMMON/PRFLR/XPRF(6),YPRF(6,25),QPRF(6,25),NPRF(6),MAX,NMAX
000002        COMMON/OUTPN/IPNCH,IPRNT,ICOMP
000002        DIMENSION C(16),D(5),YWALL(10,2)
000002        FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000022    500 FORMAT (1H1,//27X,18A4,//)
000022    502 FORMAT (20X,3HI =I3,5X,3HJ =I3,5X,9HXPRF(I) =F8.5,5X,11HYPRF(I,J)
             1=F8.5,5X,11HQPRF(I,J) =F8.5)
000022        MAX=6
000023        NMAX=25
000024        STEP=MAX-1
000027        IF (XPRF(6) .EQ. 0.0) XPRF(6)=END
000031        DELX=(XPRF(6)-XPRF(1))/STEP
000034        IF (ISLP .GT. 2) GO TO 12
000037        DO 10 I=1,MAX
000040        NPRF(I)=0
000041        IF (I .GT. 1) XPRF(I)=XPRF(I-1)+DELX
000046     10 CONTINUE
000051     12 DO 26 I=1,MAX
000053        IF (XPRF(I) .EQ. 0.0 .OR. NPRF(I) .EQ. NMAX) GO TO 26
000064        IF (XPRF(I) .LT. XSLP(ISLP-1) .OR. XPRF(I) .GT. XSLP(ISLP))
             1   GO TO 14
000075        IF (NPRF(I) .GT. 0) GO TO 16
000100        NPRF(I)=3
000101        CALL FIND(XPRF(I),YWALL(I,1),DYWDX,1.0)
000104        CALL FIND(XPRF(I),YWALL(I,2),DYWDX,2.0)
000110        DYDX=(YSLP(ISLP)-YSLP(ISLP-1))/(XSLP(ISLP)-XSLP(ISLP-1))
000115        DPDX=(PHP(ISLP)-PHP(ISLP-1))/(XSLP(ISLP)-XSLP(ISLP-1))
000121        DTDX=(THETA(ISLP)-THETA(ISLP-1))/(XSLP(ISLP)-XSLP(ISLP-1))
000125        YPRFL=YSLP(ISLP-1)+DYDX*(XPRF(I)-XSLP(ISLP-1))
000132        PPRF=PHP(ISLP-1)+DPDX*(XPRF(I)-XSLP(ISLP-1))
000136        TPRF=THETA(ISLP-1)+DTDX*(XPRF(I)-XSLP(ISLP-1))
000142        YPRF(I,1)=1.0
000143        YPRF(I,2)=(YPRFL-YWALL(I,1))/(YWALL(I,2)-YWALL(I,1))
000151        YPRF(I,3)=YPRF(I,2)
000152        QPRF(I,1)=FUNM(GAMS,PPRF/PTS(NITER))
000161        QPRF(I,2)=QPRF(I,1)
000163        QPRF(I,3)=FUNM(GAMP,PPRF)
000170     14 IF (ICONE .EQ. 1) GO TO 16
000172        IF (XCONE(ICONE-1) .LE. XPRF(I) .AND. XCONE(ICONE) .GE.
             1   XPRF(I)) GO TO 22
000204        IF (XCONE(ICONE-1) .GT. XPRF(I)) GO TO 26
000210     16 IF (XPRF(I) .LT. X(1,1)) GO TO 26
000213        DO 18 J=2,100
000215        IF (P(1,J) .EQ. 0.0) GO TO 26
000217        IF (X(1,J-1) .LE. XPRF(I) .AND. X(1,J) .GE. XPRF(I)) GO TO 20
000232     18 CONTINUE
000234        GO TO 26
000234     20 NPRF(I)=1+NPRF(I)
000237        IPRF=NPRF(I)
000240        DYDX=(Y(1,J)-Y(1,J-1))/(X(1,J)-X(1,J-1))
000246        DTDX=(T(1,J)-T(1,J-1))/(X(1,J)-X(1,J-1))
000253        CALL COAVE(XAVE,YAVE,PAVE,TAVE,1,J-1,1,J)
000264        CALL COEFF(XAVE,YAVE,PAVE,TAVE,C,GAMP)
000270        D(3)=1.0/(C(3)*C(5)*C(7))
000274        D(5)=FDIM*C(9)*C(11)/(C(13)*C(15))
000300        YPRFL=Y(1,J-1)+DYDX*(XPRF(I)-X(1,J-1))
000307        TPRF=T(1,J-1)+DTDX*(XPRF(I)-X(1,J-1))
000314        PPRF=(D(3)*P(1,J-1)-(TPRF-T(1,J-1))-D(5)*(YPRFL-Y(1,J-1)))/D(3)
000326        YPRF(I,IPRF)=(YPRFL-YWALL(I,1))/(YWALL(I,2)-YWALL(I,1))
000336        QPRF(I,IPRF)=FUNM(GAMP,PPRF)
000345        GO TO 26
000346     22 IF (ICONE .GT. 2) GO TO 24
000352        XCONE(1)=XSONIC(NDATA)
000354        YCONE(1)=YSONIC(NDATA)
000355        PCONE(1)=PSONIC(NDATA)
000357        TCONE(1)=TSONIC(NDATA)
000360        IF (XPRF(I) .LT. XCONE(1)) GO TO 26
000363     24 NPRF(I)=1+NPRF(I)
000366        IPRF=NPRF(I)
000367        DPDX=(PCONE(ICONE)-PCONE(ICONE-1))/(XCONE(ICONE)-XCONE(ICONE-1))
000373        DTDX=(TCONE(ICONE)-TCONE(ICONE-1))/(XCONE(ICONE)-XCONE(ICONE-1))
000377        PPRF=PCONE(ICONE-1)+DPDX*(XPRF(I)-XCONE(ICONE-1))
```

