CASE FILE COPY

NASA TECHNICAL NOTE

NASA TN D-4960

NASA TN D-4960

DESIGN OF SUPERSONIC INLETS BY A COMPUTER PROGRAM INCORPORATING THE METHOD OF CHARACTERISTICS

by Bernhard H. Anderson
Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JANUARY 1969

NASA TN D-4960

DESIGN OF SUPERSONIC INLETS BY A COMPUTER PROGRAM INCORPORATING THE METHOD OF CHARACTERISTICS

By Bernhard H. Anderson

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

ABSTRACT

A FORTRAN IV computer program which incorporates the method of characteristics was written to assist in the design of supersonic inlets. The computer program was written with the intention of studying many types of inlet configurations with a minimum of computer time. Particular attention was given to a discussion of a reformulation of the boundary value problem to introduce throat Mach number and flow angle as direct program input quantities. Comparison between the computer results and experimental data indicates good agreement for a number of different configurations.

DESIGN OF SUPERSONIC INLETS BY A COMPUTER PROGRAM INCORPORATING THE METHOD OF CHARACTERISTICS

by Bernhard H. Anderson

Lewis Research Center

SUMMARY

A FORTRAN IV computer program which incorporates the method of characteristics was written to assist in the design of supersonic inlets. The program is versatile so that many inlet types can be studied with a minimum of computer time.

Particular attention was given to reformulating the boundary value problem so that the engineering design parameters, that is, throat Mach number and flow angle, could be introduced as direct input quantities to the computer program. Attention was also given to a discussion of streamline tracing by a numerical integration of the mass flux from a known physical boundary to determine the opposite boundary.

Selected examples are presented and compared with experimental results to illustrate and corroborate the numerical methods used. The report also includes FORTRAN listings for the main programs and subroutines and discussions of their various functions.

INTRODUCTION

The mathematical methods for designing supersonic inlets by characteristic theory have been available for many years (ref. 1). However, the major obstacle in the design of supersonic inlets by the method of characteristics has been the great complexity of the numerical procedures required for configurations other than simple wave solutions. Thus, to facilitate the design of inlets, computational programs have been written for use on high-speed computers (refs. 2 to 4). Because of the often complex nature of the boundaries which constitute a supersonic inlet, these programs have been limited to specific configurations. In addition, computational methods currently being used for the design of high-performance inlets often necessitate extensive trial-and-error procedures to arrive at an optimum design. This results from the fact that some of the primary input data to the computer program are mathematical in significance and only indirectly related to important engineering design parameters, such as throat Mach number, flow angle, and distortion.

For these reasons, a FORTRAN IV computer program, employing the method of characteristics, was written with two objectives in mind: (1) to study a greater variety of supersonic inlet configurations and (2) to reduce the time required for trial-and-error procedures to arrive at optimum inlet design. The computer program was written with the intention of being able to construct a variety of inlet configurations by interchanging specific subroutines. In this manner, greater flexibility of choice was attained, and the time required to program a specific inlet configuration was greatly reduced. The second objective was accomplished by a reformulation of the boundary value problem for hyperbolic equations. By this reformulation of the boundary data, the engineering design quantities, such as throat Mach number and flow angle, were introduced as direct input quantities to the computer program. As a consequence of introducing the engineering parameters as input, the computer program will calculate the surface contours required to satisfy the specified throat conditions with the minimum inviscid throat distortion.

To facilitate the use of the method of characteristics in the design of supersonic inlets, numerical procedures have been derived from the general theory and adapted to high-speed computers. Particular attention was given in this study to a discussion of boundary data and ways they may be applied for more effective inlet design. In addition, attention was given to a discussion of streamline tracing by a numerical integration of the mass flux from a known physical boundary. This technique was used extensively in the computer program with great effectiveness.

To illustrate and corroborate the numerical methods, selected examples are presented and compared with experimental results. Sample problems are also included to illustrate the use of the program to achieve specific objectives. This report also presents FORTRAN listings for the main programs and subroutines, descriptions and functions of the subroutines, explanations of the input-output quantities, and general program procedures.

SYMBOLS

A	dimensionless coefficient defined by eq. (3)
B	dimensionless coefficient defined by eq. (4)
C	characteristic
$C_{d,a}$	additive drag coefficient
c	dimensionless speed of sound (see appendix A) referenced to critical speed

D	dimensionless coefficient defined by eq. (5)
E	dimensionless coefficient defined by eq. (6)
F	dimensionless coefficient defined by eq. (7)
G	dimensionless coefficient defined by eq. (7)
M	Mach number
m	mass flow
P	total pressure
p	static pressure
q	dimensionless local velocity relative to critical speed
R	coefficients for equation of boundary
r	dimensionless radius, $\sqrt{x^2 + y^2}$
u	dimensionless x-component of velocity (see appendix A) relative to critical speed
v	dimensionless y-component of velocity (see appendix A) relative to critical speed
x	dimensionless x-coordinate relative to cowl lip radius
y	dimensionless y-coordinate relative to cowl lip radius
$\alpha(x, y)$	parameter defining C_+ characteristic family (see eq. (A10))
$\beta(x, y)$	parameter defining C_- characteristic family (see eq. (A10))
β_s	shock angle
γ	ratio of specific heats
δ	deflection angle
ϵ	tangent of solid boundary
η	cotangent of conic ray half angle (see eq. (B1))
λ	Eigen values of eq. (A7)
Θ	conic ray half angle
μ	local Mach angle with respect to local flow angle
ρ	dimensionless density relative to critical conditions
ϑ	local flow angle with respect to coordinate system
$\zeta_\pm$	characteristic directions (see eq. (A15))

ψ	stream function
σ^2	parameter defined as $(\sigma - 1)/(\sigma + 1)$
τ	parameter defined by eq. (A6)

Subscripts:

c	cowl surface
cb	centerbody surface
ref	reference conditions
w	wall conditions
0	free-stream conditions

Superscript:

—	average conditions

GOVERNING EQUATIONS

To facilitate the use of the characteristic theory in the design of supersonic inlets, numerical procedures were developed with the following underlying assumptions:

(1) Steady three-dimensional flow with cylindrical symmetry

(2) Homentropic flow within shockless flow regions

Assumption (1) is not overly restrictive since a large class of two- and three-dimensional inlets of practical interest are satisfied by this condition; assumption (2) is implemented along physical boundaries by basing the calculations on the total-pressure recovery along that streamline. In general, the accuracy of results with assumption (2) was not appreciably different from isentropic calculations. In general, these errors were of the order of 0.10 to 0.50 percent in local Mach number.

Characteristic Equations

Flow with cylindrical symmetry is characterized by the conditions that all pertinent quantities depend only on the abscissa x along the axis and the distance y from the axis. The characteristic equations and compatibility conditions are found in reference 1 and derived in appendix A. They are given, respectively, by

$$\left.\begin{aligned} dy &= A\,dx \qquad C_+ \\ dy &= B\,dx \qquad C_- \end{aligned}\right\} \tag{1}$$

$$\left.\begin{array}{ll} du = D\,dv + F\,dx & C_+ \\ du = E\,dv + G\,dx & C_- \end{array}\right\} . \qquad (2)$$

where

$$A = \tan(\vartheta + \mu) \qquad (3)$$

$$B = \tan(\vartheta - \mu) \qquad (4)$$

$$D = -\tan(\vartheta - \mu) \qquad (5)$$

$$E = -\tan(\vartheta + \mu) \qquad (6)$$

$$F = G = \frac{c^2}{u^2 - c^2}\,\frac{v}{y} \qquad (7)$$

In addition to the above equations, Bernoulli's Law gives the following relation:

$$\sigma^2(u^2 + v^2) + (1 - \sigma^2)c^2 = 1 \qquad (8)$$

where

$$\sigma^2 = \frac{\gamma - 1}{\gamma + 1} \qquad (9)$$

The set of equations (1), (2), and (8) have been made dimensionless by dividing the spatial coordinates by the cowl lip radius and the velocity coordinates by the critical speed. The equations for two-dimensional flow are obtained by setting $F = G = 0$.

Conical Flow Field Equations

Further mathematical simplifications are possible for the case of axisymmetric flow if the flow field considered is that produced by a cone with sufficiently small half angle that the shock front is attached. Under these conditions, the velocity components u, v and speed of sound c depend only on the angle between the axis of symmetry and the family of concentric cones between the obstacle cone and shock wave. When the velocity components u and v are introduced as the independent and dependent variables (see appendix B), respectively, the governing equation becomes

$$vv_{uu} = \left(1 + v_u^2\right) - \frac{(1 - \sigma^2)\,(u + vv_u)^2}{1 - \sigma^2(u^2 + v^2)} \tag{10}$$

where the subscripts indicate differentiation. Along the cone surface, the boundary condition is specified by the relation

$$v_u = \frac{-u}{v} \tag{11}$$

The conditions to be satisfied by the velocity immediately behind the conical shock are specified by the oblique shock equations

$$u = (1 - \sigma^2)q_0 \cos^2 \beta_s + \frac{1}{q_0} \tag{12}$$

$$v = (q_0 - u) \cot \beta_s \tag{13}$$

$$v_u = \frac{v}{u - q_0} \tag{14}$$

where q_0 is the dimensionless free-stream velocity and β_s is the shock-wave angle. Equations (10) to (14) are derived in reference 1 and listed in appendix B.

Oblique Shock-Wave Relations

The standard oblique shock relation used was obtained from reference 5 and is given by

$$\sin^6 \beta_s + b \sin^4 \beta_s + c \sin^2 \beta_s + d = 0 \tag{15}$$

where

$$b = -\frac{M^2 + 2}{M^2} - \gamma \sin^2 \delta \tag{16}$$

$$c = \frac{2M^2 + 1}{M^4} + \left[\left(\frac{\gamma + 1}{2}\right)^2 + \frac{\gamma - 1}{M^2}\right] \sin^2 \delta \tag{17}$$

$$d = -\frac{\cos^2 \delta}{M^4} \tag{18}$$

M is the Mach number upstream of the shock wave, and δ is the deflection angle.

FORMULATION OF BOUNDARY DATA

The type of data which must be prescribed to have a well-set problem is fundamental to establishing a numerical solution to the characteristic equations. In the design of supersonic inlets, a common situation which arises includes a known physical boundary along with an initial datum line which can be either noncharacteristic (such as a shock wave) or one member of the two-family set of characteristics (fig. 1). The solution of the flow field can then be established in the triangular domain ABC in the x,y plane.

A more general way of formulating boundary requirements is to consider the flow field to be constrained between two boundaries (fig. 2), one of which may not be defined. In order to establish a well-set problem, data must be prescribed along the initial datum line and a pair of boundary data must be provided. The possible combinations of boundary data include

(1) Prescribing the two boundary curves, $y = y_{cb}(x)$ and $y = y_c(x)$, and solving for the flow field in the domain ABCD and conditions along each of the bounding surfaces

(2) Prescribing one boundary surface, $y = y_{cb}(x)$, and the corresponding velocity distribution along the same surface, $q = q_{cb}(x)$ (Under these conditions, the boundary value problem is to establish the flow field and the necessary contour of the second surface, $y = y_c(x)$, which are consistent with the initial data and prescribed boundary information.)

(3) Prescribing the velocity distribution for each of the surfaces, for example, $q = q_{cb}(x)$ and $q = q_c(x)$ (Under these conditions, only the magnitude of the velocity is known on each of the two boundaries. The solution must thus include establishing the unknown surfaces and flow field which satisfy the prescribed initial and boundary data.)

For the purposes of this report, the three types of boundary data described are termed

(1) Fixed-boundary problem (Case 1)

(2) Free-boundary problem (Case 2)
(3) Doubly-free-boundary problem (Case 3)

Fixed-Boundary Problem

When both boundaries are fixed, the equations of the bounding surfaces become the controlling influence (i.e., they are program input variables). For this situation, if a particular flow field must be established within the region ABCD (i.e., throat Mach number and flow angle (fig. 2)), a trial-and-error iteration for the surfaces $y = y_{cb}(x)$ and $y = y_c(x)$ must be used.

Free-Boundary Problem

The free-boundary problem is a useful way of prescribing boundary data if a particular flow field is to be established, for example, in the throat region of a supersonic inlet. Prescribing the centerbody contour and velocity distribution establishes the manner in which the flow is to be compressed and the coordinates of the upper surface, $y = y_c(x)$, which is the streamline which passes through the cowl lip. To establish a uniform flow field downstream of domain ABCD, the velocity and surface angle of boundary (1) are held constant downstream from point C. This establishes the requirement of a particular throat Mach number with minimum inviscid flow distortions.

For the free-boundary problem, the establishment of the unknown surface, $y = y_c(x)$, essentially reduces to finding the streamline which passes through point A in figure 2. In any inviscid flow, the streamlines, by definition, form surfaces across which there is no flow and consequently may be replaced by a solid boundary. The streamlines can be determined either by constructing them incrementally from known local conditions or by integration of the mass flux from known boundary conditions. This latter method was used extensively throughout the computer programs. Equations used for the streamline tracing are derived in appendixes C and D, and discussions as to their use appear in detail in later sections.

The boundary for which the pair of data, $y = y_w(x)$ and $q = q_w(x)$, can be prescribed must satisfy the condition that the downstream leading characteristics from the initial datum line do not intersect that surface. If the wrong boundary is chosen, an overspecified problem will result.

Doubly-Free-Boundary Problem

The doubly-free-boundary problem is an alternate way of prescribing boundary data and may have useful applications. Its disadvantage lies in the loss of direct control of flow direction, which becomes important in inlet design. This immediate disadvantage does not occur for either the fixed or free-boundary problems. In any case, the doubly-free-boundary problem was not explored further in this report.

NUMERICAL PROCEDURES

Each net point in the x, y plane is determined by the characteristic equations and compatibility conditions specified by equations (1) and (2). The properties u, v are obtained from the compatibility conditions, while the characteristic equations locate the point in the x, y plane. The computational work divides itself into four distinct groups; namely,

(1) Conical flow field calculations
(2) Basic characteristic field point
(3) Basic characteristic boundary point
(4) Shock point calculations

Conical Flow Field Calculations

The mathematical construction for the flow past an infinite cone with attached shock fronts is obtained from a numerical solution of equation (10) by means of a Runge-Kutta integration. The conical shock angle β_s is established by an iterative process which begins with an initial estimate of the shock angle for the given free-stream conditions. The relations governing the transitions through conical shocks are the same as those for plane oblique shocks; the curvature of the shock cone has no effect. The necessary boundary conditions to be satisfied along the shock front can thus be described by equations (12) to (14). A solution of equation (10) is found which is compatible with the shock values of u, v defined by equations (12) and (13) and has an initial slope specified by equation (14). The solution is then continued so that v_u increases until

$$\frac{x}{y} = \frac{u}{v} = -v_u$$

along the cone surface. The cone angle thus obtained is compared to the specified cone angle, and new estimates of the shock angle β_s are made until the cone angle and flow angle are equal. When convergence has been reached, the boundary conditions u, v on the cone have been established. The solution for any point in the conical flow field can thus be calculated from equation (10) by specifying the ray angle associated with the flow field point. The formulae used for additive drag and mass flow spillage calculation in the conical flow field appear in appendix E.

Basic Characteristic Field Point

The program specifies each field point by a (K, I) index combination. Increasing the K-index (with I held constant) occurs along a C_+ characteristic (fig. 3), while increasing the I-index (with K held constant) indicates a C_- characteristic. The unknown position coordinates x(K, I), y(K, I) are obtained by a simultaneous solution of characteristic equations (1) in finite difference form, that is,

$$x(K, I) = \frac{y(K, I-1) - y(K-1, I) + \bar{A}x(K-1, I) - \bar{B}x(K, I-1)}{\bar{A} - \bar{B}} \tag{19}$$

$$y(K, I) = y(K-1, I) + \bar{A}\left[x(K, I) - x(K-1, I)\right]$$

where

$$\bar{A} = 0.50\left[A(K, I) + A(K-1, I)\right]$$

$$\bar{B} = 0.50\left[B(K, I) + B(K, I-1)\right]$$

The unknown velocity coordinates are obtained from a simultaneous solution of the compatibility conditions (eq. (2)) in finite difference form. that is.

$$v(K, I) = \frac{u(K, I-1) - v(K-1, I) + \bar{D}v(K-1, I) - \bar{E}v(K, I-1) + \bar{G}\left[x(K, I) - x(K, I-1)\right] - \bar{F}\left[x(K, I) - x(K-1, I)\right]}{\bar{D} - \bar{E}} \tag{20}$$

$$u(K, I) = u(K, I-1) + \bar{E}\left[v(K, I) - v(K, I-1)\right] + \bar{G}\left[x(K, I) - x(K, I-1)\right]$$

where

$$\bar{D} = 0.5\left[D(K, I) + D(K-1, I)\right]$$

$$\bar{E} = 0.5\left[E(K, I) + E(K, I-1)\right]$$

$$\bar{F} = 0.5[F(K, I) + F(K - 1, I)]$$

$$\bar{G} = 0.5[G(K, I) + G(K, I - 1)]$$

The iterative process begins with an estimate of the flow properties (i.e., u, v, and c at the field point x(K, I), y(K, I)) which determine the coefficients $\bar{A}$, $\bar{B}$, $\bar{D}$, $\bar{E}$, and $\bar{F}$. The field point and the flow properties are then calculated from equations (19) and (20). New values of the quantities $\bar{A}$, $\bar{B}$, $\bar{D}$, $\bar{E}$, and $\bar{F}$ from which the field point and its properties can be recalculated are thus obtained. The iteration continues until a specified convergence between successive calculations is reached.