```
000404          TPRF=TCONE(ICONE-1)+DTDX*(XPRF(1)-XCONE(ICONE-1))
000410          YPRF(I,IPRF)=0.0
000413          WPRF(I,IPRF)=FUNM(GAMP,PPRF)
000422       26 CONTINUE
000425          RETURN
000426          END
```

```
                SUBROUTINE PERF
          C
          C     CALCULATION OF PERTINENT EJECTOR PARAMERERS
          C
000002          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1        PHS(100),ASASS(100),DASDX(100),ISLP
000002          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000002          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1        XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2        PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3        SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002          COMMON/BNLYR/XSUM(100),YSUM(100),DELSHD(100),THETAS(100),
               1        CFSHD(100),XCNE(100),YCNE(100),DELCNE(100),THETAC(100),
               2        CFCNE(100),REYPRM,POP,AOP,VOP,REYSEC,POS,AOS,VOS,PEX,XSCALE
000002          FUNA(G,AM)=((G+1.0)/2.0)**(-(G+1.0)/(2.0*(G-1.0)))*1.0/AM*(1.0+
               1            (G-1.0)/2.0*AM*AM)**((G+1.0)/(2.0*(G-1.0)))
000033          FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000052          FUNP(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-G/(G-1.0))
000066          FUNV(G,AM)=SQRT(AM*AM/(1.0+(G-1.0)/2.0*AM*AM))
000102          FIDEAL(G,PH)=SQRT(2.0*G*G/(G-1.0)*(2.0/(G+1.0))**((G+1.0)/(G-1.0))
               1              *(1.0-PH**((G-1.0)/G)))
000133          AVE(X1,X2)=(X1+X2)/2.0
          C
          C
          C     FORMAT STATEMENTS
          C
000141      600 FORMAT (//37X,31HNOZZLE PRESSURE RATIO, PTP/PO =F10.5)
000141      602 FORMAT (//37X,36HPRIMARY STREAM THRUST, FP/(PTP*AP) =F9.6)
000141      604 FORMAT (//37X,38HSECONDARY STREAM THRUST, FS/(PTP*AP) =F9.6)
000141      606 FORMAT (//37X,41HPRESSURE FORCE ON SHROUD, FSHD/(PTP*AP) =F10.6)
000141      608 FORMAT (//37X,39HPRESSURE FORCE ON BODY, FBDY/(PTP*AP) =F9.6)
000141      609 FORMAT (//37X,36HSKIN FRICTION DRAG, FDRAG/(PTP*AP) =F9.6)
000141      610 FORMAT (//37X,34HTOTAL STREAM THRUST, FT/(PTP*AP) =F9.6)
000141      612 FORMAT (//37X,38HGROSS STREAM THRUST, FGROSS/(PTP*AP) =F9.6)
000141      618 FORMAT (1H1,//49X,30HEJECTOR THRUST CHARACTERISTICS/49X,30H*******
               1***********************)
000141      620 FORMAT (//26X,6HPTP/P0,8X,6HFGROSS,10X,3HFIP,11X,3HFIS,11X,
               13HCVP,11X,2HCV)
000141      622 FORMAT (19X,6F14.5)
          C
          C     SUBROUTINE PERF
          C
000141          CPL=1.0
000143          IREF=ISLP
000144          CFI=0.003
000146          PRES=FUNP(GAMP,AMR)
000150          FP=PRES*(1.0+GAMP*CFL*CVL*AMR*AMR)
000156          ISLP=1
000157          CALL FIND(XSLP(1),YWALL,DYWDX,2.0)
000162          CALL AJAX(XSEC,YSEC,ALPHA,AREF,DADX)
000166          IF (REYPRM .EQ. 0.0) YSUM(1)=YWALL
000171          YWALL=YWALL-DELSHD(1)
000173          ASEC=(YWALL*YWALL**FDIM-YSLP(1)*YSLP(1)**FDIM)/APRIM
000205          IF (ASSAPS .GT. 0.0) ASASS(1)=ASEC/APRIM*APREF/ASSAPS
```