Basic Characteristic Boundary Point

Calculation of the basic boundary point (which occurs in the fixed-boundary problem) is performed in an analogous manner to the basic field point except for the replacement of one of the characteristic equations by boundary conditions. The condition of a solid boundary requires that the normal component of velocity at the point x(K, I), y(K, I) (fig. 4) be identically zero.

The boundary curve is specified as a polynomial of third order, that is,

$$y_w = R_0 + R_1 x + R_2 x^2 + R_3 x^3$$

Hence, the boundary point x(K, I), y(K, I) is determined by the simultaneous solution of the C_- characteristic equation with the equation of the boundary curve. Thus,

$$\left.\begin{aligned} R_3 x(K, I)^3 + R_2 x(K, I)^2 + (R_1 - \bar{B})x(K, I) + R_0 - y(K, I - 1) + \bar{B}x(K, I - 1) &= 0 \\ y_w(K, I) = R_0 + R_1 x(K, I) + R_2 x(K, I)^2 + R_3 x(K, I)^3 & \end{aligned}\right\} \quad (21)$$

The velocity components are obtained from the compatibility condition along the C_- characteristic and the condition of no flow normal to the boundary. This gives

$$u(K, I) = \frac{[u(K, I - 1) - \bar{E}v(K, I - 1] + \bar{G}[x(K, I) - x(K, I - 1)]}{1.0 - \epsilon \bar{E}} \quad (22)$$

$$v(K, I) = \epsilon u(K, I)$$

where

$$\epsilon = \frac{dy_w}{dx} = R_1 + 2.0\ R_2 x(K,\ I) + 3.0\ R_3 x(K,\ I)^2$$

The iteration for the basic characteristic boundary point proceeds in a like manner as iteration for the basic field point.

Shock Field Point Calculation

Shock field point calculations are inherently different from field point calculations in that the conditions known are just upstream of the shock wave. Thus, the oblique shock relations must be included in the calculations of the flow field behind the shock wave. The shock point x(K, I), y(K, I) is geometrically located on the downstream side of the intersection of the shock wave and the opposite family of characteristics in the upstream region (fig. 5). The upstream properties of the shock point x(K, I), y(K, I) are found by a linear interpolation between the proper net points in the upstream region. The iterative process to determine the downstream conditions begins with the assumption that the deflection angle at the point x(K, I), y(K, I) is the same as the deflection angle at the previously calculated shock point x(K - 1, I), y(K - 1, I) (fig. 6). From the oblique shock relations, the first estimate of the flow properties behind the shock are established. A C_+ characteristic is then constructed from the shock point x(K, I), y(K, I), as shown by the dashed line in figure 6. The intersection x_{ref}, y_{ref} of this characteristic with its opposite member is the reference point. Conditions at this reference point are obtained by a linear interpolation between the previously calculated points x(K - 1, I), y(K - 1, I) and x(K - 1, I + 1), y(K - 1, I + 1). A basic characteristic field point calculation, previously described, is then performed to obtain the lattice point x(K, I + 1), y(K, I + 1). This calculation is based on the two net points x(K, I), y(K, I) and x(K - 1, I + 1), y(K - 1, I + 1). The shock point x(K, I), y(K, I) is then recomputed by a field point calculation based on the lattice points x_{ref}, y_{ref} and x(K, I + 1), y(K, I + 1). A new estimate of the deflection angle at the shock point is then obtained, and the sequence is repeated until the desired convergence has been obtained.

Shock Boundary Point Calculation

To obtain the quantities at a shock point x(K, I), y(K, I) under the conditions where the previous shock point lies on a physical boundary (fig. 7), a C_+ characteristic is constructed from x(K, I), y(K, I) (dashed line) which intersects the boundary. The intersection point of this characteristic with the boundary becomes the reference point x_{ref}, y_{ref}. The conditions behind the shock at the point x(K, I), y(K, I) are obtained in the manner described in the previous section. The iteration proceeds in a similar manner as the shock field point calculation except the properties at the points x_{ref}, y_{ref} and x(K, I + 1), y(K, I + 1) are obtained by a basic boundary point calculation previously described.

CONSTRUCTION OF FLOW FIELD

Two basic methods of specifying boundary data were used in the computer program. These are (1) the fixed-boundary problem in which the centerbody and cowl contours are the primary input variables to the program and (2) the free-boundary problem where a single surface curve and Mach number distribution are the primary input data. In the actual construction of the flow field, both methods are used in conjunction with each other.

Cowl Lip Calculation

One example of a fixed-boundary problem is illustrated in figure 8. The two boundary curves, $y = y_{cb}(x)$ and $y = y_c(x)$, represent the centerbody and cowl surfaces of the diffuser. The initial data lie along the incident shock; however, these data are incrementally determined as the flow field behind the incident shock is constructed. The construction begins with a shock boundary point calculation, the boundary point being located at point A in figure 8. The calculations always start with a downstream characteristic from the initial datum line, in this case the incident shock. Since the boundary has been reached in the first calculation (point A), the computations proceed with a shock field point calculation to locate the next shock point. Field point calculations are then performed to determine the next downstream characteristic from the shock to the cowl boundary, until the point just before the boundary has been reached. At that time, a boundary point calculation is performed to determine the flow properties on the cowl surface. This sequence of calculations is continued until the entire incident shock has been constructed.

At this point in the calculations, there are several alternatives available in the computer program. These alternatives are discussed in the following sections.

Reflecting Shock Problem

The incident shock in figure 8 can be allowed to reflect off the centerbody surface represented by $y = y_{cb}(x)$. In this case, the construction of the flow field behind the reflected shock proceeds in exactly the same manner as the flow field behind the incident shock, which has previously been discussed.

Canceled-Shock Problem

Another variation of the fixed-boundary problem which is used in the computer program is to specify a lower surface $y = y_{cb}(x)$ such that the shock is canceled at that boundary. The initial datum line now becomes the last downstream characteristic from the incident shock to the upper surface, $y = y_c(x)$. Under these conditions, the lattice points along the initial datum line are known. Hence, only field point and boundary point calculations are used sequentially in the construction of the flow field. If two characteristics of the same family intersect, the upstream characteristic is deleted and the calculations continue. Under this condition, the program automatically prints out the locus of the intersecting characteristics of the same family.

Free-Boundary Problem

Another alternative which is provided in the computer program is to specify that the incident shock (fig. 8) be canceled and to furnish the centerbody contour, $y = y_{cb}(x)$, and the velocity distribution along this surface. This is the free-boundary problem and is illustrated in figure 9. In this case, the initial datum line is specified to be the first downstream characteristic from the upper surface that intersects the lower surface of the diffuser (line AB in fig. 9) which was determined from the previous calculations.

The calculations begin with a stream function integration along the initial characteristic AB from the lower to the upper surface (see appendix C). This integration establishes the numerical value of the streamline which is the upper surface $\psi = \psi_{ref}$ since the point A lies along this contour. The numerical value of the streamline which makes up the lower surface is arbitrary and, hence, set equal to 1 ($\psi = 1$). The construction of the flow field and the determination of the unknown surface, represented by $\psi = \psi_{ref}$, proceed in the following manner. Since the contour and velocity distribution along the lower surface are specified, the sequence of calculations begins with a field point calculation for the second lattice point D along the characteristic CE. Simultaneous with the field point calculation, a mass flux integration is performed. This sequence of calculations is continued along the characteristic CE until the calculated stream function first exceeds the value of the reference stream function (i.e., $\psi = \psi_{ref}$). At this time, an iteration is performed to locate the point along the characteristic CE for which the value of the stream function matches the reference value (i.e., $\psi = \psi_{ref}$). This iteration is based on the two lattice points $\psi < \psi_{ref}$ and $\psi > \psi_{ref}$ indicated by square symbols in figure 9. The boundary coordinates (point D) in addition to the properties at this point are thus established by this iteration. The calculations are continued in the manner described until the entire flow field has been established.

Another example where specifying the contour and the velocity distribution proves advantageous is in the construction of the flow field resulting from an isentropic spike (fig. 10). In this example, the upper surface degenerates to the focal point; hence, this point is a singularity. Conditions at the focal point are specified by assuming two-dimensional reverse Prandtl-Meyer flow. The initial data are obtained from conical flow field calculations at specified intervals. The reference stream function ψ_{ref} is specified by integration of the mass flux along the initial characteristic from the focal point to point A on the cone surface. Field point calculations are then performed simultaneously with a stream-function integration to establish the next leading characteristic through the focal point, OB in figure 10. When the value of the stream function first becomes less than the value of the reference stream function $\psi < \psi_{ref}$ an iteration is performed to locate the surface coordinates, point B. Successive characteristics are then constructed until the flow field and spike contour have been established.

In principle, the establishment of one physical boundary by means of a mass flux integration does not preclude flows where shock waves are present. Under these conditions, however, the integration must account for the change in entropy experienced across the shock wave.

SELECTED INLET EXAMPLES

To indicate the use of the computer program in the design of supersonic inlets, four examples are presented and discussed. The purpose of this section is to indicate the manner in which the computer program can be used to satisfy different design requirements. There are two types of calculations which can be used to solve for the internal flow field. In one type the inlet geometry is specified (fixed-boundary problem) and in the other the Mach number and surface angle distribution must be specified (free-boundary problem). In addition, additive drag and mass flow spillage calculations are included (see appendixes D and E) for conical flow or simple wedge spillage. For those inlet configurations for which the boundary is to be determined, the computer program will curve-fit the unspecified surfaces, and the coefficients will appear in the output listings.

This section also illustrates the use of the primary subroutines and the functional organization of the main program. FORTRAN listings of the main programs and subroutines are found in appendix F. A complete listing of the program is presented in appendix G.

Bicone Inlet

Presented in figure 11 is a 10°-18.5° bicone mixed-compression inlet designed for a free-stream Mach number of 2.50. The second oblique shock intersected the cowl lip for the design condition (fig. 11(b)) while about 0.5 percent of the capture mass flow was spilled supersonically by positioning the cone shock forward of the cowl lip. The initial internal cowl angle was set at 5.0°, while a Mach number of 1.30 and nominal flow angle of -1.4° were specified for the throat conditions. In addition, it was specified that a constant throat Mach number section extending a distance downstream of the throat station be included in the inlet design. Under these conditions, about 60 percent of the supersonic area contraction took place externally. The theoretical total-pressure recovery at the throat was 0.968 behind the terminal shock.

The conical shock angle and boundary conditions on the first cone were computed in subroutine CSA by specifying a free-stream Mach number (AMO) of 2.500 and a cone angle (THETAC) of 10.0°. The profile of the second shock as well as the resulting flow field were computed in subroutine CONE. This was accomplished by specifying the deflection angle (ALPHA) between the two cones and the initial spacing parameter (START) which locates the original net-point spacings along the shock. The coordinate system always places the origin at the spike tip. For the example shown in figure 11, the parameter (START) was set to a value of 0.050, which gave a total of 27 net points along the second shock. The cowl lip is located at unit radius by specifying the angle between the centerline and angular cowl position (THETAL) and setting SPILL = 1.0. This positions the intersection of the two cones relative to the cowl lip by requiring the second oblique shock to intersect the cowl lip. Additive drag and mass flow spillage calculations are then performed in subroutine DRAG, assuming only conical flow spillage (see appendix E).

The internal flow is constructed by using subroutines SHOCK and SURF in combination. Subroutine SHOCK constructs the cowl oblique shock and the resulting flow field by using the specified initial cowl angle (COWLA), which in this example is 5.0°. If there are no coefficients specified for the cowl surface, the program automatically assumes a straight line whose slope is tan(COWLA) and which passes through the cowl lip position. For this particular example, the cowl oblique shock was canceled at the centerbody surface, $x = 2.78$ in figure 11(b). This was accomplished in the program by setting NSHK = 1 (i.e., there should be only one oblique shock internal to the inlet with no shock reflections). Under this condition, subroutine SHOCK locates the first characteristic behind the cowl oblique shock which intersects both the cowl and centerbody surfaces and performs a mass flux integration to determine the reference streamline. This establishes the initial datum line to be used in subroutine SURF and the reference streamline of the cowl. All other information computed downstream of this initial characteristic is thus ignored. Hence, under the condition where NSHK = 1, subroutine SHOCK constructs the flow field within the domain in which the cowl surface can influence the oblique shock profile. Downstream of the initial characteristic which intersects the centerbody determined in subroutine SHOCK, the internal flow as well as the contour coordinate points are constructed in subroutine SURF. This is accomplished by specifying in the main program as input data the parameters AMT(J), THR(J), ANG(J), and NIS(J) (see appendix F). With this information, subroutine SURF constructs a parabolic surface of length THR(J), measured from the last known centerbody surface point, and having a final angle of ANG(J). On this surface, the program assumes a linear Mach number distribution which has a final Mach number of AMT(J) at the end of the section. Subroutine SURF, with this information, constructs the cowl contour which satisfies the boundary conditions specified. The number of net points for this calculation is controlled by the index parameter NIS(J). It can be seen that very complex surfaces which have precise distributions can be constructed from the basic forms described by specifying as many sections (indicated by the J index) as needed. There are several options constructed into this program which serve useful functions. By setting THR(J) = 0, the program will focus all leading characteristics from the cowl at the point which starts the J section. Conditions at the focal point are automatically assumed to satisfy the Prandtl-Meyer relation. By setting ANG(J) = 0, subroutine SURF will construct a centerbody surface whose Mach number distribution is specified by AMT(J) and whose angular distribution is satisfied by focused Prandtl-Meyer compression at a point below the body surface. The final angle in this case is determined by the program. For the example shown in figure 11, the input data were as follows:

SEGMENT	AMT(J)	THR(J)	ANG(J)	NIS(J)
1	1.300	0.400	0	10
2	1.300	.400	0	8
3	1.300	.400	0	8

Subroutine SURF, in this example, constructs the internal contours of the inlet such that the flow is initially compressed isentropically to a Mach number of 1.300 over a distance equal to 40 percent of the cowl lip radius, measured along the centerline from the shock impingement point. This segment of the centerbody surface was assumed to be a streamline constructed from the Prandtl-Meyer relations. Since the flow on the centerbody has already been compressed to Mach 1.300, the data input for segment 2 was used to construct a transition section on the cowl to reduce its Mach number to 1.300. This transition section on the cowl can be seen in figure 11 and lies between $x = 3.02$ and $x = 3.22$. The data for segment 3 were used to hold the 1.300 throat Mach number a distance of 0.400 downstream of segment 2. When the inlet flow field calculations are completed, the program curve-fits the centerbody and cowl contours and lists the table of coefficients as output. These coefficients can then be used to evaluate the off-design capability of the inlet.

Two-Dimensional Focused Compression Inlet

Shown in figure 12 is a two-dimensional isentropic ramp inlet, designed for a free-stream Mach number of 2.70, in which the external flow was compressed to a focal point Mach number of 2.05. The initial ramp angle and internal cowl angle were both set at 5.0°, while at the inlet throat a Mach number of 1.300 and flow angle of -5.00° were specified. With these conditions, approximately 70 percent of the supersonic area contraction took place externally. The Mach number distribution and the characteristic solution along the centerbody and cowl contours are presented in figures 12(a) and (b), respectively. For this inlet, the theoretical total-pressure recovery behind the terminal shock was computed to be 0.956.

For this inlet example, it was specified that two-dimensional flow calculations should be performed. This was accomplished by setting NDIM = 2, as opposed to NDIM = 3 for the previous example. The calculations to construct the isentropic ramp contours and flow field were performed in subroutine SPIKE. Again, the spacing parameter START must be specified to establish the net-point spacings along the initial datum line. In addition, an M-index parameter must be specified to divide the flow field into $M + 1$ C_+ characteristics passing through the focal point. Thus, the M-index parameter divides the Mach number distribution at the focal point into M equal Mach number segments between the initial Mach number and the focal point Mach number (AMF). Downstream of the isentropic section, the flow field is constructed in subroutine SHAPE. The initial datum line in this subroutine is taken to be the last of the family of C_+ characteristics computed in subroutine SPIKE. Unless otherwise specified, the program automatically assumes a straight line segment aft of the isentropic ramp, as was the case for the example shown in figure 12. This condition is indicated by the constant Mach number distribution along the ramp surface downstream of $x = 1.34$ (circular symbols in fig. 12(b)).

For this inlet example, the cowl oblique shock was again canceled at the ramp surface. The internal flow field and surface contours were determined in the same manner as for the bicone inlet previously discussed.