```
000211          IF (WTFL .GT. 0.0) CALL ASTAR(AMS(1),ASASS(1),GAMS)
000215          PHP(1)=PTS(NITER)*FUNP(GAMS,AMS(1))
000222          ANGSEC=AVE(THETA(1),ATAN(DYWDX))
000226          AMSEC=AMS(1)*COS(ANGSEC)
000231          FS=PHP(1)*(1.0+GAMS*AMSEC*AMSEC)*CPL*ASEC
000236          FSHD=0.0
000237          FBDY=0.0
000240          FDRAG=0.0
000241          DO 10 I=2,IREF
000242          IF (REYPRM .EQ. 0.0) CALL FIND(XSLP(I),YSUM(I),DYWDX,2.0)
000246          XAVE=AVE(XSLP(I),XSLP(I-1))
000251          YAVE=AVE(YSUM(I),YSUM(I-1))
000254          IF (XSLP(I) .LE. XSEC) PAVE=AVE(PHP(1),PHP(2))
000261          IF (XSLP(I) .GT. XSEC) PAVE=AVE(PHP(I),PHP(I-1))
000267          CFAVE=AVE(CFSHD(I),CFSHD(I-1))
000272          AMAVE=AVE(AMS(I),AMS(I-1))
000275          AMSQ=AMAVE*AMAVE
000276          DX=XSLP(I)-XSLP(I-1)
000300          DY=YSUM(I)-YSUM(I-1)
000302          DA=(YSUM(I)*YSUM(I)**FDIM-YSUM(I-1)*YSUM(I-1))/APRIM
000312          DS=YAVE**FDIM*SQRT(DX*DX+DY*DY)/APRIM
000324          PDA=PAVE*DA
000326          DRAG=CFAVE*GAMS*PAVE*AMSQ*DS
000332          FSHD=FSHD+PDA
000334          FDRAG=FDRAG+DRAG
000336       10 CONTINUE
000340       12 IF (NBDY .EQ. 0) GO TO 16
000341          DO 14 I=2,ICONE
000343          YAVE=AVE(YCONE(I),YCONE(I-1))
000346          PAVE=AVE(PCONE(I),PCONE(I-1))
000351          AMAVE=FUNM(GAMP,PAVE)
000355          CFAVE=AVE(CFCNE(I),CFCNE(I-1))
000360          AMSQ=AMAVE*AMAVE
000361          DX=XCONE(I-1)-XCONE(I)
000364          DY=YCONE(I-1)-YCONE(I)
000366          DA=(YCONE(I-1)*YCONE(I-1)**FDIM-YCONE(I)*YCONE(I)**FDIM)/APRIM
000400          DS=YAVE**FDIM*SQRT(DX*DX+DY*DY)/APRIM
000412          PDA=PAVE*DA
000414          DRAG=CFAVE*GAMP*PAVE*AMSQ*DS
000420          FBDY=FBDY+PDA
000422          FDRAG=FDRAG+DRAG
000424       14 CONTINUE
000426       16 FTOTAL=FS+FP+FSHD+FBDY-FDRAG
000433          CALL FIND(END,YP,DYPDX,2.0)
000436          AEXIT=YP*YP**FDIM/APRIM
000444          FGROSS=FTOTAL-PHP(IREF)*AEXIT
000447          HPP0=1.0/PHP(IREF)
000451          WRITE (6,600) HPP0
000457          WRITE (6,602) FP
000465          WRITE (6,604) FS
000473          WRITE (6,606) FSHD
000501          WRITE (6,608) FBDY
000507          WRITE (6,609) FDRAG
000515          WRITE (6,610) FTOTAL
000523          WRITE (6,612) FGROSS
000531          WRITE (6,618)
000535          WRITE (6,620)
000541          HPP0=1.0
000543          DO 18 I=2,40
000544          HPP0=1.0+HPP0
000546          P0HP=1.0/HPP0
000547          P0HS=P0HP/PTS(NITER)
000551          FIP=CFL*FIDEAL(GAMP,P0HP)/APREF
000557          IF (P0HS .GE. 1.0) FIS=0.0
000562          IF (P0HS .LT. 1.0) FIS=PTS(NITER)*ASSAPS*FIDEAL(GAMS,P0HS)/APREF
000573          FGROSS=FTOTAL-P0HP*AEXIT
000576          IF (FGROSS .LT. 0.0) GO TO 18
000600          CVI=FGROSS/(FIP+FIS)
000602          CVP=FGROSS/FIP
000603          WRITE (6,622) HPP0,FGROSS,FIP,FIS,CVP,CVI
000623       18 CONTINUE
000625          RETURN
000626          END
```

```
              SUBROUTINE OUTSLP
        C
        C     WRITE-OUT OF PERTINENT INFORMATION IN SECONDARY FLOW FIELD
        C
000002        COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002        COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
             1       PHS(100),ASASS(100),DASDX(100),ISLP
000002        COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000002        COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002        COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002        COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002        COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002        COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
             1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
             2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
             3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002        COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002        COMMON/OUTPN/IPNCH,IPRNT,ICOMP
000002        FUNA(G,AM)=((G+1.0)/2.0)**(-(G+1.0)/(2.0*(G-1.0)))*1.0/AM*(1.0+
             1           (G-1.0)/2.0*AM*AM)**((G+1.0)/(2.0*(G-1.0)))
000033        FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
        C
        C     FORMAT STATEMENTS
        C
000052    600 FORMAT (1H1,//28X,18A4,//)
000052    602 FORMAT (25X,6HWTFL =F9.6,5X,7HWLEAK =F9.6,5X,9HPTS/PTP =F9.6,5X,
             18HAS/AS* =F9.6)
000052    604 FORMAT (//,20X,4HXSLP,8X,4HYSLP,8X,3HAMP,8X,5HTHETA,7X,5HP/PTP,8X,
             1       3HAMS,8X,5HP/PTS,7X,6HAS/AS*)
000052    606 FORMAT (13X,8F12.5)
000052    608 FORMAT (//37X,45HSECONDARY CORRECTED WEIGHT FLOW RATIO, WTFL =
             1                    F9.6)
000052    610 FORMAT (//37X,41HSECONDARY TOTAL PRESSURE RATIO, PTS/PTP =F9.6)
000052    612 FORMAT (//37X,40HSECONDARY CRITICAL AREA RATIO, AS*/AP* =F9.6)
000052    614 FORMAT (//37X,38HPRIMARY NOZZLE FLOW COEFFICIENT, CFL =F8.5)
000052    616 FORMAT (//37X,42HPRIMARY NOZZLE VELOCITY COEFFICIENT, CVL =F8.5)
        C
        C     SUBROUTINE OUTSLP
        C
000052        IF (XSLP(ISLP) .LE. END) GO TO 8
000055        RATIO=PHP(ISLP)/PHS(ISLP)
000057        DX=XSLP(ISLP)-XSLP(ISLP-1)
000061        DY=YSLP(ISLP)-YSLP(ISLP-1)
000063        DP=PHP(ISLP)-PHP(ISLP-1)
000065        DT=THETA(ISLP)-THETA(ISLP-1)
000067        XSLP(ISLP)=END
000071        YSLP(ISLP)=YSLP(ISLP-1)+(DY/DX)*(XSLP(ISLP)-XSLP(ISLP-1))
000077        PHP(ISLP)=PHP(ISLP-1)+(DP/DX)*(XSLP(ISLP)-XSLP(ISLP-1))
000105        THETA(ISLP)=THETA(ISLP-1)+(DT/DX)*(XSLP(ISLP)-XSLP(ISLP-1))
000113        AMP(ISLP)=FUNM(GAMP,PHP(ISLP))
000121        PHS(ISLP)=PHP(ISLP)/RATIO
000123        AMS(ISLP)=FUNM(GAMS,PHS(ISLP))
000131        ASASS(ISLP)=FUNA(GAMS,AMS(ISLP))
000135        CALL SHLYR(DDSDX)
000137      8 WRITE (6,600) (TITLE(K), K=1,18)
000151        WRITE (6,602) WTFL,WLEAK(NITER),PTS(NITER),AREA(NITER)
000165        WRITE (6,604)
000171        DO 12 I=1,ISLP
000173        IF (I .LE. 45 .OR. I .GE. 47) GO TO 10
000203        WRITE (6,600) (TITLE(K), K=1,18)
000214        WRITE (6,602) WTFL,WLEAK(NITER),PTS(NITER),AREA(NITER)
000230        WRITE (6,604)
000234     10 ANGLE=CONVA*THETA(I)
000237        WRITE (6,606) XSLP(I),YSLP(I),AMP(I),ANGLE,PHP(I),AMS(I),
             1              PHS(I),ASASS(I)
000262     12 CONTINUE
000265        IF (CHOKE .EQ. -1.0) GO TO 20
000267        IF (ICOMP .EQ. 2) CALL OUTPRF
000272        CALL OUTLYR
000273     18 IF (NBDY .GT. 0) CALL OUTCNE
000276        CALL OUTSNP
000277        HSHP=PTS(NITER+1)
```

```
000301          ASSAPS=FUNG*WTFL/PTS(NITER+1)
000304          WRITE (6,600) (TITLE(K), K=1,18)
000315          WRITE (6,608) WTFL
000323          WRITE (6,610) PTS(NITER+1)
000331          WRITE (6,612) ASSAPS
000337          WRITE (6,614) CFL
000345          WRITE (6,616) CVL
000353          PTS(NITER)=PTS(NITER+1)
000355          IF (NSHD .EQ. 1) CALL PERF
000360          IF (NSHD .GT. 1 .AND. XSLP(ISLP) .EQ. END) CALL PERF
000374       20 RETURN
000375          END




                SUBROUTINE OUTLYR
          C
          C     WRITE-OUT OF PERTINENT BOUNDARY LAYER INFORMATION
          C
000002          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1       PHS(100),ASASS(100),DASDX(100),ISLP
000002          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000002          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2       PAMB,YRATIO,PI,CONVA,CONVP,FDIM,NDATA,NSTOP,
               3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002          COMMON/BNLYR/XSUM(100),YSUM(100),DELSHD(100),THETAS(100),
               1       CFSHD(100),XCNE(100),YCNE(100),DELCNE(100),THETAC(100),
               2       CFCNE(100),REYPRM,POP,AOP,VOP,REYSEC,POS,AOS,VOS,PEX,XSCALF
000002          COMMON/OUTPN/IPNCH,IPRNT,ICOMP
000002          DIMENSION ZSLP(100),ZCONE(100),AMCNE(100)
000002          FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000022      600 FORMAT (1H1,//28X,18A4,//)
000022      602 FORMAT (22X,6HWTFL =F9.6,5X,9HPTS/PTS =F9.6,5X,8HREYPRM =E10.3,
               1       5X,8HREYSEC =E10.3)
000022      604 FORMAT (//32X,4HXSHD,8X,4HYSHD,8X,3HAMS,8X,6HDELSHD,6X,6HTHETAS,
               1       6X,5HCFSHD)
000022      606 FORMAT (25X,6F12.5)
000022      608 FORMAT (//32X,4HXCNE,8X,4HYCNE,8X,3HAMC,8X,6HDELCNE,6X,6HTHETAC,
               1       6X,5HCFCNE)
000022      610 FORMAT (1H0)
          C
          C     SUBROUTINE OUTLYR
          C
000022          IF (REYPRM .EQ. 0.0) GO TO 18
000023          WRITE (6,600) (TITLE(K), K=1,18)
000035          WRITE (6,602) WTFL,PTS(NITER),REYPRM,REYSEC
000051          WRITE (6,604)
000055          DO 10 I=1,ISLP
000057          ZSLP(I)=XSLP(I)+XSUM(I)/XSCALE
000063          WRITE (6,606) XSLP(I),YSUM(I),AMS(I),DELSHD(I),THETAS(I),CFSHD(I)
000102          IF (I .NE. 45) GO TO 10
000104          WRITE (6,600) (TITLE(K), K=1,12)
000116          WRITE (6,602) WTFL,PTS(NITER),REYPRM,REYSEC
000132          WRITE (6,604)
000136       10 CONTINUE
000141          IF (NBDY .EQ. 0) GO TO 14
000142          ISTOP=ICONE-1
000144          WRITE (6,600) (TITLE(K), K=1,18)
000155          WRITE (6,602) WTFL,PTS(NITER),REYPRM,REYSEC
000171          WRITE (6,608)
000175          DO 12 I=1,ISTOP
000177          ZCONE(I)=XCONE(I)+XCNE(I)/XSCALE
```