Two-Dimensional Ramp Inlet

Another example of a two-dimensional inlet that was designed by the computer program is shown in figure 13. Although similar in appearance to the example previously discussed, basically different design specifications required a somewhat different computational approach. Foremost in the design specifications was the condition that the initial cowl angle be set at 0°. For optimum overall performance, this requirement dictated that, at most, 50 percent of the supersonic area contraction takes place externally. The throat Mach number and flow angle were set at 1.300 and -5.00°, respectively; the same as the previously discussed two-dimensional inlet. To accomplish these objectives, the external ramp contour was designed such that all the compression waves entered the inlet (fig. 13(b)). This had the desirable effect of reducing the deflection angle at the cowl lip from the 12.2° of the previous example to 5.0°, while keeping the same amount of flow compression on the ramp surface upstream of the oblique shock. Consequently, the theoretical total-pressure recovery behind the terminal shock in the throat was 0.971 for a free-stream Mach number of 2.700.

The external contour of the ramp surface was constructed in two segments. The first segment was a plane two-dimensional ramp with an angle of 5.0°. The second segment was composed of a parabolic section chosen such that the final Mach number on the ramp surface before the cowl oblique shock was about 2.00. Computational work for the external flow field was performed in subroutine CONE by specifying ALPHA = 0 and introducing the proper coefficients for the ramp surface. The internal contours of this inlet were determined in the manner described in the previously discussed examples.

Self-Starting Inlet

The condition of self-starting with a minimum cowl wave drag imposes unusual computational requirements on the fourth inlet example (fig. 14) so that it warrants discussion. Externally, the flow was compressed isentropically to a final focal point Mach number (AMF) of 1.600. The flow field and contour of the isentropic spike were computed in subroutine SPIKE. The initial half-cone angle (THETAC) was set at 16.0° and the free-stream Mach number was specified to be 2.500. It was also specified that for an average entering Mach number of 1.60, the minimum internal Mach number should not be less than 1.35. In addition, the requirement of minimum cowl wave drag dictated that the initial internal cowl angle be about 17.0°, which set the Mach number just behind the oblique shock on the cowl at 1.35. In order to turn the cowl contour back as quickly as possible to reduce the cowl projected area, external expansion was provided on the spike contour aft of the station $x = 1.61$. The contour of this segment of the spike was chosen to provide a nearly uniform Mach number profile at the cowl lip. Results for this section of the flow field were computed in subroutine SHAPE by specifying the appropriate coefficients. A nominal throat flow angle of -3.0° was set for this inlet configuration. The problem, therefore, in this inlet design reduces itself to finding the proper Mach number distribution and contour along the centerbody surface which would ensure a constant Mach number of 1.35 along the cowl surface while turning the flow along the cowl a total of 20.0°.

The calculation for the solution of this problem was accomplished in subroutine SURF by systematic variations of the program input variables AMT(J), THR(J), and ANG(J). By successive approximations, several inlet designs were established, each of which came nearer to satisfying all the requirements. The characteristic solution for the final configuration is shown in figure 14(b), while the Mach number distribution is shown in figure 14(a). The overall total-pressure recovery behind the terminal shock in the throat region was 0.955 for a design Mach number of 2.500.

Off-Design Calculation

Off-design calculations can be performed on the various types of inlet configurations discussed. In each case, the coefficients of the equations specifying the various surfaces are input data to the program. If the cowl oblique shock was to be canceled at the centerbody shoulder for on-design operation, as was the case in the inlets discussed, the location of the shoulder must be specified for off-design calculations. This is done by specifying the index parameter (ISHK) which selects the position point BODY(ISHK) as the shoulder point. It must be understood that canceling the shock for design conditions does not necessarily imply cancellation at off-design Mach numbers.

The position of the cowl lip for off-design calculations is specified by the coordinates X COWL, Y COWL. If these two parameters are nonzero quantities, the program will automatically shift the coefficients describing the cowl contours to their proper positions. If the cowl lip is positioned outside the region of influence of the previously computed flow field, the calculations will terminate, and an error message will be indicated in the output listings. This can be changed by choosing another value of BODY(2) along the centerbody surface which initiates the forward flow field calculations.

COMPARISON WITH EXPERIMENTAL DATA

To illustrate the agreement between the numerical methods described in the previous sections and experimental data, selected examples are presented in figures 15 to 18.

Attention is given in the discussion of these examples to show the influence of boundary layers and to indicate the degree of accuracy of this type of flow calculation.

Isentropic Spike Configurations

Figure 15 presents an isentropic spike forebody configuration designed for a free-stream Mach number of 2.49 and a final focal point Mach number of 1.46. The characteristic solution is shown in figure 15(b), while a comparison between the measured and calculated static-pressure distribution is presented in figure 15(a). The previously unpublished data were obtained from the investigation of reference 6. In general, the boundary layer did not greatly influence the static-pressure distribution along the spike surface, as testified by the good agreement between the calculated (solid line) and measured (symbols) values. Some boundary-layer influence was noticed, however, near the end of the isentropic section, but the focusing characteristics were not appreciably impaired, as indicated by the Schlieren photograph (fig. 15(c)).

More pronounced boundary-layer influence was indicated on an isentropic spike forebody configuration designed for a free-stream Mach number of 3.85 (fig. 16). The axisymmetric characteristic solution for this configuration is shown in figure 16(b). The cone had an initial half angle of 8.0°, while the final focal point Mach number was set at 1.75. Figure 16(a) shows a comparison between the calculated (solid line) and measured (symbols) static-pressure distribution along the spike contour obtained from reference 7. Generally, good agreement was obtained between the experimental and calculated static pressures; however, all data tended to fall somewhat higher than the predicted distribution. This was typical everywhere except for a short range in the no-roughness condition (circular symbols) immediately following the laminar boundary-layer separation region (dashed line). Turbulent boundary layer was undoubtedly induced along the spike surface by the application of roughness at the spike tip, as indicated by the slightly higher static-pressure level (comparing circular and square symbols). This was realized everywhere except in the separation zone of the no-roughness case.

Bicone Forebody Configuration

Shown in figure 17 is a bicone forebody configuration in which the first cone had a half angle of 10.0° while the second cone half angle was 18.5°. Calculations were performed at a free-stream Mach number of 2.49, which was identical to that for the data. The data were obtained in an unpublished investigation conducted in the Lewis 10- by 10-foot supersonic wind tunnel at maximum Reynolds number. The diameter of the oblique shock intersection point was 40.6 centimeters. The axisymmetric characteristic solution is presented in figure 17(b). Coordinate points along the contour were normalized such that the intersection of the first and second oblique shocks occurred at a radius (dimensionless y coordinate) of 1. A comparison between the experimental data (circular symbols) and calculated values (solid line) of static pressure along the surface contour indicated generally fair agreement (fig. 17(a)). Upstream of the junction between the first and second cones, the data were slightly lower than calculated values. Downstream of the juncture point, slightly higher static pressures were measured than were calculated, indicating some boundary-layer influence. Surface disturbances on the second cone, clearly visible in the Schlieren photograph (fig. 17(c)), prevented the possibility of distinguishing any boundary-layer influence aft of $x = 1.40$. There appears to be no boundary-layer "bridging" at the junction of the conical surfaces. This type of flow separation may induce a greatly modified shock pattern, particularly in the vicinity of the cowl lip of an inlet.

Single-Cone Inlet Configuration

Figure 18 presents the computer solution obtained for a single-cone mixed-compression inlet designed for a free-stream Mach number of 2.500. External compression was obtained by a 12.5° half-angle conic forebody. The initial cowl angle was set at 0° in this inlet example; the oblique shock originating from the cowl lip was reflected from the internal contour surfaces. The internal contour surfaces were obtained by curve-fitting the contour specifications used in fabricating the inlet.

The characteristic solution for this inlet configuration is presented in figure 18(b). Shown in figure 18(a) is a comparison between the static-pressure distribution obtained from the computer program (solid and dashed lines) and experimental data (symbols) obtained from reference 8. In general, there was good agreement between the theoretically determined static pressures and measured pressures upstream of the throat region ($x = 3.48$). Within the throat section, agreement between calculations and data is more difficult to realize owing to the complex shock boundary-layer interactions. These interactions were partially minimized by porous bleed in the regions indicated in figure 18(b). In general, the shock reflection points (as indicated by data) appear to occur upstream of the calculated points and thus indicate reflections off the boundary layer. This tends to foreshorten the shock regions in the downstream direction. More detailed discussions of the experimental data can be found in reference 8.

Off-design performance calculations of this inlet at Mach numbers of 2.30 and 2.02 reveal increasingly better agreement in the throat region with decreasing free-stream Mach number. This is probably due to the lessening boundary-layer influence.

CONCLUDING REMARKS

The theory, numerical methods, and computer program for the design of supersonic inlets have been presented. The computer program was written with the intention of designing many types of inlets with a minimum of computer time. For example, the execution time for the inlet presented in figure 11 was 0.62 minute with 30 initial net points along the starting shock. The execution time for the inlet presented in figure 18 was 1.23 minutes for five internal shocks with 31 net points along the first cowl oblique shock wave.

Because computer programs of this type are used primarily for design application, it becomes important for the programmer to use methods whereby optimum designs are more readily obtained. By introducing throat Mach number and flow angle as direct program input variables, the difficulty in obtaining the proper surface contour was considerably reduced. This reduced the overall time required to obtain the designed inlet configuration. In addition, minimum inviscid throat Mach number distortions could be achieved.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, September 3, 1968,
126-15-02-11-22.

APPENDIX A

DERIVATION OF GENERAL COMPATIBILITY AND CHARACTERISTIC RELATIONS

The set of equations which describe steady, irrotational isentropic flow in three dimensions with cylindrical symmetry are

$$(c^2 - u^2)u_x - uv(u_y + v_x) + (c^2 - v^2)v_y + \frac{c^2 v}{y} = 0 \tag{A1}$$

$$v_x - u_y = 0 \tag{A2}$$

$$\sigma^2(u^2 + v^2) + (1 - \sigma^2)c^2 = 1 \tag{A3}$$

where the velocity components u, v and the speed of sound c are dimensionless with respect to the critical speed, and the coordinates of a point x, y are dimensionless with respect to a characteristic length. In equation (A1) the subscripts x, y indicate differentiation. A linear combination of equations (A1) and (A2) is

$$pu_x + qu_y + rv_x + sv_y + t = 0 \tag{A4}$$

where

$$\left.\begin{aligned} p &= \lambda_1(c^2 - u^2) \\ q &= -\lambda_2 \\ r &= \lambda_2 - 2\lambda_1 uv \\ s &= \lambda_1(c^2 - v^2) \\ t &= \frac{\lambda_1 c^2 v}{y} \end{aligned}\right\} \tag{A5}$$

The characteristic equations are obtained by requiring the derivatives of u, v with respect to x, y to form derivatives in the same direction (i.e., dx:dy = p:q = r:s. If

$x(\tau)$ and $y(\tau)$ represent a curve with $x_z:y_z = p:q = r:s$, equation (A4) represents the derivatives of u, v along this curve, and $\tau(x, y)$ define the characteristics. Thus,

or

$$\left.\begin{aligned} \frac{p}{q} &= \frac{r}{s} = \frac{x_\tau}{y_\tau} \\ py_\tau - qx_\tau &= 0 \\ ry_\tau - sx_\tau &= 0 \end{aligned}\right\} \tag{A6}$$

Using the set of equations (A6) gives the pair of linear algebraic equations

$$\left.\begin{aligned} (c^2 - u^2)y_\tau\lambda_1 + x_\tau\lambda_2 &= 0 \\ \left[-2uvy_\tau - (c^2 - v^2)x_\tau\right]\lambda_1 + y_\tau\lambda_2 &= 0 \end{aligned}\right\} \tag{A7}$$

For the set of equations (A7) to have a nontrivial solution for λ, the determinant of the coefficients of λ must vanish; thus,

$$(c^2 - u^2)y_\tau^2 + 2uvx_\tau y_\tau + (c^2 - v^2)x_\tau^2 = 0 \tag{A8}$$

The characteristic directions are therefore given by

$$\zeta_\pm = \frac{y_\tau}{x_\tau} = \frac{1}{c^2 - u^2}\left[-uv \pm \sqrt{(uv)^2 - (c^2 - u^2)(c^2 - v^2)}\right] \tag{A9}$$

and the characteristics equations become

$$\left.\begin{aligned} y_\alpha &= \zeta_+ x_\alpha \\ y_\beta &= \zeta_- x_\beta \end{aligned}\right\} \tag{A10}$$

Equations (A10) are two separate ordinary differential equations of first order which define two one-parameter families of characteristics C_+ and C_- in the x, y plane. These families are represented in the form $\alpha(x, y) = \text{Constant}$, $\beta(x, y) = \text{Constant}$ and form a curvilinear coordinate net such that β is constant along a C_+ characteristic and α is constant along C_-. To obtain the compatibility conditions, equation (A4) was successively multiplied by x and y. Using equation (A6) gives

$$\left.\begin{aligned} pu_\tau + rv_\tau + tx_\tau = 0 \\ qu_\tau + sv_\tau + ty_\tau = 0 \end{aligned}\right\} \tag{A11}$$

Using the set of equations (A5) results in the pair of algebraic equations

$$\left.\begin{aligned} \left[(c^2 - u^2)u_\tau - 2uvv_\tau + \frac{c^2 v}{y} x_\tau\right]\lambda_1 + v_\tau\lambda_2 = 0 \\ \left[(c^2 - v^2)v_\tau + \frac{c^2 v}{y} y_\tau\right]\lambda_1 - u_\tau\lambda_2 = 0 \end{aligned}\right\} \tag{A12}$$

Using the first equation of sets (A7) and (A12), respectively, and requiring that this pair has a nontrivial solution for λ gives the results

$$u_\tau - \zeta_\pm v_\tau - \left(\frac{2uv}{c^2 - u^2}\right) v_\tau + \left(\frac{c^2}{c^2 - u^2}\,\frac{v}{y}\right) x_\tau = 0$$

The pair of compatibility conditions are thus given by

$$\left.\begin{aligned} u_\alpha + \zeta_- v_\alpha + \left(\frac{c^2}{c^2 - u^2}\,\frac{v}{y}\right) x_\alpha = 0 \\ u_\beta + \zeta_+ v_\beta + \left(\frac{c^2}{c^2 - u^2}\,\frac{v}{y}\right) x_\beta = 0 \end{aligned}\right\} \tag{A13}$$

The characteristic equations and compatibility relations can be more conveniently expressed by introducing the angle ϑ between the flow direction and the positive x-axis; then,

$$\left.\begin{aligned} u = q \cos \vartheta \\ v = q \sin \vartheta \end{aligned}\right\} \tag{A14}$$

The roots ζ_+ and ζ_-, being the slopes of characteristic directions of C_+ and C_-, are thus

$$\left.\begin{aligned} \zeta_+ &= \tan\,(\vartheta + \mu) \\ \zeta_- &= \tan\,(\vartheta - \mu) \end{aligned}\right\} \tag{A15}$$

where μ is the angle between the flow direction and that of the corresponding characteristic.

The characteristic expression, equation (A8), can be written in the form

$$c^2(1 + \zeta_\pm^{\,2}) - (u\zeta_\pm - v)^2 = 0$$

Using equations (A14) and (A15) gives the following result:

$$c^2 = q^2 \sin^2 \mu \tag{A16}$$

Thus, the speed of sound is simply the component of flow velocity normal to the direction of a characteristic.