```
000203          AMCNE(I)=FUNM(GAMP,PCONE(I))
000210          WRITE (6,606) XCONE(I),YCONE(I),AMCNE(I),DELCNE(I),THETAC(I),
               1              CFCNE(I)
000230          IF (I .NE. 46) GO TO 12
000232          WRITE (6,600) (TITLE(K), K=1,18)
000244          WRITE (6,602) WTFL,PTS(NITER),REYPRM,REYSEC
000260          WRITE (6,608)
000264       12 CONTINUE
000267       14 IF (IPNCH .EQ. 0) GO TO 18
          C
          C     PUNCH CARDS FOR SASMAN CRESCI TURBULENT BOUNDARY LAYER PROGRAM
          C
000270      500 FORMAT (9A6)
000270      502 FORMAT(6F12.6)
000270      504 FORMAT (5I6)
000270      506 FORMAT (2F10.6,10X,5F10.6)
000270          PR=0.705
000272          RECOV=0.950
000273          DIM=(2.0+FDIM)
000275          SCALE=DPRIM/2.0
000277          DIV=4.0
000300          IPROF=0
000301          ICARD=0
000302          IPLOT=1
000303          IF (IPNCH .EQ. 2) GO TO 16
000305          POS=POS/144.0
000307          THETA1=SCALE*THETAS(1)
000311          H1=DELSHD(1)/THETAS(1)
000312          PUNCH 500, (TITLE(K), K=1,9)
000324          PUNCH 502, POS,TOS,AMHU0,GAMS,PR,RECOV
000344          PUNCH 502, DIM,SCALE,AMS(1),THETA1,H1,DIV
000364          PUNCH 504, ISLP,ISLP,IPROF,ICARD,IPLOT
000402          PUNCH 506, (ZSLP(I),YSUM(I),AMS(I),PHS(I),DELSHD(I),THETAS(I),
               1           CFSHD(I), I=1,ISLP)
000431       16 IF (IPNCH .EQ. 1 .OR. NBDY .EQ. 0) GO TO 18
000440          POP=POP/144.0
000442          THETA1=SCALE*THETAC(1)
000444          H1=DELCNE(1)/THETAC(1)
000445          PUNCH 500, (TITLE(K), K=1,9)
000457          PUNCH 502, POP,TOP,AMHU0,GAMP,PR,RECOV
000477          PUNCH 502, DIM,SCALE,AMCNE(1),THETA1,H1,DIV
000517          PUNCH 504, ISTOP,ISTOP,IPROF,ICARD,IPLOT
000535          PUNCH 506, (ZCONE(I),YCONE(I),AMCNE(I),PCONE(I),DELCNE(I),
               1           THETAC(I),CFCNE(I), I=1,ISTOP)
000564       18 RETURN
000565          END
```

```
                SUBROUTINE OUTSNP
          C
          C     WRITE-OUT OF CONDITIONS ALONG SONIC LINE
          C
000002          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1       PHS(100),ASASS(100),DASDX(100),ISLP
000002          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000002          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002          COMMON/BNLYR/XSUM(100),YSUM(100),DELSHD(100),THETAS(100),
               1       CFSHD(100),XCNE(100),YCNE(100),DELCNE(100),THETAC(100),
               2       CFCNE(100),REYPRM,POP,AOP,VOP,REYSEC,POS,AOS,VOS,PEX,XSCALE
```

```
000002          REAL K1,K2
000002          AVE(X1,X2)=(X1+X2)/2.0
000011      600 FORMAT (1H1,//33X,18A4,//)
000011      604 FORMAT (35X,8HXSONIC =F9.5,3X,8HYSONIC =F8.5,8X,8HTSONIC =F10.5)
000011          PEX=7.0
000013          CFL=0.0
000014          CVL=0.0
000015          APRM=0.0
000016          PAVE=PSONIC(1)
000017          WRITE (6,600) (TITLE(K), K=1,18)
000031          DO 10 I=1,ISONIC
000033          ALPHA=CONVA*TSONIC(I)
000035          WRITE (6,604) XSONIC(I),YSONIC(I),ALPHA
000047          IF (I .EQ. 1) GO TO 10
000051          YAVE=AVE(YSONIC(I),YSONIC(I-1))
000054          TAVE=AVE(TSONIC(I),TSONIC(I-1))
000057          DX=XSONIC(I-1)-XSONIC(I)
000061          DY=YSONIC(I-1)-YSONIC(I)
000063          APRM=APRM+(1.0+FDIM)*YAVE**FDIM*DY
000073          CVL=CVL+(1.0+FDIM)*YAVE**FDIM*COS(TAVE)*(COS(TAVE)*DY-
               1         SIN(TAVE)*DX)
000117          CFL=CFL+(1.0+FDIM)*YAVE**FDIM*(COS(TAVE)*DY-SIN(TAVE)*DX)
000137       10 CONTINUE
000142          CVL=CVL/CFL
000143          CFL=CFL/APRM
000145          IF (REYPRM .EQ. 0.0) GO TO 12
000146          K1=4.0*0.370/(1.0+PEX)/COS(ABS(ANGR))
000156          K2=4.0*0.370*PEX/((1.0+PEX)*(2.0+PEX))
000164          CFL=(1.0-K1*REYPRM**(-0.20))*CFL
000172          CVL=(1.0-K2*REYPRM**(-0.20))*CVL
000200       12 RETURN
000201          END
```

```
                SUBROUTINE OUTCNE
          C
          C     WRITE-OUT OF PERTINENT INFORMATION ALONG THE EJECTOR CENTERBODY
          C
000002          COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002          COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
               1         PHS(100),ASASS(100),DASDX(100),ISLP
000002          COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000002          COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002          COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002          COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002          COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002          COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
               1         XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
               2         PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
               3         SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002          COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002          REAL MACH
000002          FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000022      600 FORMAT (1H1,//28X,18A4,//)
000022      602 FORMAT (21X,4HXC =F8.5,4X,4HYC =F8.5,4X,10HMACH NO. =F8.5,4X,7HTHE
               1TA =F8.5,4X,6HP/HP =F8.5)
000022          LCHK=45
000023          LINE=0
000024          WRITE (6,600) (TITLE(K), K=1,18)
000036          DO 10 I=1,ICONE
000040          MACH=FUNM(GAMP,PCONE(I))
000045          WRITE (6,602) XCONE(I),YCONE(I),MACH,TCONE(I),PCONE(I)
000062          LINE=1+LINE
000064          IF (LINE .LE. LCHK) GO TO 10
000066          IF (I .LT. ICONE) WRITE (6,600)  (TITLE(K), K=1,12)
000102       10 CONTINUE
000105          RETURN
000106          END
```