With these rotations, the characteristic equations assume the form

$$\left.\begin{aligned} y_\alpha &= \tan(\vartheta + \mu)x_\alpha \qquad C_+ \\ y_\beta &= \tan(\vartheta - \mu)x \qquad C_- \\ u_\alpha &= -\tan(\vartheta - \mu)v - \frac{c^2}{c^2 - u^2}\,x \qquad C_+ \\ u_\beta &= -\tan(\vartheta + \mu)v_\beta - \frac{c^2}{c^2 - u^2}\,x \qquad C_- \end{aligned}\right\} \tag{A17}$$

APPENDIX B

DERIVATION OF DIFFERENTIAL EQUATIONS OF CONICAL FLOW

For the mathematical construction of conical flow patterns, assume that the velocity components u, v and hence the speed of sound c depend only on the ratio

$$\eta = \frac{x}{y} \tag{B1}$$

Equation (A2) thus becomes

$$v_\eta + \eta u_\eta = 0 \tag{B2}$$

and equation (A1) becomes

$$(c^2 - u^2)u_\eta - 2uvv_\eta - (c^2 - v^2)\eta v_\eta + c^2 v = 0 \tag{B3}$$

where subscripts refer to differentiation. Equation (B3) assumes a particularly useful form when v is introduced as a function of u; thus, from (B2)

$$\eta = -\frac{v_\eta}{u_\eta} = -v_u \tag{B4}$$

Differentiating (B4) with respect to u leads to the following relation:

$$\left.\begin{aligned} u_\eta &= -\frac{1}{v_{uu}} \\ v_\eta &= -\frac{v_u}{v_{uu}} \end{aligned}\right\} \tag{B5}$$

Introducing equations (B4) and (B5) into equation (B3) leads to the particularly simple form of the Taylor-Maccoll equation

$$vv_{uu} = 1 + v_u^2 - c^{-2}(u + vv_u)^2 \tag{B6}$$

Eliminating c^2 by mean of Bernoulli's equations yields

$$vv_{uu} = \left(1 + v_u^2\right) - \frac{(1 - \sigma^2)\,(u + vv_u)^2}{1 - \sigma^2(u^2 + v^2)} \tag{B7}$$

Along the cone surface, the flow has the direction of the ray $\eta = x/y$ traced by the cone in the x, y plane, and thus

$$\left.\begin{aligned} \eta &= \frac{u}{v} \\ v_u &= -\frac{u}{v} \end{aligned}\right\} \tag{B8}$$

The conditions to be satisfied along the conical shock (ref. 1) are given by

$$\left.\begin{aligned} u &= (1 - \sigma^2)q_0 \cos^2\beta_s + \frac{1.0}{q_0} \\ v &= (q_0 - u) \cot \beta_s \end{aligned}\right\} \tag{B9}$$

where q_0 is the dimensionless free-stream velocity and β_s is the conical shock angle. In addition, the initial slope along the shock is given by

$$v_u = \frac{v}{u - q_0} \tag{B10}$$

APPENDIX C

DERIVATION OF STREAMLINES IN AXISYMMETRIC FLOWS

The continuity equation for axisymmetric flow may be written as

$$(\rho u y)_x + (\rho v y)_y = 0 \tag{C1}$$

From equation (C1), there exists a stream function x, y such that

$$\left.\begin{aligned} \psi_x &= -\rho v y \\ \psi_y &= \rho u y \end{aligned}\right\} \tag{C2}$$

Along the two one-parameter families of characteristics C_+ and C_- in the x, y plane, which are represented by $\beta(x, y) = \text{Constant}$ and $\alpha(x, y) = \text{Constant}$,

$$\left.\begin{aligned} \psi_\alpha &= \psi_x x_\alpha + \psi_y y_\alpha \\ \psi_\beta &= \psi_x x_\beta + \psi_y y_\beta \end{aligned}\right\} \tag{C3}$$

where the derivatives x_α, y_α and x_β, y_β are given by

$$\left.\begin{aligned} x_\alpha &= \cos(\vartheta + \mu) \quad \text{along } C+ \\ y_\alpha &= \sin(\vartheta + \mu) \quad \text{along } C_+ \\ x_\beta &= \cos(\vartheta - \mu) \quad \text{along } C_- \\ y_\beta &= \sin(\vartheta - \mu) \quad \text{along } C_- \end{aligned}\right\} \tag{C4}$$

When cylindrical coordinates are introduced such that

$$u = q \cos \vartheta$$

$$v = q \sin \vartheta$$

and the relations, equations (C4) are used, equation (C3) becomes

$$\left.\begin{aligned}\psi_\alpha &= qy\left[\sin(\vartheta + \mu)\cos\vartheta - \cos(\vartheta + \mu)\sin\vartheta\right]\\ \psi_\beta &= qy\left[\sin(\vartheta - \mu)\cos\vartheta - \cos(\vartheta - \mu)\sin\vartheta\right]\end{aligned}\right\} \tag{C5}$$

Simplifying the above equations results in

$$\left.\begin{aligned}\psi_\alpha &= qy \sin\mu = \frac{\rho qy}{M}\\ \psi_\beta &= -qy \sin\mu = \frac{-\rho qy}{M}\end{aligned}\right\} \tag{C6}$$

From the set of equations (C4),

$$\left.\begin{aligned}d\alpha &= \frac{dy}{\sin(\mu + \vartheta)}\\ d\beta &= -\frac{dy}{\sin(\mu - \vartheta)}\end{aligned}\right\} \tag{C7}$$

Therefore,

$$\psi_B = \psi_A + \int_{Y_A}^{Y_B} \frac{\rho qy\, dy}{M \sin(\mu \pm \vartheta)} \tag{C8}$$

where the + sign is used for integration along a C_+ characteristic and the - sign is used for integration along a C_- characteristic. If the interval AB is used as the distance between successive points on the characteristic net (along the appropriate characteristic), it may be assumed that $\rho q/M \sin(\mu \pm \vartheta)$ takes the average of the values at points A and B; thus,

$$\psi_B = \psi_A + \frac{1}{4}\left\{\left[\frac{\rho q}{M \sin(\mu \pm \vartheta)}\right]_A + \left[\frac{\rho q}{M \sin(\mu \pm \vartheta)}\right]_B\right\}\left(y_B^2 - y_A^2\right) \tag{C9}$$

For two-dimensional flows, equation (C9) reduces to the expression

$$\psi_B = \psi_A + \frac{1}{2}\left\{\left[\frac{\rho q}{M \sin(\mu \pm \vartheta)}\right]_A + \left[\frac{\rho q}{M \sin(\mu \pm \vartheta)}\right]_B\right\}(y_B - y_A) \tag{C10}$$

The derivation of the stream-function integration procedure along a C_+ characteristic appeared first in reference 9. For a simple wave region (two-dimensional flow) the integration of equation (C8) takes on a particularly simple form since the integrand becomes constant along either the C_+ or C_- family of characteristics; thus,

$$\psi_B = \psi_A + \frac{\rho q}{M \sin(\mu \pm \vartheta)} (y_B - y_A) \tag{C11}$$

APPENDIX D

DERIVATION OF STREAMLINES IN CONICAL FLOW FIELD

The construction of streamlines in a conical flow field becomes particularly simple if the integration of mass flow is performed along rays passing through the apex of the cone. Thus, equation (C3) is replaced by

$$\psi_r = \psi_x x_r + \psi_y y_r \tag{D1}$$

where

$$r = \sqrt{x^2 + y^2}$$

The derivatives x_r and y_r are given by

$$\left.\begin{aligned} x_r &= \cos \Theta \\ y_r &= \sin \Theta \end{aligned}\right\} \tag{D2}$$

where Θ is the ray angle. Using equations (C2) and the polar form of the velocity components gives

$$\left.\begin{aligned} \psi_r &= \rho q y(\sin \Theta \cos \vartheta - \cos \Theta \sin \vartheta) \\ \psi_r &= \rho q y \sin(\Theta - \vartheta) \end{aligned}\right\} \tag{D3}$$

Therefore, when equation (D2) is used, the streamline is specified by

$$\psi = \psi_A + \int_{y_A}^{y_B} \left[\frac{\rho q \sin(\Theta - \vartheta)}{\sin \Theta}\right] y \, dy \tag{D4}$$

Since integration of the mass flux is performed along rays passing through the cone apex, the quantity inside the brackets is constant, and hence

$$\psi = \psi_A + \left\{\rho(\Theta)q(\Theta)\,\frac{\sin\left[\Theta - \vartheta(\Theta)\right]}{\sin\Theta}\right\}\frac{1}{2}\left(y_B^2 - y_A^2\right)$$

When ψ_A, $y_A = 0$ with no loss in generality, the stream function becomes a function of y along a specified ray of angle Θ; thus,

$$\psi = \frac{\rho(\Theta)q(\Theta)\sin\left[\Theta - \vartheta(\Theta)\right]}{2\sin\Theta}\,y^2 \tag{D5}$$

APPENDIX E

ADDITIVE DRAG AND MASS FLOW SPILLAGE IN CONICAL FLOW FIELD

For a conical flow field, the transition across the shock wave is governed by the oblique shock relations and is followed by a continuous isentropic compression to surface conditions (fig. 19). The capture streamline is established by an integration of the mass flux along successive generatrix rays lying between the cowl lip location (point P in fig. 19) and the shock wave. Derivations of the integral formulas used in this calculation are presented in appendix D. The reference stream-function value (capture streamline) is first established by integration of the mass flux along the generatrix ray OP by using equation (D5); that is,

$$\psi_{ref} = \frac{\rho(\Theta_p)q(\Theta_p)\sin\left[\Theta_p - \vartheta(\Theta_p)\right]}{2.0 \sin \Theta_p} y(\Theta_p)^2 \tag{E1}$$

Subsequent points along the capture streamline (i.e., points Q, R, and S in fig. 19) are thus obtained from the relation

$$y(\Theta) = \sqrt{\frac{2.0\, \psi_{ref}(\sin \Theta)}{\rho(\Theta)q(\Theta)\sin\left[\Theta - \vartheta(\Theta)\right]}} \tag{E2}$$

When the generatrix ray becomes coincident with the conical shock wave, the capture mass flow is obtained from the equation

$$\frac{m}{m_0} = \frac{\rho_0 q_0 y(\Theta_S)^2}{\rho_0 q_0 y(\Theta_P)^2} = \frac{y(\Theta_S)^2}{y(\Theta_P)^2} \tag{E3}$$

When the notation in figure 19 is used, the additive drag coefficient is defined as

$$C_{d,a} = -\frac{4}{\gamma p_0 M_0^2 y(\Theta_P)^2} \int_{Y_P}^{Y_S} (p - p_0)y\, dy \tag{E4}$$

Equation (E4) may be approximated by the method of numerical integration. Thus, the additive drag was computed by using the relation

$$C_{d,a} = - \frac{1}{\gamma M_0^2 y(\Theta_p)^2} \sum_j \left[\frac{p(j)}{p_0} + \frac{p(j+1)}{p_0} - 2\right] \left[y(j)^2 - y(j+1)^2\right] \qquad \text{(E5)}$$

where the J-index designates the intersection points of the capture streamline and generatrix rays, starting from the cowl lip position.

APPENDIX F

COMPUTER PROGRAM FOR DESIGN OF SUPERSONIC INLETS

The required input data are presented for this program so that it may be readily employed. In actuality, there are two main programs, INLET 1 and INLET 3, to assemble the subroutines to arrive at the desired inlet design. The difference between INLET 1 and INLET 3 is the type of forebody configuration desired for the inlet. INLET 1 designs an inlet with a focused isentropic forebody configuration, while INLET 3 assumes a bicone or biramp forebody. The input data required for both of these main programs are the same except for three parameters which are noted. In addition, the computer program contains several options, depending on the type and amount of calculations required. This is made clear in the following outline of the input required for this program.

INPUT VARIABLES AND EXPLANATION

Card	FORTRAN symbol	Card columns	Description and comments
1	AMO	1-12	free-stream Mach number
	GAM	13-24	ratio of specific heats
	THETC	25-36	initial cone or ramp angle, radians
	BETAE	37-48	estimate of cone shock angle, radians
	AMF	49-60	for INLET 1 only: final focal point Mach number for isentropic forebody configuration
	ALPHA	49-60	for INLET 3 only: deflection angle
	DELB	61-72	initial increment used in calculation for cone shock angle; set DELB = 1.0×10^{-4}
2	DELU1	1-12	integration increment for conic flow field calculations; set DELU1 = 1.0×10^{-5}
	ERROR	13-24	convergence parameter; set ERROR = 1.0×10^{-4} for most applications
	COWLA	25-36	initial cowl angle, radians
	THETAP	37-48	for INLET 1 only: location of focal point relative to spike tip (measured in radians)

Card	FORTRAN symbol	Card columns	Description and comments
2	THETAL	37-48	for INLET 3 only: location of cowl lip relative to spike tip (measured in radians)
	START	49-60	initial net-point spacing parameter

Both THETAP and THETAL are angular locations relative to the inlet centerline. These parameters are used only when the cowl contour is unspecified. Under this condition, the program always takes the cowl lip radius as 1.

3	FOCUS	1-12	for INLET 1 only: set FOCUS = 0 to locate the focal point along the initial shock wave; set FOCUS = 1.0 to locate the focal point at THETAP (In both of these cases, the cowl lip is placed at the focal point.)
	SPILL	1-12	for INLET 3 only: set SPILL = 0 to locate the cowl along the initial shock wave; set SPILL = 1.0 to locate the cowl at THETAL
	DELP	13-24	parameter which limits the extent of the initial datum line for off-design calculations (OFF = 1.0); ratio of angular location of last initial data point to initial shock angle

The program constructs the initial datum line from the point on the surface, BODY (2), to a point in the field whose angular location is DELP times the initial shock angle.

3	OFF	25-36	set OFF = 0 for on-design calculations; set OFF = 1.0 to indicate off-design calculations
	XCOWL	37-48	x-coordinate of cowl lip position
	YCOWL	49-60	y-coordinate of cowl lip position

The parameters XCOWL and YCOWL are used when a relative shift between the spike tip and cowl lip is desired. Under these conditions, it is recommended that the cowl contours corresponding to the design conditions be used. The program will automatically shift the cowl contour such that the cowl lip is located at XCOWL, YCOWL. If there is no shift, set XCOWL, YCOWL = 0. If the cowl lip falls outside the region of influence of the previously computed flow field, an error message will be printed out and the calculations terminated. This can be corrected by choosing a different BODY (2) position, which is a dummy point used to start the initial flow field calculations.

Card	FORTRAN symbol	Card columns	Description and comments
3	PRINT	61-72	set PRINT = 0 to bypass the output printout for the forebody flow field calculations; otherwise set PRINT = 1.0
4	NDIM	1-6	set NDIM = 3 for axisymmetric flow field calculations; set NDIM = 2 for two-dimensional flow field calculations
	M	7-12	number of initial I-net point spacings for isentropic spike or ramp calculations
	NR	13-18	number of sets of cowl contour coefficients read in as input
	NS	19-24	number of sets of body contour coefficients read in as input
	NTHR	25-30	number of internal isentropic compression sections
	ND	31-36	number of increments in additive drag calculation
	ISHK	37-42	index which specifies at which centerbody contour point BODY (ISHK) a discontinuity occurs for cowl shock cancellation; used with off-design calculations only, i.e., OFF = 1.0; otherwise set ISHK = 1, indicating no shock cancellation
	NSHK	43-48	number of internal shock waves to be computed; NSHK-1 specifies the number of internal shock reflection points; to cancel cowl lip shock, set NSHK = 1
5	R(I, 1)	1-12	cowl contour is specified by a third-order polynomial of form $Y = R(I,1) + R(I,2)X + R(I,3)X^2 + R(I,4)X^3$
	R(I, 2)	13-24	
	R(I, 3)	25-36	
	R(I, 4)	37-42	COWL(I) specifies the x-location after which the I-set of coefficients is valid; hence, the I-set of coefficients is valid in the region $COWL(I) \leq X < COWL(I+1)$
	COWL(I)	43-54	
	I=1, NR		

Card	FORTRAN symbol	Card columns	Description and comments
5	S(I, 1)	1-12	centerbody is specified by equation
	S(I, 2)	13-24	$y = S(I, 1) + S(I, 2)X + S(I, 3)X^2 + S(I, 4)X^3$
	S(I, 3)	25-36	
	S(I, 4)	37-42	valid in the region $BODY(I) \leq X < BODY(I+1)$
	BODY(I) I=1, NS	43-54	BODY(I) performs the same function as the parameter COWL(I)

The coefficients for the NR^{th} and NS^{th} set are dummy variables and are set equal to zero, while COWL(NR) and BODY(NR) specify the region in which the (NR-1), (NS-1) set of coefficients are valid. If the cowl and centerbody contours are to be computed, set NR = 1 and NS = 1 and set the coefficients to zero. If a set of coefficients is provided, for example, for the forebody contour, the program will automatically recompute the internal contours for the conditions specified. The following set of cards read in information pertinent to computing the internal cowl and centerbody contours, otherwise they are unnecessary.

6	AMT(J) J=1, NTHR	1-12	final Mach number specified for the end of the J^{th} section
	THR(J)	13-24	length of J^{th} section
	ANG(J)	25-36	final surface angle (radians) for end of J^{th} section
	NIS(J)	37-42	number of surface net points within THR(J)

```
$IBFTC INLET3  LIST,REF,DECK,DEBUG
C
C     CHARACTERISTIC PROGRAM FOR TWO RAMP INLET, NDIM=2
C
C     CHARACTERISTIC PROGRAM FOR TWO CONE INLET, NDIM=3
C
C     AMO=FREE STREAM MACH NUMBER
C     GAM=RATIO OF SPECIFIC HEATS
C     THETC=INITIAL CONE OR RAMP ANGLE (RADIANS)
C     ALPHA=SECOND CONE OR RAMP DEFLECTION ANGLE (RADIANS)
C     COWLA=COWL ANGLE (RADIANS)
C     THETAL=LOCATION OF COWL RELATIVE TO SPIKE TIP (RADIANS)
C     BETAE=ESTIMATE OF CONE SHOCK ANGLE (RADIANS)
C     R(I,J)=COEFFICIENTS OF COWL CONTOUR SECTIONS
C     S(I,J)=COEFFICIENTS OF CENTERBODY CONTOUR SECTIONS
C     COWL(I)=COWL CONTOUR REGIONS
C     BODY(I)=CENTERBODY CONTOUR REGIONS
C     DELU1=INTEGRATION INCREMENT IN CONE FIELD CALCULATION
C     ERROR=CONVERGENCE PARAMETER, SET ERROR=0.1-3 FOR MOST APPLICATIONS
C     COWLX=X-COORDINATE OF COWL LIP
C     COWLY=Y-COORDINATE OF COWL LIP
C
C     M=NUMBER OF INITIAL I-NET SPACE
C     NR=NUMBER OF COWL CONTOUR SECTIONS
C     NS=NUMBER OF CENTERBODY CONTOUR SECTIONS
C     NTHR=NUMBER OF ISENTROPIC COMPRESSION SECTIONS
C     NSHK=NUMBER OF INTERNAL SHOCKS
C     ND=NUMBER OF INTEGRATION INCREMENTS IN ADDITIVE DRAG CALCULATION
C     ISHK=SHOCK IMPINGMENT POINT ON BODY CONTOUR SECTION
C
C     COMMAND PARAMETERS
C
C     SET NDIM=2 FOR TWO DIMENSIONAL FLOW CALCULATIONS
C     SET NDIM=3 FOR THREE DIMENSIONAL FLOW CALCULATIONS
C     SET ALPHA=0.0 FOR A SINGLE CONE INLET
C     SET SPILL=0.0 FOR COWL LOCATED ON FIRST OBLIQUE SHOCK
C     SET SPILL=1.0 FOR COWL LOCATED AT THETAL
C     SET NSHK=1 FOR ISENTROPIC INTERNAL COMPRESSION
C     SET NSHK GREATER THAN 1, ISHK=1, FOR INTERNAL SHOCKS TO REFLECT
C     SET OFF=0.0 FOR ON-DESIGN CALCULATIONS
C     SET OFF=1.0 FOR OFF-DESIGN CALCULATIONS
C
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      DIMENSION AMT(10),THR(10),ANG(10),NIS(10)
    5 READ (5,400) AMO,GAM,THETC,BETAE,ALPHA,DELB,DELU1,ERROR,COWLA,
     1THETAL,START
      READ (5,402) SPILL,DELP,OFF,XCOWL,YCOWL,PRINT
      READ (5,404) NDIM,M,NR,NS,NTHR,ND,ISHK,NSHK
      READ (5,406) ((R(I,J),J=1,4),COWL(I),I=1,NR)
      READ (5,406) ((S(I,J),J=1,4),BODY(I),I=1,NS)
  400 FORMAT (6E12.0/5E12.0)
  402 FORMAT(6E12.0)
```

```
  404 FORMAT (8I6)
  406 FORMAT (5E12.0)
  408 FORMAT (3E12.0,I6)
  500 FORMAT (//37X,42HCHARACTERISTIC PROGRAM FOR A BI-RAMP INLET//)
  502 FORMAT (//37X,42HCHARACTERISTIC PROGRAM FOR A BI-CONE INLET//)
  600 FORMAT (//30X,3HAMO,10X,5HTHETC,10X,2HBS,10X,5HALPHA,10X,5HCOWLA,
     1//)
  602 FORMAT (23X,1P5E14.5)
  604 FORMAT (//42X,23HCOWL LOCATED AT THETA =1PE12.5//)
  606 FORMAT (//28X,62HSHIFT IN CENTERBODY COEFFICIENTS DUE TO SHOCK IMP
     1INGMENT POINT//)
  608 FORMAT (//40X,6HDELX =1PE12.5,2X,6HDELY =1PE12.5//)
  610 FORMAT (1H1)
      REG=1.0
      SHIFTX=XCOWL-COWL(1)
      SHIFTY=YCOWL-1.000
      IF (XCOWL .EQ. 0.0) SHIFTX=0.0
      IF (YCOWL .EQ. 0.0) SHIFTY=0.0
      DO 10 I=1,NR
      CALL SHIFT(I,SHIFTX,SHIFTY)
   10 CONTINUE
      DO 12 K=1,50
      DO 12 I=1,50
      CALL EQUK(K,I,0.0,0.0,0.0,0.0)
      CALL EQUI(I,0.0,0.0,0.0,0.0)
   12 CONTINUE
      CONVP=(1.0+(GAM-1.0)/2.0)**(-GAM/(GAM-1.0))
      CONVD=(1.0+(GAM-1.0)/2.0)**(-1.0/(GAM-1.0))
      SIGSQ=(GAM-1.0)/(GAM+1.0)
      IF (NDIM .GT. 2) GO TO 14
      CALL PSA(AMO,0.0,THETC)
      UC=UK
      VC=VK
      BS=BETAK
      DELC=TAN(THETC)
      DELS=TAN(BS)
      GO TO 16
   14 CALL CSA(BETAE)
      U1=UC
      V1=VC
      DEL1=-1.0/DELC
   16 CALL OSWR(AMO,BS)
      IREV=1
      RECV(1,IREV)=RECOV
      RECV(2,IREV)=RECOV
      FACTR=RECOV*CONVP
      K=1
      I=1
      SIGN=-1.0
      WRITE (6,610)
      IF (NDIM .GT. 2) GO TO 18
      WRITE (6,500)
      GO TO 20
   18 WRITE (6,502)
   20 WRITE (6,600)
      WRITE (6,602) AMO,THETC,BS,ALPHA,COWLA
      IF (OFF .EQ. 0.0) GO TO 26
      XCOWL=COWL(1)
      YCOWL=1.000
      THETAL=ATAN(YCOWL/XCOWL)
```

```
22 WRITE (6,604) THETAL
   DELS=DELP*DELS
24 CALL OUT
   WRITE (6,610)
   IF (ABS(ALPHA) .GT. 0.0) GO TO 30
   IF (ABS(S(1,2)-S(2,2)) .GE. ERROR) GO TO 30
   CALL DRAG(THETAL,1.0,ND)
   DELS=DELP*TAN(BS)
   GO TO 30
26 IF (SPILL .GT. 0.0) GO TO 28
   THETAL=BS
28 CALL DRAG(THETAL,1.0,ND)
30 K=1
   I=1
   SIGN=-1.0
   EXTEND=ISHK
   BODY(ISHK)=EXTEND*BODY(ISHK)
   REG=-1.0
   CALL CONE(K,I,START,ALPHA,SIGN,PRINT)
   IREF=1
   IF (NR .GT. 1) GO TO 32
   XS=X(KREF,IREF)
   YS=Y(KREF,IREF)
   DELSP=TAN(COWLA)
   XP=XS+3.0
   REG=1.0
   CALL CURVE(XS,YS,DELSP,XP,0.0,0.0,1.0,NR)
   NR=NR+1
   GO TO 34
32 IF (ABS(ALPHA) .GT. 0.0) GO TO 34
   IF (OFF .GT. 0.0) GO TO 34
   XCOWL=COWL(1)
   YCOWL=1.000
   DIST=XCOWL-X(KREF,IREF)
   IF (ABS(DIST) .GT. ERROR) GO TO 38
36 CALL EQUK(KREF,IREF,XCOWL,YCOWL,U(KREF,IREF),V(KREF,IREF))
   GO TO 42
34 IF (OFF .EQ. 0.0) GO TO 42
38 CALL PUT(XCOWL,YCOWL)
   BODY(ISHK)=BODY(ISHK)/EXTEND
42 DELTA=COWLA-ATAN(V(KREF,IREF)/U(KREF,IREF))
   IF (ABS(DELTA) .LT. 0.010) DELTA=0.0
   IF (DELTA .GT. 0.0) DELTA=0.0
   IF (NSHK .EQ. 0) GO TO 64
   IF (NSHK .GT. 1) GO TO 44
   SETP=OFF
   SETQ=OFF
   GO TO 46
44 SETP=1.0
   SETQ=0.0
46 CALL MATRX
   KR=KREF
   IR=IREF
   DO 48 J=1,NSHK
   SIGN=(-1.0)**(J+1)
   REG=SIGN
   IF (NSHK .EQ. 1) BODY(ISHK)=BODY(ISHK)*EXTEND
   CALL SHOCK(KR,IR,DELTA,SIGN,SETP,SETQ)
   IF (NSHK .EQ. 1) BODY(ISHK)=BODY(ISHK)/EXTEND
   KR=KREF
```

```
      IR=1
      STFR=PSI
      DELTA=-DELA
   48 CONTINUE
      IF (NSHK .GT. 1) GO TO 64
      IF (OFF .EQ. 0.0) GO TO 60
      IF (ISHK .EQ. 1) GO TO 56
      XSHK=BODY(ISHK)
      YSHK=S(ISHK,1)+S(ISHK,2)*XSHK+S(ISHK,3)*XSHK**2+S(ISHK,4)*XSHK**3
      DELX=X(1,1)-XSHK
      DELY=Y(1,1)-YSHK
      IF (ABS(DELX) .GT. ERROR) GO TO 50
      IF (ABS(DELY) .LT. ERROR) GO TO 56
   50 REG=-1.0
      DO 52 I=ISHK,NS
      CALL SHIFT(I,DELX,DELY)
   52 CONTINUE
      WRITE (6,610)
      WRITE (6,606)
      WRITE (6,608) DELX,DELY
   54 CALL OUT
   56 IF (NTHR .EQ. 0) GO TO 64
      DO 58 J=1,NTHR
      REG=(-1.0)**(J)
      CALL SHAPE(1,1,REG,1.0,1.0)
   58 CONTINUE
      GO TO 66
   60 IF (NTHR .EQ. 0) GO TO 64
      KR=KREF
      IR=IREF
      READ (5,408) (AMT(J),THR(J),ANG(J),NIS(J), J=1,NTHR)
      DO 62 J=1,NTHR
      CALL SURF(KR,IR,NIS(J),AMT(J),THR(J),ANG(J),STFR,1.0)
      KR=KREF
      IR=1
   62 CONTINUE
   64 WRITE (6,610)
      CALL OUT
   66 GO TO 5
      END
```

```
$IBFTC INLET1  LIST,REF,DECK,DEBUG
C
C     CHARACTERISTIC PROGRAM FOR ISENTROPIC RAMP INLET, NDIM=2
C
C     CHARACTERISTIC PROGRAM FOR ISENTROPIC SPIKE INLET, NDIM=3
C
C     AMO=FREE STREAM MACH NUMBER
C     GAM=RATIO OF SPECIFIC HEATS
C     THETC=INITIAL CONE OR RAMP ANGLE (RADIANS)
C     AMF=EXTERNAL FOCAL POINT MACH NUMBER
C     COWLA=COWL ANGLE (RADIANS)
C     THETAP=LCOATION OF FOCAL POINT (RADIANS)
C     BETAE=ESTIMATE OF CONE SHOCK ANGLE (RADIANS)
C     R(I,J)=COEFFICIENTS OF COWL CONTOUR SECTIONS
C     S(I,J)=COEFFICIENTS OF CENTERBODY CONTOUR SECTIONS
C     COWL(I)=COWL CONTOUR REGIONS
C     BODY(I)=CENTERBODY CONTOUR REGIONS
C     DELB=INITIAL INCREMENT FOR CONE SHOCK ANGLE CALCULATION (RADIANS)
C     DELU1=INTEGRATION INCREMENT IN CONE FIELD CALCULATION
C     ERROR=CONVERGENCE PARAMETER, SET ERROR=0.1-3 FOR MOST APPLICATIONS
C     SHIFTX=SHIFT IN COWL X-COORDINATE FROM DESIGN POSITION
C     SHIFTY=SHIFT IN COWL Y-COORDINATE FROM DESIGN POSITION
C
C     M=NUMBER OF INITIAL I-NET SPACE
C     NR=NUMBER OF COWL CONTOUR SECTIONS
C     NS=NUMBER OF CENTERBODY CONTOUR SECTIONS
C     NTHR=NUMBER OF ISENTROPIC COMPRESSION SECTIONS
C     NSHK=NUMBER OF INTERNAL SHOCKS
C     ND=NUMBER OF INTEGRATION INCREMENTS IN ADDITIVE DRAG CALCULATION
C     ISHK=SHOCK IMPINGMENT POINT ON BODY CONTOUR SECTION
C     INDEX=K-INDEX TO INCREASE NUMBER OF GRID MESH POINTS
C     NGRID=NUMBER OF ADDITIONAL GRID SPACINGS
C
C     COMMAND PARAMETERS
C
C     SET NDIM=2 FOR TWO DIMENSIONAL FLOW CALCULATIONS
C     SET NDIM=3 FOR THREE DIMENSIONAL FLOW CALCULATIONS
C     SET FOCUS=0.0 FOR EXTERNAL COMPRESSION TO FOCUS ON SHOCK
C     SET FOCUS=1.0 FOR EXTERNAL COMPRESSION TO FOCUS AT THETAP
C     SET NSHK=1 FOR ISENTROPIC INTERNAL COMPRESSION
C     SET NSHK GREATER THAN 1, ISHK=1, FOR INTERNAL SHOCKS TO REFLECT
C     SET OFF=0.0 FOR ON-DESIGN CALCULATIONS
C     SET OFF=1.0 FOR OFF-DESIGN CALCULATIONS
C
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      DIMENSION AMT(10),THR(10),ANG(10),NIS(10)
    5 READ (5,400) AMO,GAM,THETC,BETAE,AMF,DELB,DELU1,ERROR,COWLA,
     1THETAP,START
      READ (5,402) FOCUS,DELP,OFF,XCOWL,YCOWL,PRINT
      READ (5,404) NDIM,M,NR,NS,NTHR,ND,ISHK,NSHK
      READ (5,406) ((R(I,J),J=1,4),COWL(I),I=1,NR)
      READ (5,406) ((S(I,J),J=1,4),BODY(I),I=1,NS)
  400 FORMAT (6E12.0/5E12.0)
  402 FORMAT (6E12.0)
```

```
404 FORMAT (8I6)
406 FORMAT (5E12.0)
408 FORMAT (3E12.0,I6)
500 FORMAT (//33X,50HCHARACTERISTIC PROGRAM FOR A ISENTROPIC RAMP INLE
   1T//)
502 FORMAT (//33X,51HCHARACTERISTIC PROGRAM FOR A ISENTROPIC SPIKE INL
   1ET//)
600 FORMAT (//30X,3HAMO,10X,5HTHETC,10X,2HBS,10X,5HCOWLA,10X,3HAMF//)
602 FORMAT (23X,1P5E14.5)
604 FORMAT(//33X,41HISENTROPIC COMPRESSION FOCUSES AT THETA =1PE12.5//
   1)
606 FORMAT (//28X,62HSHIFT IN CENTERBODY COEFFICIENTS DUE TO SHOCK IMP
   1INGMENT POINT//)
608 FORMAT (//40X,6HDELX =1PE12.5,2X,6HDELY =1PE12.5//)
610 FORMAT(1H1)
    SHIFTX=XCOWL-COWL(1)
    SHIFTY=YCOWL-1.000
    IF (XCOWL .EQ. 0.0) SHIFTX=0.0
    IF (YCOWL .EQ. 0.0) SHIFTY=0.0
    REG=1.0
    DO 10 I=1,NR
    CALL SHIFT(I,SHIFTX,SHIFTY)
 10 CONTINUE
    DO 12 K=1,50
    DO 12 I=1,50
    CALL EQUK(K,I,0.0,0.0,0.0,0.0)
    CALL EQUI(I,0.0,0.0,0.0,0.0)
 12 CONTINUE
    CONVP=(1.0+(GAM-1.0)/2.0)**(-GAM/(GAM-1.0))
    CONVD=(1.0+(GAM-1.0)/2.0)**(-1.0/(GAM-1.0))
    SIGSQ=(GAM-1.0)/(GAM+1.0)
    IF (NDIM .GT. 2) GO TO 14
    CALL PSA(AMO,0.0,THETC)
    UC=UK
    VC=VK
    BS=BETAK
    DELC=TAN(THETC)
    DELS=TAN(BS)
    GO TO 16
 14 CALL CSA(BETAE)
    U1=UC
    V1=VC
    DEL1=-1.0/DELC
 16 CALL OSWR(AMO,BS)
    IREV=1
    RECV(1,IREV)=RECOV
    RECV(2,IREV)=RECOV
    FACTR=RECOV*CONVP
    WRITE (6,610)
    IF (NDIM .GT. 2) GO TO 18
    WRITE (6,500)
    GO TO 20
 18 WRITE (6,502)
 20 WRITE (6,600)
 22 WRITE (6,602) AMO,THETC,BS,COWLA,AMF
    IF (OFF .EQ. 0.0) GO TO 26
    XCOWL=COWL(1)
    YCOWL=R(1,1)+R(1,2)*XCOWL+R(1,3)*XCOWL**2+R(1,4)*XCOWL**3
 24 CALL OUT
    DELQ=DELP*DELS
```

```
    CALL DATUM(START,DELQ)
    GU TO 34
 26 IF (FOCUS .GT. 0.0) GO TO 28
    THETAP=BS
    GO TO 30
 28 DELS=TAN(THETAP)
 30 WRITE (6,604) THETAP
    K=1
    I=1
    X(K,I)=1.0/DELS
    Y(K,I)=1.0
    CALL DRAG(THETAP,Y(K,I),ND)
    CALL SPIKE(K,I,M,START,AMF,1.0,PRINT)
    IREF=1
    IF (NS .GT. 1) GO TO 32
    XS=X(1,1)
    YS=Y(1,1)
    DELSP=V(1,1)/U(1,1)
    XP=XS+3.0
    REG=-1.0
    CALL CURVE(XS,YS,DELSP,XP,0.0,0.0,1.0,NS)
    NS=NS+1
 32 IF (NR .GT. 1) GO TO 34