```
              SUBROUTINE OUTFLD(I)
        C
        C     WRITE-OUT OF PERTINENT INFORMATION IN PRIMARY FLOW FIELD
        C
000003        COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000003        COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
             1       PHS(100),ASASS(100),DASDX(100),ISLP
000003        COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000003        COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000003        COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000003        COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000003        COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000003        COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
             1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
             2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
             3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000003        COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000003        COMMON/CPLOT/IPLOT,XSTART,YSTART,XSPAN,YSPAN,SCALE,SPAN,AXIS,
             1     XORGN,YORGN,XSHFT,YSHFT,KKK(14),PP(14),XDOWN(100),YACROS(100)
000003        REAL MACH
000003        FUNM(G,PH)=SQRT(2.0/(G-1.0)*(PH**(-(G-1.0)/G)-1.0))
000024    600 FORMAT (1H1,//48X,36HCONDITIONS IN THE PRIMARY FLOW FIELD,//)
000024    602 FORMAT (24X,4HXP =F9.5,4X,4HYP =F8.5,4X,10HMACH NO. =F8.5,4X,6HP/H
             1P =F8.5,4X,7HTHETA =F9.5)
000024        IF (IPLOT .EQ. 1) GO TO 12
000026        WRITE (6,600)
000032        DO 10 K=1,100
000035        IF (P(I,K) .EQ. 0.0) GO TO 12
000037        MACH=FUNM(GAMP,P(I,K))
000044        WRITE (6,602) X(I,K),Y(I,K),MACH,P(I,K),T(I,K)
000072     10 CONTINUE
000075     12 IF (IPLOT .EQ. 1) CALL PLOTF(I)
000101        RETURN
000102        END
```

```
              SUBROUTINE OUTPRF
        C
        C     WRITE-OUT OF MACH NUMBER PROFILES
        C
000002        COMMON X(2,100),Y(2,100),P(2,100),T(2,100)
000002        COMMON XSLP(100),YSLP(100),AMP(100),THETA(100),PHP(100),AMS(100),
             1       PHS(100),ASASS(100),DASDX(100),ISLP
000002        COMMON XIS(21,26),YIS(21,26),W(21),TAU(26),NSONIC,NANGLE
000002        COMMON XSONIC(26),YSONIC(26),PSONIC(26),TSONIC(26),ISONIC
000002        COMMON XCONE(100),YCONE(100),PCONE(100),TCONE(100),ICONE
000002        COMMON XSHD(100),YSHD(100),DYSDX(100),NSHD
000002        COMMON XBDY(100),YBDY(100),DYBDX(100),NBDY
000002        COMMON WTFL,HSHP,TOS,TOP,GAMS,GAMP,FUNG,AMR,ANGR,APREF,ASSAPS,
             1       XPRIM,YPRIM,APRIM,DPRIM,CFL,CVL,CONA,DBDY,DSHD,END,
             2       PAMB,YRATIO,PI,CONVA,CONVR,FDIM,NDATA,NSTOP,
             3       SOLVE,CHOKE,CHANGE,CHARGE,TYPE,POINT,STAG
000002        COMMON PTS(25),AREA(25),WLEAK(25),TITLE(18),NITER,TRY
000002        COMMON/PRFLR/XPRF(6),YPRF(6,25),QPRF(6,25),NPRF(6),MAX,NMAX
000002        COMMON/OUTPN/IPNCH,KPNCH,ICOMP
000002        FUNP(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-G/(G-1.0))
000017        FUNT(G,AM)=(1.0+(G-1.0)/2.0*AM*AM)**(-1)
000030    600 FORMAT (1H1,//27X,12A6,///)
000030    602 FORMAT (19X,4HXP =F9.5,4X,4HYP =F8.5,4X,10HMACH NO. =F8.5,4X,7HP/P
             1OP =F8.5,4X,7HT/TOP =F8.5)
000030    604 FORMAT (/1H0)
000030    700 FORMAT (12A6)
000030    702 FORMAT (36HEJECTOR MACH NO. FLOW FIELD AT X/R =F6.3)
000030    704 FORMAT (I3)
000030    706 FORMAT (8F10.5)
000030        ISKIP=0
000031        WRITE (6,600) (TITLE(J), J=1,12)
000043        IF (KPNCH .GT. 0) PUNCH 700, (TITLE(I), I=1,12)
000056        DO 14 I=1,MAX
```

```
000060          IF (NPRF(I) .EQ. 0) GO TO 14
000061          KPRF=NPRF(I)
000063          ISKIP=1+ISKIP
000064          IF (ISKIP .EQ. 2) WRITE (6,604)
000071          DO 10 K=1,KPRF
000073          PRES=FUNP(GAMP,QPRF(I,K))
000077          TEMP=FUNT(GAMP,QPRF(I,K))
000104          WRITE (6,602) XPRF(I),YPRF(I,K),QPRF(I,K),PRES,TEMP
000125       10 CONTINUE
000130          IF (ISKIP .EQ. 1) GO TO 12
000132          IF (I .EQ. MAX) GO TO 12
000134          ISKIP=0
000134          WRITE (6,600) (TITLE(J), J=1,12)
000146       12 IF (KPNCH .EQ. 0) GO TO 14
000147          PUNCH 702, XPRF(I)
000155          PUNCH 704, KPRF
000163          PUNCH 706, (YPRF(I,K),QPRF(I,K), K=1,KPRF)
000203       14 CONTINUE
000206          RETURN
000207          END
```

```
                SUBROUTINE PLOTC
          C
          C     CALCOMP PLOT OF SHROUD AND CENTERBODY CONTOUR
          C
000002       20 RETURN
000003          END
```

```
                SUBROUTINE PLOTF(I)
          C
          C     CALCOMP PLOT OF CHARACTERISTIC FIELD
          C
000003          RETURN
000004          END
```