    XS=X(KREF,IREF)
    YS=Y(KREF,IREF)
    DELSP=TAN(COWLA)
    XP=XS+3.0
    REG=1.0
    CALL CURVE(XS,YS,DELSP,XP,0.0,0.0,1.0,NR)
    NR=NR+1
 34 KR=1
    IR=1
    REG=-1.0
    EXTEND=ISHK
    BODY(ISHK)=BODY(ISHK)*EXTEND
 36 CALL SHAPE(KR,IR,-1.0,0.0,PRINT)
    IREF=1
    IF (OFF .EQ. 0.0) GO TO 42
 38 CALL PUT(XCOWL,YCOWL)
 42 BODY(ISHK)=BODY(ISHK)/EXTEND
    DELTA=COWLA-ATAN(V(KREF,IREF)/U(KREF,IREF))
    IF (ABS(DELTA) .LT. 0.010) DELTA=0.0
    IF (DELTA .GT. 0.0) DELTA=0.0
    IF (NSHK .EQ. 0) GO TO 64
    IF (NSHK .GT. 1) GO TO 44
    SETP=OFF
    SETQ=OFF
    GO TO 46
 44 SETP=1.0
    SETQ=0.0
 46 CALL MATRIX
    KR=KREF
    IR=IREF
    DO 48 J=1,NSHK
    SIGN=(-1.0)**(J+1)
    REG=SIGN
    CALL SHOCK(KR,IR,DELTA,SIGN,SETP,SETQ)
    KR=KREF
    IR=1
    STFR=PSI
```

```
      DELTA=-DELA
   48 CONTINUE
      IF (NSHK .GT. 1) GO TO 64
      IF (OFF .EQ. 0.0) GO TO 60
      IF (ISHK .EQ. 1) GO TO 56
      XSHK=BODY(ISHK)
      YSHK=S(ISHK,1)+S(ISHK,2)*XSHK+S(ISHK,3)*XSHK**2+S(ISHK,4)*XSHK**3
      DELX=X(1,1)-XSHK
      DELY=Y(1,1)-YSHK
      IF (ABS(DELX) .GT. ERROR) GO TO 50
      IF (ABS(DELY) .LT. ERROR) GO TO 56
   50 REG=-1.0
      DO 52 I=ISHK,NS
      CALL SHIFT(I,DELX,DELY)
   52 CONTINUE
      WRITE (6,610)
      WRITE (6,606)
      WRITE (6,608) DELX,DELY
   54 CALL OUT
      IF (DELX .LT. 0.0) GO TO 56
      GO TO 66
   56 IF (NTHR .EQ. 0) GO TO 64
      DO 58 J=1,NTHR
      REG=(-1.0)**(J)
      CALL SHAPE(1,1,REG,1.0,1.0)
   58 CONTINUE
      GO TO 66
   60 IF (NTHR .EQ. 0) GO TO 64
      KR=KREF
      IR=IREF
      READ (5,408) (AMT(J),THR(J),ANG(J),NIS(J), J=1,NTHR)
      DO 62 J=1,NTHR
      CALL SURF(KR,IR,NIS(J),AMT(J),THR(J),ANG(J),STFR,1.0)
      KR=KREF
      IR=1
   62 CONTINUE
   64 WRITE (6,610)
      CALL OUT
   66 GO TO 5
      END
```

```
$IBFTC EQUKS    LIST,REF,DECK,DEBUG
C
C      EQUIVALENCE SUBROUTINE
C
       SUBROUTINE EQUK(KR,IR,XA,YA,UA,VA)
       COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1        Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2        XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3        X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4        U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5        XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6        REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
       X(KR,IR)=XA
       Y(KR,IR)=YA
       U(KR,IR)=UA
       V(KR,IR)=VA
       IF (X(KR,IR) .EQ. 0.0) GO TO 10
       CALL CFPR(X(KR,IR),Y(KR,IR),U(KR,IR),V(KR,IR))
    10 RETURN
       END
```

```
$IBFTC EQUIS    LIST,REF,DECK,DEBUG
C
C      EQUIVALENCE SUBROUTINE
C
       SUBROUTINE EQUI(IR,XA,YA,UA,VA)
       COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1        Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2        XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3        X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4        U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5        XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6        REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
       XB(IR)=XA
       YB(IR)=YA
       UB(IR)=UA
       VB(IR)=VA
       IF (XB(IR) .EQ. 0.0) GO TO 10
       CALL CFPR(XB(IR),YB(IR),UB(IR),VB(IR))
    10 RETURN
       END
```

```
$IBFTC CURVES  LIST,REF,DECK,DEBUG
C
C     ROUTINE FOR CALCULATING COEFFICIENTS OF SURFACE CONTOURS
C
      SUBROUTINE CURVE (X1,Y1,SLP1,X2,Y2,SLP2,EXP,INDEX)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1      Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2      XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3      X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4      U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5      XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6      REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      Z=X2-X1
      IF (Z .GT. 0.0) GO TO 8
      A0=0.0
      A1=0.0
      A2=0.0
      A3=0.0
      GO TO 14
    8 A0=Y1
      A1=SLP1
      DYR=SLP2
      IF (EXP .EQ. 3.0) GO TO 10
      IF (EXP .EQ. 2.0) GO TO 12
      A2=0.0
      A3=0.0
      GO TO 14
   10 FUN1=Y2-Y1-A1*Z
      FUN2=DYR-A1
      DELTA=Z**4
      A2=(3.0*FUN1-Z*FUN2)*Z**2/DELTA
      A3=(Z*FUN2-2.0*FUN1)*Z/DELTA
      GO TO 14
   12 A2=(DYR-A1)/(2.0*Z)
      A3=0.0
   14 P0=A0-A1*X1+A2*X1**2-A3*X1**3
      P1=A1-2.0*A2*X1+3.0*A3*X1**2
      P2=A2-3.0*A3*X1
      P3=A3
      IF (REG .EQ. -1.0) GO TO 18
      R(INDEX,1)=P0
      R(INDEX,2)=P1
      R(INDEX,3)=P2
      R(INDEX,4)=P3
      COWL(INDEX)=X1
      COWL(INDEX+1)=X2
      DO 16 J=1,4
      R(INDEX+1,J)=0.0
   16 CONTINUE
      GO TO 22
   18 S(INDEX,1)=P0
      S(INDEX,2)=P1
      S(INDEX,3)=P2
      S(INDEX,4)=P3
      BODY(INDEX)=X1
      BODY(INDEX+1)=X2
      DO 20 J=1,4
      S(INDEX+1,J)=0.0
   20 CONTINUE
   22 RETURN
      END
```

```
$IBFTC SHIFS   LIST,REF,DECK,DEBUG
C
C
C
      SUBROUTINE SHIFT(I,DELX,DELY)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1      Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2      XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3      X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4      U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5      XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      IF (REG .EQ. -1.0) GO TO 10
      P1=R(I,1)+DELY-R(I,2)*DELX+R(I,3)*DELX**2-R(I,4)*DELX**3
      P2=R(I,2)-2.0*R(I,3)*DELX+3.0*R(I,4)*DELX**2
      P3=R(I,3)-3.0*R(I,4)*DELX
      P4=R(I,4)
      R(I,1)=P1
      R(I,2)=P2
      R(I,3)=P3
      R(I,4)=P4
      COWL(I)=COWL(I)+DELX
      GO TO 12
   10 P1=S(I,1)+DELY-S(I,2)*DELX+S(I,3)*DELX**2-S(I,4)*DELX**3
      P2=S(I,2)-2.0*S(I,3)*DELX+3.0*S(I,4)*DELX**2
      P3=S(I,3)-3.0*S(I,4)*DELX
      P4=S(I,4)
      S(I,1)=P1
      S(I,2)=P2
      S(I,3)=P3
      S(I,4)=P4
      BODY(I)=BODY(I)+DELX
   12 RETURN
      END


$IBFTC OUTR    LIST,REF,DECK,DEBUG
C
C     ROUTINE FOR WRITING OUT TABLE OF CONTOUR COEFFICIENTS
C
      SUBROUTINE OUT
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1      Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2      XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3      X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4      U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5      XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//45X,26HTABLE OF COWL COEFFICIENTS//)
  102 FORMAT (//30X,2HR1,12X,2HR2,12X,2HR3,12X,2HR4,11X,4HCOWL//)
  104 FORMAT (23X,1P5E14.5//)
  106 FORMAT (//43X,32HTABLE OF CENTERBODY COEFFICIENTS//)
  108 FORMAT (//30X,2HS1,12X,2HS2,12X,2HS3,12X,2HS4,10X,4HBODY//)
      WRITE (6,100)
      WRITE (6,102)
      DO 10 I=1,NR
      WRITE (6,104) (R(I,K),K=1,4),COWL(I)
   10 CONTINUE
      WRITE (6,106)
      WRITE (6,108)
      DO 12 I=1,NS
      WRITE (6,104) (S(I,K),K=1,4),BODY(I)
   12 CONTINUE
      RETURN
      END
```

```
$IBFTC FINDS   LIST,REF,DECK,DEBUG
C
C
C
      SUBROUTINE FIND(POINT)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//25X,52HCOWL POINT LIES OUTSIDE RANGE OF INPUT CONTOURS,
     1X =1PE12.5//)
  102 FORMAT (//25X,52HBODY POINT LIES OUTSIDE RANGE OF INPUT CONTOURS,
     1X =1PE12.5//)
      MR=NR-1
      MS=NS-1
      IF (REG .EQ. -1.0) GO TO 18
      REF=COWL(1)-POINT
      IF (REF .GT. 0.0) GO TO 14
      DO 10 I=1,MR
      REF=COWL(I+1)-POINT
      IF (REF .GT. 0.0) GO TO 12
   10 CONTINUE
   12 IF (REF .GT. 0.0) GO TO 16
   14 WRITE (6,100) POINT
      CALL CFPR(0.0,0.0,0.0,0.0)
      CALL EXIT
   16 ICOWL=I
      GO TO 28
   18 REF=BODY(1)-POINT
      IF (REF .GT. 0.0) GO TO 24
      DO 20 I=1,MS
      REF=BODY(I+1)-POINT
      IF (REF .GT. 0.0) GO TO 22
   20 CONTINUE
   22 IF (REF .GT. 0.0) GO TO 26
   24 WRITE (6,102) POINT
      CALL CFPR(0.0,0.0,0.0,0.0)
      CALL EXIT
   26 IBODY=I
   28 RETURN
      END
```

```
$IBFTC CCRAS    LIST,REF,DECK,DEBUG
C
C      SUBROUTINE FOR STARTING SHOCK SOLUTION
C
      SUBROUTINE CCRA(KQ,IQ,XP,YP,UP,VP,ANGLE,SIGN)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//39X,47HUNABLE TO OBTAIN CONVERGENCE IN SUBROUTINE CCRA//
     1)
  102 FORMAT (//32X,4HXREF,10X,4HYREF,10X,3HAMR,11X,4HQREF,9X,6HTHETAR//
     1)
  104 FORMAT (28X,1P5E14.5//)
      XREF=XP
      YREF=YP
      DELTA=ANGLE
      CALL CFPR(XP,YP,UP,VP)
      AMP=AM
      THETAP=THETA
      QREF=Q
      ITER=0
   10 TEST=QREF
      ITER=ITER+1
      CALL PSA(AMP,THETAP,DELTA)
      UREF=UK
      VREF=VK
   12 CALL CCRE(XREF,YREF,UREF,VREF,SIGN)
      XJ=XK
      YJ=YK
      UJ=UK
      VJ=VK
      IF (ABS(ANGLE) .LT. 0.001) GO TO 22
   14 CALL CCRE(XREF,YREF,UREF,VREF,-SIGN)
      XQ=XK
      YQ=YK
      TANQ=VK/UK
      DU=(UJ-U(KQ,IQ))
      DX=(XJ-X(KQ,IQ))
      DY=(YJ-Y(KQ,IQ))
      DS=SQRT(DX**2+DY**2)
      DUDS=DU/DS
      DELP=XQ-X(KQ,IQ)
      DELQ=YQ-Y(KQ,IQ)
      DELR=SQRT(DELP**2+DELQ**2)
      UQ=U(KQ,IQ)+DUDS*DELR
      VQ=TANQ*UQ
   16 CALL CCRC(XJ,YJ,UJ,VJ,XQ,YQ,UQ,VQ,SIGN)
      XREF=XP
      YREF=YP
      UREF=UK
      VREF=VK
   18 CALL CFPR(XREF,YREF,UREF,VREF)
      QREF=Q
      DELTA=THETA-THETAP
      IF (ABS(DELTA) .LT. ERROR) DELTA=0.0
      IF (ITER .LT. 25) GO TO 20
```

```
      WRITE (6,100)
      WRITE (6,102)
      CALL CFPR(XREF,YREF,UREF,VREF)
      WRITE (6,104) XREF,YREF,AM,Q,THETA
      CALL EXIT
   20 IF (ABS(TEST-QREF) .GT. ERROR*QREF) GO TO 10
   22 XK=XREF
      YK=YREF
      UK=UREF
      VK=VREF
      RETURN
      END
```

```
$IBFTC CCRBS   LIST,REF,DECK,DEBUG
C
C     SUBROUTINE FOR CALCULATING SHOCK SOLUTION
C
      SUBROUTINE CCRB(KQ,IQ,XP,YP,UP,VP,ANGLE,SIGN)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//39X,47HUNABLE TO OBTAIN CONVERGENCE IN SUBROUTINE CCRB//
     1)
  102 FORMAT (//34X,4HXREF,10X,4HYREF,10X,3HAMR,11X,4HQREF,9X,6HTHETAR//
     1)
  104 FORMAT (28X,1P5E14.5//)
      XREF=XP
      YREF=YP
      IP=IQ
      DELTA=ANGLE
      CALL CFPR(XP,YP,UP,VP)
      AMP=AM
      THETAP=THETA
      QREF=Q
      ITER=0
   10 TEST=QREF
      ITER=ITER+1
      CALL PSA(AMP,THETAP,DELTA)
      UREF=UK
      VREF=VK
      CALL CFPR(XREF,SIGN*YREF,UREF,SIGN*VREF)
      SLOPE=SIGN*B
   12 CALL CCRC(XREF,YREF,UREF,VREF,X(KQ,IQ),Y(KQ,IQ),U(KQ,IQ),
     1         V(KQ,IQ),SIGN)
      XJ=XK
      YJ=YK
      UJ=UK
      VJ=VK
      IF (ABS(ANGLE) .LT. 0.001) GO TO 22
      IF ((XJ-XREF) .LT. 0.0) GO TO 22
```

```
   14 CALL BOUND(KQ,1,KQ,IP,XREF,YREF,SLOPE,SIGN)
      XQ=XK
      YQ=YK
      UQ=UK
      VQ=VK
      REF=X(KQ,IP)-XQ
      IF (REF .GE. 0.0) GO TO 16
      IP=IP+1
      GO TO 14
   16 CALL CCRC(XJ,YJ,UJ,VJ,XQ,YQ,UQ,VQ,SIGN)
      XREF=XP
      YREF=YP
      UREF=UK
      VREF=VK
   18 CALL CFPR(XREF,YREF,UREF,VREF)
      QREF=Q
      DELTA=THETA-THETAP
      IF (ABS(DELTA) .LT. ERROR) DELTA=0.0
      IF (ITER .LT. 100) GO TO 20
      WRITE (6,100)
      WRITE (6,102)
      CALL CFPR(XREF,YREF,UREF,VREF)
      WRITE (6,104) XREF,YREF,AM,Q,THETA
      CALL OUT
      CALL CFPR(0.0,0.0,0.0,0.0)
      CALL EXIT
   20 IF (ABS(TEST-QREF) .GT. ERROR*QREF) GO TO 10
   22 XK=XREF
      YK=YREF
      UK=UREF
      VK=VREF
      RETURN
      END
```

```
$IBFTC CCRCS   LIST,REF,DECK,DEBUG
C
C     CONIC CHARACTERISTIC ROUTINE C
C
      SUBROUTINE CCRC(X2,Y2,U2,V2,X3,Y3,U3,V3,SIGN)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//39X,47HUNABLE TO OBTAIN CONVERGENCE IN SUBROUTINE CCRC//
     1)
      ITER=0
      XI=X2
      YI=SIGN*Y2
      UI=U2
      VI=SIGN*V2
      XF=X3
      YF=SIGN*Y3
      UF=U3
      VF=SIGN*V3
```

```
    8 CALL CFPR (XI,YI,UI,VI)
      AI = A
      BI = B
      DI = D
      EI = E
      FI = F
      GI = G
      QI=Q
      CALL CFPR(XF,YF,UF,VF)
      AJ = A
      BJ = B
      DJ = D
      EJ = E
      FJ = F
      GJ = G
      QJ=Q
      AK = .5*(AI + AJ)
      BK = .5*(BI + BJ)
      DK = .5*(DI + DJ)
      EK = .5*(EI + EJ)
      FK = .5*(FI + FJ)
      GK = .5*(GI + GJ)
      QK= .5*(QI+QJ)
   10 ITER=1+ITER
      A = .5 * (AI + AK)
      B = .5 * (BJ + BK)
      D = .5 * (DI + DK)
      E = .5 * (EJ + EK)
      F = .5 * (FI + FK)
      G = .5 * (GJ + GK)
      XK= ((YF-YI)+A*XI-B*XF)/(A-B)
      YK= YF+B*(XK-XF)
      VK= ((UF-UI)+D*VI-E*VF+G*(XK-XF)-F* (XK-XI))/(D-E)
      UK= UF+E*(VK-VF)+G*(XK-XF)
      TEST=QK
      CALL CFPR (XK,YK,UK,VK)
      AK = A
      BK = B
      DK = D
      EK = E
      FK = F
      GK = G
      QK=Q
      IF (ITER .LT. 25) GO TO 12
      WRITE (6,100)
      CALL CFPR(0.0,0.0,0.0,0.0)
   12 IF (ABS(TEST-QK) .GT. ERROR*QK) GO TO 10
      XK=XK
      YK=SIGN*YK
      UK=UK
      VK=SIGN*VK
   14 RETURN
      END
```

```
$IBFTC CCRES   LIST,REF,DEBUG
C
C     CHARACTERISTIC ROUTINE FOR A CURVED SURFACE
C
      SUBROUTINE CCRE(X3,Y3,U3,V3,SIGN)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//39X,38HIMPROPER CALCULATION PERFORMED IN CCRE//)
  102 FORMAT (//22X,2HP1,12X,2HP2,12X,2HP3,12X,2HP4,12X,2HP5,12X,2HP6//)
  104 FORMAT (15X,1P6E14.5//)
  106 FORMAT (//48X,7HPOINT =1PE12.5//)
  108 FORMAT (//39X,47HUNABLE TO OBTAIN CONVERGENCE IN SUBROUTINE CCRE//
     1)
      ITER=0
      XREF=X3
      XC=X3
      YC=SIGN*Y3
      UC=U3
      VC=SIGN*V3
      CALL CFPR(XC,YC,UC,VC)
      AK=A
      DK=D
      FK=F
      AREF=AK
      UREF=UC
      VREF=VC
      QREF=Q
      POINT=X3
   10 ITER=1+ITER
      CALL FIND(XREF)
      IF (REG .GT. 0.0) GO TO 12
      A0=SIGN*S(IBODY,1)
      A1=SIGN*S(IBODY,2)
      A2=SIGN*S(IBODY,3)
      A3=SIGN*S(IBODY,4)
      GO TO 14
   12 A0=SIGN*R(ICOWL,1)
      A1=SIGN*R(ICOWL,2)
      A2=SIGN*R(ICOWL,3)
      A3=SIGN*R(ICOWL,4)
   14 B0=A0-YC+AREF*XC
      B1=A1-AREF
      CALL ROOT3(A3,A2,B1,B0)
      M=0
      DO 16 I=1,3
      J=2*I-1
      K=2*I
      P(J)=P(J)-POINT
      IF (ABS(P(K)) .GT. ERROR) GO TO 16
      M=M+1
      PSIR(M)=P(J)
   16 CONTINUE
      XMIN=10.0
      IF (M .GT. 0) GO TO 18
      WRITE (6,100)
```

```
      WRITE (6,106) POINT
      WRITE (6,102)
      WRITE (6,104) (P(I),I=1,6)
      CALL CFPR(0.0,0.0,0.0,0.0)
      CALL EXIT
   18 DO 20 I=1,M
      DIST=ABS(PSIR(I))
      XMIN=AMIN1(XMIN,DIST)
   20 CONTINUE
      DO 22 I=1,M
      DIST=ABS(PSIR(I))
      IF (DIST .EQ. XMIN) GO TO 24
   22 CONTINUE
   24 XREF=PSIR(I)+POINT
      YREF=A0+A1*XREF+A2*XREF**2+A3*XREF**3
      CALL CFPR(XREF,YREF,UREF,VREF)
      TEST=Q
      DREF=0.5*(DK+D)
      FREF=0.5*(FK+F)
      SLOPE=A1+2.0*A2*XREF+3.0*A3*XREF**2
      UREF=(UC-DREF*VC+FREF*(XREF-XC))/(1.0-DREF*SLOPE)
      VREF=UREF*SLOPE
      CALL CFPR(XREF,YREF,UREF,VREF)
      QREF=Q
      AREF=0.5*(AK+A)
      IF (ITER .LT. 25) GO TO 25
      WRITE (6,108)
      CALL CFPR(0.0,0.0,0.0,0.0)
   25 IF (ABS(TEST-QREF) .GT. ERROR*QREF) GO TO 10
      XK=XREF
      YK=SIGN*YREF
      UK=UREF
      VK=SIGN*VREF
   26 RETURN
      END
```

```
$IBFTC MATRIX  LIST,REF,DECK,DEBUG
C
C     MATRIX INVERSION SUBROUTINE
C
      SUBROUTINE MATRX
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1          Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2          XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3          X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4          U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5          XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6          REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      DO 12 K=1,50
      DO 10 I=1,50
      IF (I .LE. K) GO TO 10
      CALL EQUK(K,I,0.0,0.0,0.0,0.0)
   10 CONTINUE
```

```
   12 CONTINUE
      DO 16 K=1,50
      DO 14 I=1,50
      IF (I .EQ. K) GO TO 16
      CALL EQUK(I,K,X(K,I),Y(K,I),U(K,I),V(K,I))
   14 CONTINUE
   16 CONTINUE
      DO 20 I=1,50
      DO 18 K=1,50
      IF (K .LE. I) GO TO 18
      CALL EQUK(K,I,0.0,0.0,0.0,0.0)
   18 CONTINUE
   20 CONTINUE
      RETURN
      END
```

```
$IBFTC CORP    LIST,REF,DECK,DEBUG
C
C     CALCULATION OF LOCAL STREAMLINE CONDITIONS
C
      SUBROUTINE CORE(K,I,SIGN)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25).NR,NS,ICOWL,IBODY
      CALL CFPR(X(K,I),Y(K,I),U(K,I),V(K,I))
      PRHO=PRES*FACTR
      THETAR=THETA
   10 FACTA=(GAM-1.0)/2.0
      FACTB=(GAM+1.0)/2.0
      EXP=-(GAM-1.0)/GAM
      CONVP=(1.0+FACTA)**(1.0/EXP)
      RAVE=FACTR/CONVP
      KSIGN=SIGN
      KREV=(3+KSIGN)/2
      RATIO=RAVE/RECV(KREV,IREV)
      IF (ABS(RATIO-1.000) .LE. ERROR) GO TO 14
      PRH=(PRHO/RAVE)*RATIO
      AMRSQ=((PRH)**EXP-1.0)/FACTA
      QRSQ=FACTB*AMRSQ/(1.0+FACTA*AMRSQ)
      QR=SQRT(QRSQ)
      UR=QR*COS(THETAR)
      VR=QR*SIN(THETAR)
      CALL EQUK(K,I,X(K,I),Y(K,I),UR,VR)
      IF (I .LT. K) GO TO 14
   12 CALL EQUI(I,XB(I),YB(I),UR,VR)
   14 RETURN
      END
```

```
$IBFTC ROOTE    LIST,REF,DECK,DEBUG
C
C      DETERMINATION OF THE ROOTS OF A CUBIC EQUATION
C
      SUBROUTINE ROOT3(Z1,Z2,Z3,Z4)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      DIMENSION AUG(10),ALPHI(10),BETI(10),RG(10)
  100 FORMAT (//43X,39HIMPROPER IMPUT DATA TO SUBROUTINE ROOT3//)
  102 FORMAT (//42X,2HZ1,12X,2HZ2,12X,2HZ3,12X,2HZ4//)
  104 FORMAT (34X,1P4E14.5//)
      IF (ABS(Z1) .LT. 0.00001) GO TO 26
      R2=Z2/Z1
      R3=Z3/Z1
      R4=Z4/Z1
      A=(3.0*R3-R2**2)/3.0
      B=(2.0*R2**3-9.0*R2*R3+27.0*R4)/27.0
      TEST=B**2/4.0+A**3/27.0
      EXAM=-B/2.0
      CHECK=SQRT(ABS(TEST))
      IF (CHECK .GT. ERROR) GO TO 8
      TEST=0.0
    8 CALL ROOTN(TEST,0.0,2)
      TESTR=P(1)
      TESTI=P(2)
      DO 18 I=1,2
      SIGN=(-1)**(I+1)
      PARTR=EXAM+SIGN*TESTR
      PARTI=SIGN*TESTI
      CALL ROOTN(PARTR,PARTI,3)
      IF (TEST .GT. 0.0) GO TO 12
      DO 10 L=1,2
      SIGN=(-1)**(L+1)
      J=2*L-1
      K=2*L
      AUG(J)=P(1)
      AUG(K)=SIGN*P(2)
   10 CONTINUE
      GO TO 20
   12 J=2*I-1
      K=2*I
      DO 14 L=1,3
      M=2*L-1
      N=2*L
      IF (ABS(P(N)) .LE. ERROR) GO TO 16
   14 CONTINUE
   16 AUG(J)=P(M)
      AUG(K)=P(N)
   18 CONTINUE
   20 CALL ROOTN(-3.0,0.0,2)
      DO 22 I=1,3
      J=2*I-1
      K=2*I
      ALPHI(J)=AUG(1)+AUG(3)
      ALPHI(K)=AUG(2)+AUG(4)
```

```
      BETI(J)=AUG(1)-AUG(3)
      BETI(K)=AUG(2)-AUG(4)
      RG(J)=P(1)
      RG(K)=P(2)
   22 CONTINUE
      DO 24 I=1,3
      J=2*I-1
      K=2*I
      DEL1=(I-1)
      DEL2=(I-2)
      DELTA=DEL2/2.0+DEL1-6.0*DEL1*DEL2/4.0
      DELTB=DEL2/2.0
      P(J)=-R2/3.0+DELTA*ALPHI(J)+DELTB*(BETI(J)*RG(J)-BETI(K)*RG(K))
      P(K)=DELTA*ALPHI(K)+DELTB*(BETI(K)*RG(J)+BETI(J)*RG(K))
   24 CONTINUE
      GO TO 38
   26 IF (ABS(Z2) .LT. 0.00001) GO TO 32
      R3=Z3/(2.0*Z2)
      R4=Z4/Z2
      TEST=R3**2-R4
      CALL ROOTN(TEST,0.0,2)
      DO 28 I=1,2
      J=2*I-1
      K=2*I
      RG(J)=P(J)
      RG(K)=P(K)
   28 CONTINUE
      DO 30 I=1,3
      J=2*I-1
      K=2*I
      SIGN=(-1)**(I+1)
      DEL1=(I-1)
      DEL2=(I-2)
      DELTA=1.0-DEL1*DEL2/2.0
      P(J)=DELTA*(-R3+SIGN*RG(1))
      P(K)=DELTA*SIGN*RG(2)
   30 CONTINUE
      GO TO 38
   32 IF (Z4.EQ. 0.0) GO TO 36
      DO 34 I=1,3
      J=2*I-1
      K=2*I
      DEL1=(I-1)
      DEL2=(I-2)
      DELTA=1.0-DEL1+DEL1*DEL2/2.0
      P(J)=-DELTA*(Z4/Z3)
      P(K)=0.0
   34 CONTINUE
      GO TO 38
   36 WRITE (6,100)
      WRITE (6,102)
      WRITE (6,104) Z1,Z2,Z3,Z4
      CALL OUT
      CALL EXIT
   38 RETURN
      END
```

```
$IBFTC ROOTI   LIST,REF,DECK,DEBUG
C
C     DETERMINATION OF THE ROOTS OF A COMPLEX NUMBER
C
      SUBROUTINE ROOTN(A,B,N)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      PI=3.1415927
      EN=N
      EXP=1.0/EN
      TEST=SQRT(A**2+B**2)
      IF (TEST .EQ. 0.0) GO TO 16
      AMAG=TEST**EXP
      IF (ABS(A) .GT. 0.0) GO TO 8
      SIGN=B/ABS(B)
      AMAG=ABS(B)**EXP
      AUG=SIGN*PI/2.0
      GO TO 12
    8 IF (ABS(B) .GT. 0.0) GO TO 10
      SIGN=A/ABS(A)
      AMAG=ABS(A)**EXP
      AUG=(1.0-SIGN)*PI/2.0
      GO TO 12
   10 ANGLE=ATAN(ABS(B/A))
      SIGNA=A/ABS(A)
      SIGNB=B/ABS(B)
      IF (A .GT. 0.0) GO TO 11
      AUG=PI-SIGNB*ANGLE
      GO TO 12
   11 AUG=(1.0-SIGNB)*PI+SIGNB*ANGLE
   12 DO 14 I=1,N
      J=2*I-1
      K=2*I
      AUGI=(I-1)
      TAU=EXP*(AUG+2.0*AUGI*PI)
      P(J)=AMAG*COS(TAU)
      P(K)=AMAG*SIN(TAU)
   14 CONTINUE
      GO TO 20
   16 DO 18 I=1,N
      J=2*I-1
      K=2*I
      P(J)=0.0
      P(K)=0.0
   18 CONTINUE
   20 RETURN
      END
```

```
$IBFTC CSAR    LIST,REF,DECK,DEBUG
C
C     SUBROUTINE FOR CALCULATING CONIC SHOCK ANGLE
C
      SUBROUTINE CSA(BETAE)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      DIMENSION UU(3),VV(3),DEL(3),ABETA(3),TEST(3),DBT(3)
      SIGSQ=(GAM-1.0)/(GAM+1.0)
      Q0=SQRT(AMO**2/(1.0+SIGSQ*(AMO**2-1.0)))
      DELC=TAN(THETC)
   10 DO 12 I=1,3
      SIGN=I-2
      ABETA(I)=BETAE+SIGN*DELB
      U1=(1.0-SIGSQ)*Q0*COS(ABETA(I))**2+1.0/Q0
      V1=(Q0-U1)*COS(ABETA(I))/SIN(ABETA(I))
      DEL1=V1/(U1-Q0)
      CALL CFF(1.0,DELC)
      UU(I)=UA
      VV(I)=VA
      DEL(I)=DELA
      TEST(I)=ABS(VV(I)/UU(I))-ABS(DELC)
      IF (ABS(TEST(I)) .LT. ERROR*ABS(DELC)) GO TO 16
      IF (ABS(DELB) .EQ. 0.0) GO TO 16
   12 CONTINUE
      DO 14 I=1,2
      DBT(I)=(ABETA(I+1)-ABETA(I))/(TEST(I+1)-TEST(I))
   14 CONTINUE
      DBDT=(DBT(1)+DBT(2))/2.0
      DELT1=TEST(3)-TEST(1)
      DELT2=TEST(3)-TEST(2)
      D2BDT2=((ABETA(3)-ABETA(2))-DBDT*DELT2)/(DELT1*DELT2)
      DELB=-DBDT*TEST(2)+D2BDT2*TEST(1)*TEST(2)
      BETAE=BETAE+2.0*DELB
      GO TO 10
   16 QC=SQRT(UU(I)**2+VV(I)**2)
      UC=QC*COS(THETC)
      VC=QC*SIN(THETC)
      U1=UC
      V1=VC
      DEL1=-1.0/DELC
      BS=ABETA(I)
      DELS=TAN(BS)
      RETURN
      END
```

```
$IBFTC CFFR    LIST,REF,DECK,DEBUG
C
C     FIELD CALCULATIONS IN A CONIC FLOW FIELD
C
      SUBROUTINE CFF(XP,YP)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      DELTA=YP/XP
      TEST=ABS(DELTA)-ABS(1.0/DEL1)
      IF (ABS(TEST) .GT. ABS(DELTA)*ERROR) GO TO 10
      UA=U1
      VA=V1
      DELA=DEL1
      GO TO 16
   10 SIGNT=TEST/ABS(TEST)
      UP=U1
      VP=V1
      VUP=DEL1
      DELU=SIGNT*DELU1
   12 CALL RUNGE(UP,VP,VUP,DELU,SIGSQ)
      PEST=TEST
      SIGNP=PEST/ABS(PEST)
      TEST=ABS(DELTA)-ABS(1.0/VUP)