```
                SUBROUTINE PLOTL
          C
          C     CALCOMP PLOT OF PRIMARY LIP (CONCLUSION OF PLOTTING SEQUENCE)
          C
000002          RETURN
000003          END
```

REFERENCES

1. Kochendorfer, Fred D.; and Rousso, Morris D.: Performance Characteristics of Aircraft Cooling Ejectors Having Short Cylindrical Shrouds. NACA RM E51E01, 1951.
2. Fabri, T.; and Paulon, J.: Theory and Experiments on Supersonic Air-to-Air Ejectors. NACA TN 1410, 1958.
3. Bernstein, Arthur; Hevenor, Charles; and Heiser, William: Compound-Compressible Nozzle Flow. Paper 66-663, AIAA, June 1966.
4. Hardy, J. M.; and Lacombe, H.: Supersonic Bypass Nozzles - Computing Methods. Rev. Francaise de Mécanique, 4th qtr., 1967, pp. 49-59.
5. Chow, W. L.; and Addy, A. L.: Interaction Between Primary and Secondary Streams of Supersonic Ejector Systems and Their Performance Characteristics. AIAA J., vol. 2, no. 4, Apr. 1964, pp. 686-695.
6. Anon.: Users Manual for the General Ejector Nozzle Deck (Deck Ir). Rep. PWA-3465, Suppl. A, Pratt & Whitney Aircraft, 1968.
7. Korst, H. H.; and Chow, W. L.: Non-isoenergetic Turbulent ($Pr_t = 1$) Jet Mixing Between Two Compressible Streams at Constant Pressure. NASA CR-419, 1966.
8. Beheim, Milton A.; Anderson, Bernhard H.; Clark, John S.; Corson, Blake W., Jr.; Stitt, Leonard E.; and Wilcox, Fred A.: Supersonic Exhaust Nozzles. Aircraft Propulsion. NASA SP-259, 1971, pp. 233-282.
9. Brown, Eugene F.: Compressible Flow Through Convergent Conical Nozzles with Emphasis on the Transonic Region. Ph.D. Thesis, Univ. of Illinois, 1968.
10. Anderson, B. H.: Factors which Influence the Analysis and Design of Ejector Nozzles. Paper 72-46, AIAA, Jan. 1972.
11. Anderson, Bernhard H.: Assessment of an Analytical Procedure for Predicting Supersonic Ejector Nozzle Performance. NASA TN D-7601, 1974.
12. Hartree, Douglas R.: Numerical Analysis. Second ed., Clarendon Press, 1958.
13. Stratford, B. S.; and Beavers, G. S.: The Calculation of Compressible Turbulent Boundary Layer in an Arbitrary Pressure Gradient - A Correlation of Certain Previous Methods. Rep. R&M-3207, Aeronautical Research Council, Gt. Britain, 1961.
14. Schlichting, Hermann (J. Kestin, trans.): Boundary Layer Theory. McGraw-Hill Book Co., Inc., 1955.

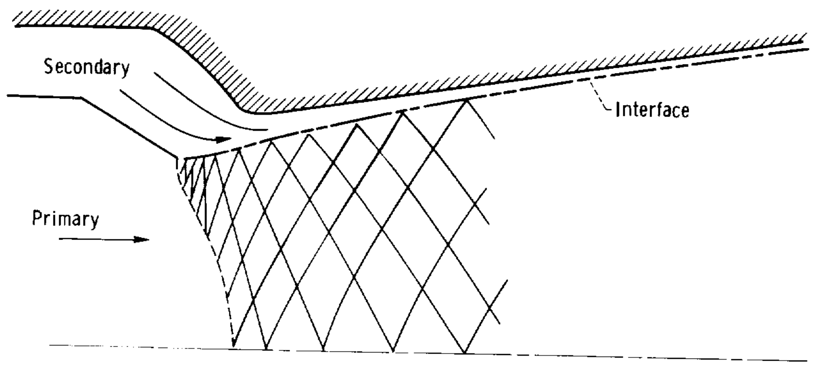


Figure 1. - Supersonic ejector system.

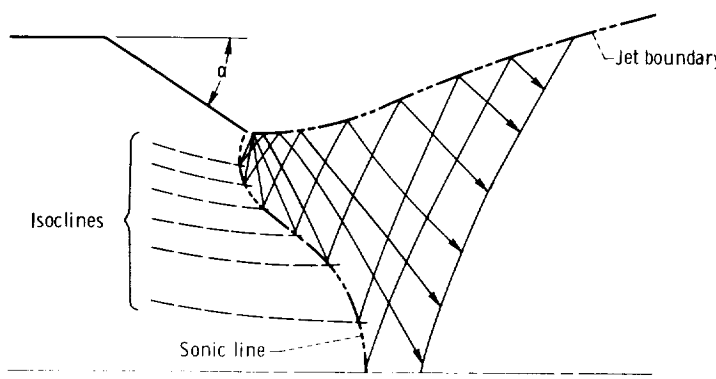


Figure 2. - Primary nozzle flow field.

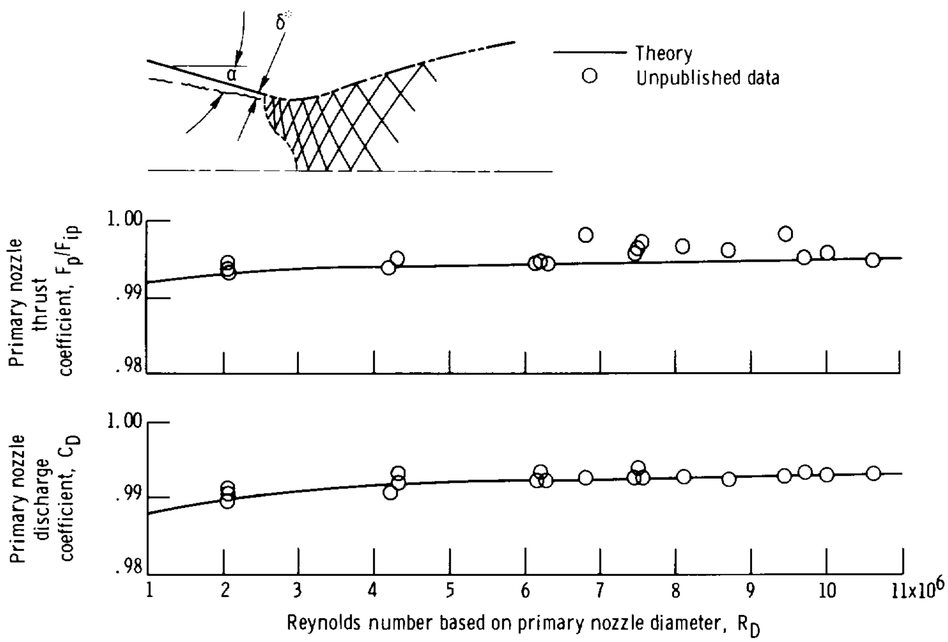
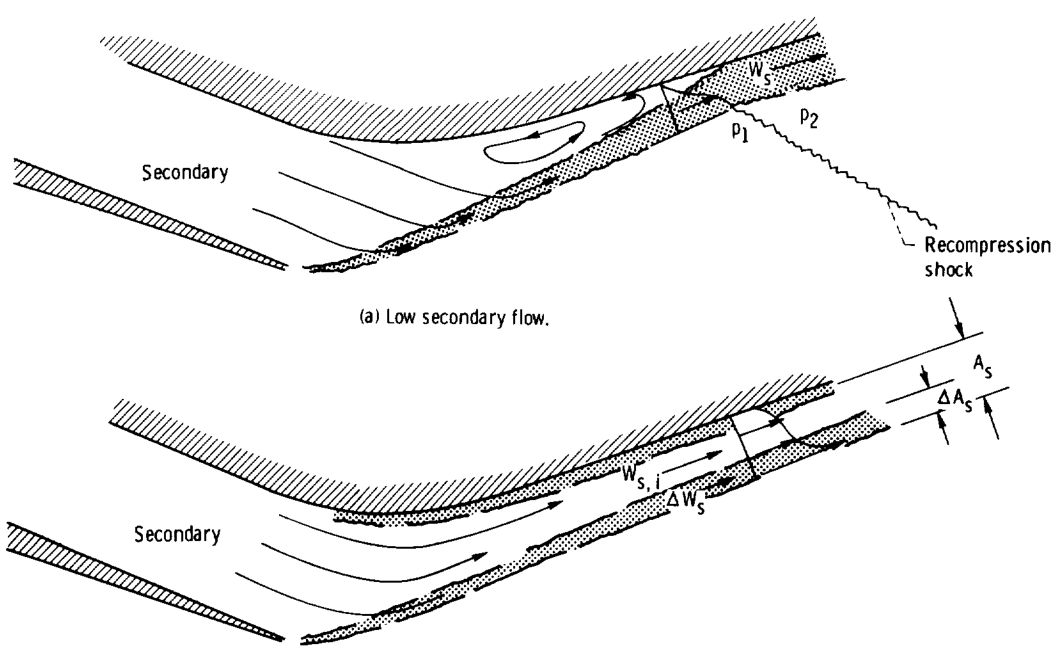


Figure 3. - Effect of Reynolds number on performance of a choked conical nozzle. Primary nozzle lip angle, α, 0; ratio of primary total pressure to reference static pressure, $P_p/p_o > 2$.



(a) Low secondary flow.

(b) High secondary flow.

Figure 4. - Secondary flow field.

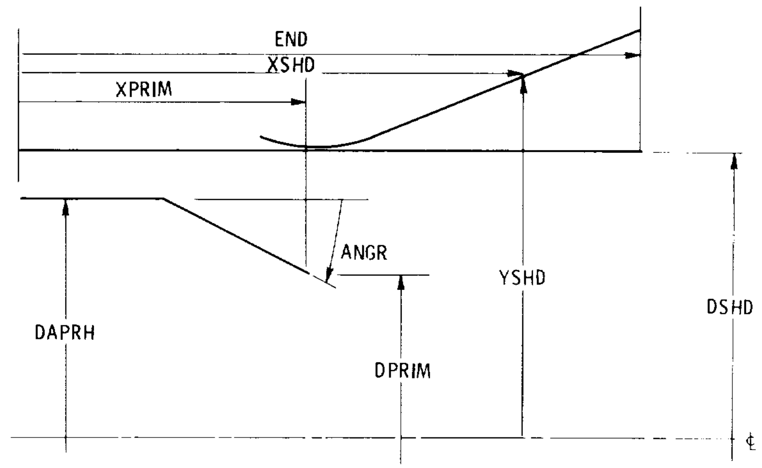


Figure 5. - Geometric input variables. YRATIO = DAPRH/DPRIM.

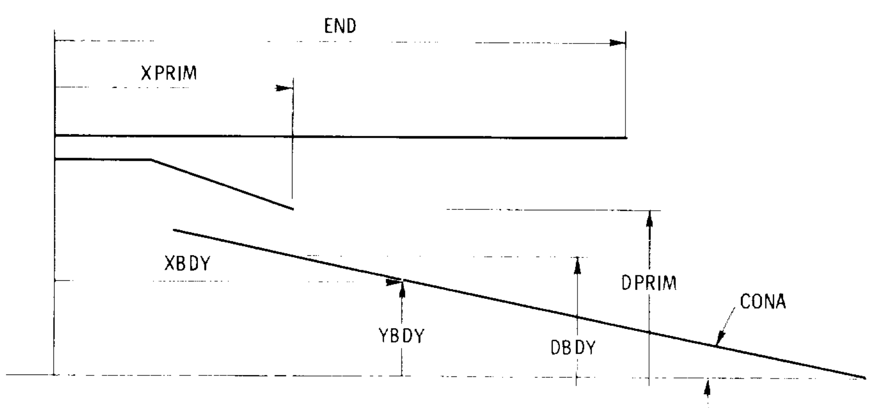


Figure 6. - Geometric input variables, plug nozzle ejector.

NASA-Langley, 1974 E-7476