      IF (ABS(TEST) .LT. ABS(DELTA)*ERROR) GO TO 14
      SIGNT=TEST/ABS(TEST)
      IF (SIGNT .EQ. SIGNP) GO TO 12
      VUU=FUNV(UP,VP,VUP,SIGSQ)
      DELU=-0.5*((1.0/DELTA)+VUP)/VUU
      IF (DELU .EQ. 0.0) GO TO 14
      GO TO 12
   14 UA=UP
      VA=VP
      DELA=VUP
      U1=UP
      V1=VP
      DEL1=VUP
   16 RETURN
      END
```

```
$IBFTC RUNGK   LIST,REF,DECK,DEBUG
C
C     RUNGE KUTTA INTEGRATION OF TAYLOR-MACCOLLI EQUATION
C
      SUBROUTINE RUNGE(UP,VP,VUP,DU,SIGSQ)
      REAL K1,K2,K3,K4
      K1=DU*FUNV(UP,VP,VUP,SIGSQ)
      K2=DU*FUNV(UP+DU/2.0,VP+DU/2.0*VUP+K1*DU/8.0,VUP+K1/8.0,SIGSQ)
      K3=DU*FUNV(UP+DU/2.0,VP+DU/2.0*VUP+K1*DU/8.0,VUP+K2/8.0,SIGSQ)
      K4=DU*FUNV(UP+DU,VP+DU*VUP+K3*DU/2.0,VUP+K3,SIGSQ)
      UP=UP+DU
      VP=VP+DU*(VUP+(K1+K2+K3)/6.0)
      VUP=VUP+(K1+2.0*K2+2.0*K3+K4)/6.0
      RETURN
      END
```

```
$IBFTC FUNVP    LIST,REF,DECK,DEBUG
C
C     TAYLOR-MACCOLLI EQUATION
C
      FUNCTION FUNV(UP,VP,VUP,SIGSQ)
      FUN1=(1.0+VUP**2)
      FUN2=(1.0-SIGSQ)*(UP+VP*VUP)**2
      FUN3=(1.0-SIGSQ*(UP**2+VP**2))
      FUNV=(1.0/VP)*(FUN1-FUN2/FUN3)
      RETURN
      END
```

```
$IBFTC CFPRS    LIST,REF,DECK,DEBUG
C
C     COMPRESSIBLE FLOW PARAMETER ROUTINE
C
      SUBROUTINE CFPR (XP,YP,UP,VP)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//36X,45HCONDITIONS OF SUBSONIC FLOW HAVE BEEN REACHED//)
  102 FORMAT (//39X,38HIMPROPER IMPUT DATA TO SUBROUTINE CFPR//)
  104 FORMAT (//18X,1HX,13X,1HY,13X,1HQ,11X,5HTHETA,11X,1HU,13X,1HV,10X,
     18HMACH NO.//)
  106 FORMAT (1HJ)
  108 FORMAT (10X,1P7E14.5//)
  110 FORMAT (//24X,4HXP =1PE12.5,2X,4HYP =1PE12.5,2X,4HUP =1PE12.5,2X,
     14HVP =1PE12.5//)
  112 FORMAT (//36X,43HCOMPUTED FLOW FIELD UP TO THE POINT OF EXIT//)
  114 FORMAT (1H1)
      Q= SQRT(UP**2+VP**2)
      AMSQ=(1.0-SIGSQ)*Q**2/(1.0-SIGSQ*Q**2)
      IF (AMSQ .GT. 0.0) GO TO 10
      WRITE (6,102)
      WRITE (6,110) XP,YP,UP,VP
      GO TO 12
```

```
10 AM=SQRT(AMSQ)
   IF (AM .GT. 1.000) GO TO 16
   WRITE (6,100)
   WRITE (6,110) XP,YP,UP,VP
12 CALL OUT
   WRITE (6,114)
   WRITE (6,112)
   WRITE (6,104)
   DO 14 K=1,50
   WRITE (6,106)
   DO 14 I=1,50
   IF (X(K,I) .EQ. 0.0) GO TO 14
   IF (I .GT. K) GO TO 14
   Q=SQRT(U(K,I)**2+V(K,I)**2)
   AM=SQRT((1.0-SIGSQ)*Q**2/(1.0-SIGSQ*Q**2))
   THETA=ATAN(V(K,I)/U(K,I))
   WRITE (6,108) X(K,I),Y(K,I),Q,THETA,U(K,I),V(K,I),AM
14 CONTINUE
   CALL EXIT
16 AMU= ARSIN(1.0/AM)
   THETA= ATAN(VP/UP)
   C= Q/AM
   A= TAN(THETA+AMU)
   B= TAN(THETA-AMU)
   D= -B
   E= -A
   IF (NDIM .GT. 2) GO TO 18
   F=0.0
   G=0.0
   GO TO 20
18 F = C**2*(VP/YP)/(UP**2-C**2)
   G= F
20 PRES = C**(2.0*GAM/(GAM-1.0))
   DENS= C**(2.0/(GAM-1.0))
   AREA = C**(-2.0/(GAM-1.0))/Q
   ENTH= C**2/(GAM-1.0)
   DYNP= DENS*Q**2/2.0
   CP= PRES/DYNP
   IF (NDIM .GT. 2) GO TO 22
   PSI=DENS*Q/(AM*SIN(THETA+AMU))
   GO TO 24
22 PSI=DENS*Q/(2.0*AM*SIN(THETA+AMU))
24 RETURN
   END
```

```
$IBFTC BOUNS    LIST,REF,DECK,DEBUG
C
C     CONIC CHARACTERISTIC BOUNDARY ROUTINE
C
      SUBROUTINE BOUND(KA,IA,KB,IB,XC,YC,SIGR,SIGN)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      XP=X(KA,IA)
      YP=SIGN*Y(KA,IA)
      UP=U(KA,IA)
      VP=SIGN*V(KA,IA)
      XQ=X(KB,IB)
      YQ=SIGN*Y(KB,IB)
      UQ=U(KB,IB)
      VQ=SIGN*V(KB,IB)
      XR=XC
      YR=SIGN*YC
      SLOPE=SIGN*SIGR
      CALL CFPR(XP,YP,UP,VP)
      AP=A
      DP=D
      FP=F
      CALL CFPR(XQ,YQ,UQ,VQ)
      AREF=.5*(A+AP)
      DREF=0.5*(D+DP)
      FREF=0.5*(F+FP)
      SIGMA=AREF
      DX=(XQ-XP)
      DY=(YQ-YP)
      IF(ABS(SIGR) .GT. ERROR) GO TO 10
      XK=XR
      YK=YP+SIGMA*(XK-XP)
      GO TO 12
   10 XK=(YR-YP+SIGMA*XP-SLOPE*XR)/(SIGMA-SLOPE)
      YK=YR+SLOPE*(XK-XR)
   12 DS=SQRT(DX**2+DY**2)
      DV=(VQ-VP)
      DVDS=DV/DS
      DELSP=SQRT((XK-XP)**2+(YK-YP)**2)
      VKP=VP+DVDS*DELSP
      UKP=UP+DREF*(VKP-VP)+FREF*(XK-XP)
      DELSQ=SQRT((XK-XQ)**2+(YK-YQ)**2)
      VKQ=VQ-DVDS*DELSQ
      UKQ=UQ+DREF*(VKQ-VQ)+FREF*(XK-XQ)
      RATIOP=(DS-DELSP)/DS
      RATIOQ=(DS-DELSQ)/DS
      UK=RATIOP*UKP+RATIOQ*UKQ
      VK=RATIOP*VKP+RATIOQ*VKQ
      YK=SIGN*YK
      VK=SIGN*VK
      RETURN
      END
```

```
$IBFTC CCSRS    LIST,REF,DECK,DEBUG
C
C     CONIC CHARACTERISTIC SURFACE ROUTINE
C
      SUBROUTINE CCSR(STFR,STF,XI,YI,UI,VI,XL,YL,UL,VL)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      CALL CFPR(XI,YI,UI,VI)
      AI=A
      DI=D
      FI=F
      PSII=PSI
      CALL CFPR(XL,YL,UL,VL)
      AL=A
      PSIK=PSI
      A=.5*(AI+AL)
   10 PSIAV=.5*(PSII+PSIK)
      IF (NDIM .GT. 2) GO TO 12
      YK=ABS(YI)+(STFR-STF)/PSIAV
      GO TO 14
   12 YK=SQRT(YI**2+(STFR-STF)/PSIAV)
   14 IF (YI .GT. 0.0) GO TO 16
      YK=-YK
   16 XK=XI+(YK-YI)/A
      RATIO=(YK-YI)/(YL-YI)
      UK=UI+RATIO*(UL-UI)
      VK=VI+RATIO*(VL-VI)
      TEST=PSIK
      CALL CFPR(XK,YK,UK,VK)
      PSIK=PSI
      IF (ABS(TEST-PSIK) .GT. ERROR ) GO TO 10
      RETURN
      END
```

```
$IBFTC PMERS    LIST,REF,DECK,DEBUG
C
C     PRANDTL MEYER EXPANSION ROUTINE
C
      SUBROUTINE PMER(AMP,ANGLE,AMF)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
   10 AK= SQRT(SIGSQ)
      AMFF= SQRT(AMF**2.0-1.0)
      THETAF= (ATAN(AK*AMFF))/AK-(ATAN(AMFF))
      AMPF= SQRT(AMP**2.0-1.0)
      THETAP= (ATAN(AK*AMPF))/AK-(ATAN(AMPF))
      THETAK= ANGLE + (THETAP-THETAF)
      QK= SQRT(AMF**2/(1.0+SIGSQ*(AMF**2-1.0)))
      UK=QK* COS(THETAK)
      VK= QK*SIN(THETAK)
      RETURN
      END
```

```
$IBFTC INCS     LIST,REF,DECK,DEBUG
C
C     INITIAL CHARACTERISTIC ROUTINE
C     FOR 2 DIMENSIONS SET DNIM=2
C     FOR 3 DIMENSIONS SET DNIM=3
C
      SUBROUTINE INCR(XI,YI)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      TANT=YI/XI-DELY
      IF (NDIM .GT. 2) GO TO 10
      CALL PSA(AMO,0.0,THETC)
      CALL CFPR(XI,YI,UK,VK)
      XK=(A*XI-YI)/(A-TANT)
      YK=YI+A*(XK-XI)
      GO TO 12
   10 DELU=-DELU1
      CALL CFF(XI,YI)
      CALL CFPR(XI,YI,UA,VA)
      XK=(A*XI-YI)/(A-TANT)
      YK=YI+A*(XK-XI)
      CALL CFF(XK,YK)
      UK=UA
      VK=VA
      DELU=-DELU1
   12 RETURN
      END
```

```
$IBFTC OBSWRS  LIST,REF,DECK,DEBUG
C
C     OBLIQUE SHOCK WAVE ROUTINE
C
      SUBROUTINE OSWR(AMR,BETAR)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      FAMR=(AMR*SIN(BETAR))**2
      RPRES=(2.0*GAM*FAMR-(GAM-1.0))/(GAM+1.0)
      RDENS=((GAM+1.0)*FAMR)/((GAM-1.0)*FAMR+2.0)
      RECOV=(RDENS)**(GAM/(GAM-1.0))*(1.0/RPRES)**(1.0/(GAM-1.0))
      IF (RECOV .GT. 0.0) GO TO 10
      ENTP=0.0
      GO TO 5
   10 ENTP=-(GAM-1.0)*ALOG(RECOV)
    5 RETURN
      END
```

```
$IBFTC PSAS    LIST,REF,DECK,DEBUG
C
C     PLANE SHOCK ANGLE ROUTINE
C
      SUBROUTINE PSA(AMP,THETP,ALPHA)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV.
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      DIMENSION RG(10),RGI(10)
  100 FORMAT (//31X,62HSUBSONIC CONDITIONS BEHIND THE OBLIQUE SHOCK HAVE
     1 BEEN REACHED//)
  102 FORMAT (//35X,53HTHE SHOCK DETATCHMENT CONDITIONS CAN NOT BE SATIS
     1FIED//)
  104 FORMAT (//30X,64HTHE SHOCK REFLECTION CONDITIONS AT THE WALL CAN N
     1OT BE SATISFIED//)
  106 FORMAT (//25X,71HTHE POINT OF INCIPIENT SHOCK DETATCHMENT HAS BEEN
     1 REACHED, WARNING ONLY//)
  108 FORMAT (//45X,37HIMPROPER INPUT DATA TO SUBROUTINE PSA//)
  110 FORMAT (//46X,3HAMP,10X,5HTHETP,10X,5HALPHA//)
  112 FORMAT (40X,1P3E14.5//)
      SIGSQ = (GAM-1.0)/(GAM+1.0)
      AMPSQ = AMP*AMP
      Q = SQRT(AMPSQ/(1.0+SIGSQ*(AMPSQ-1.0)))
      IF (ABS(ALPHA) .GT. 0.001) GO TO 8
      UK=Q*COS(THETP)
      VK=Q*SIN(THETP)
      BETAK=ARSIN(1.0/AMP)+THETP
      SIGK=TAN(THETP)
      GO TO 5
    8 COEF1=-(AMPSQ+2.0)/AMPSQ
      COEF2=-GAM*SIN(ALPHA)**2
      COEF3=(2.0*AMPSQ+1.0)/(AMPSQ*AMPSQ)
      COEF4=((GAM+1.0)/2.0)**2+(GAM-1.0)/AMPSQ
      COEF5=-COS(ALPHA)**2/(AMPSQ*AMPSQ)
      AA=1.0
      BB=COEF1+COEF2
      CC=COEF3+COEF4*SIN(ALPHA)**2
      DD=COEF5
   10 CALL ROOT3(AA,BB,CC,DD)
      DO 12 I=1,3
      J=2*I
      K=J-1
      RG(I)=P(K)
      RGI(I)=P(J
   12 CONTINUE
      DO 14  I=1,3
   14 IF (ABS(RGI(I)) .GT. ERROR) GO TO 16
      GO TO 17
   16 WRITE (6,102)
      WRITE (6,104)
      WRITE (6,110)
      WRITE (6,112) AMP,THETP,ALPHA
      CALL OUT
      CALL EXIT
   17 DO 18  I=1,3
   18 IF (RG(I).LT.0.) GO TO 20
```

```
      GO TO 21
   20 WRITE (6,108)
      WRITE (6,110)
      WRITE (6,112) AMP,THETP,ALPHA
      CALL OUT
      CALL EXIT
   21 NNR=0
   22 DO 24 I=1,3
      IF (RG(I).GT.0.) GO TO 24
      NNR=NNR+1
   24 CONTINUE
      IF (NNR .GT. 0) GO TO 26
      GO TO 28
   26 WRITE (6,108)
      WRITE (6,110)
      WRITE (6,112) AMP,THETP,ALPHA
      CALL OUT
      CALL EXIT
   28 NGR=0
   30 DO 32  I=1,3
      IF(ABS(RG(1)-RG(I)).GT.0.) GO TO 32
      NGR=NGR+1
   32 CONTINUE
      IF (NGR.EQ.1) GO TO 38
      IF (NGR.EQ.2) GO TO 34
      IF (NGR.EQ.3) GO TO 36
   34 WRITE (6,106)
      WRITE (6,110)
      WRITE (6,112) AMP,THETP,ALPHA
      GO TO 38
   36 WRITE (6,108)
      WRITE (6,110)
      WRITE (6,112) AMP,THETP,ALPHA
      CALL OUT
      CALL EXIT
   38 K=0
      HIGH =0
      DO 40  I=1,3
   40 HIGH=AMAX1(HIGH,RG(I))
      DO 42  I=1,3
      IF (RG(I).EQ.HIGH) GO TO 42
      K=K+1
      RG(K)=RG(I)
   42 CONTINUE
      SEC=0
      DO 44  K=1,2
   44 SEC=AMAX1(SEC,RG(K))
      Z=SQRT(SEC)
      BETAP=ARSIN(Z)
      SINB = SIN(BETAP)
      COSB = COS(BETAP)
      U2 = ((1.0-SIGSQ)*Q*COSB**2)+ 1.0 / Q
      V2 = ((Q - U2)*COSB)/SINB
      QK = SQRT(U2**2 + V2**2)
      ANGLE=THETP+ALPHA
      UK=QK*COS(ANGLE)
      VK=QK*SIN(ANGLE)
      AMK = SQRT(((1.0-SIGSQ)*QK**2)/(1.0-SIGSQ*QK**2))
      IF (AMK.LE.1.0) GO TO 46
      GO TO 48
```

```
   46 WRITE (6,100)
      WRITE (6,104)
      WRITE (6,110)
      WRITE (6,112) AMP,THETP,ALPHA
      CALL OUT
      CALL EXIT
   48 AMUK = ARSIN(1.0/AMK)
      BETAK=BETAP+THETP
      SIGK=TAN(ANGLE)
    5 RETURN
      END
```

```
$IBFTC DRAGS    LIST,REF,DECK,DEBUG
C
C     CALCULATION OF CAPTURE MASS FLOW AND ADDITIVE DRAG IN A CONE FIELD
C
      SUBROUTINE DRAG(THETAL,YC,ND)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1     Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2     XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3     X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4     U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5     XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6     REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//35X,48HINITIAL SHOCK IMPINGES ON INSIDE SURFACE OF COWL/
     1/)
  102 FORMAT (//16X,15HCOWL POSITION =1PE12.5,2X,19HCAPTURE MASS FLOW =1
     1PE12.5,2X,15HADDITIVE DRAG =1PE12.5//)
  104 FORMAT (//40X,39HCONDITIONS ALONG THE CAPTURE STREAMLINE//)
  106 FORMAT (//8X,1HX,13X,1HY,13X,1HQ,11X,5HTHETA,11X,1HU,13X,1HV,9X,
     18HMACH NO.,8X,4HP/HO,10X,4HP/PO//)
  108 FORMAT (1P9E14.5//)
  110 FORMAT (1H1)
      IF (THETAL .LE. BS) GO TO 10
      WRITE (6,100)
      CALL EXIT
   10 IF (THETAL .LT. BS) GO TO 12
      CAPT=1.0
      SUM=0.0
      WRITE (6,102) BS,CAPT,SUM
      GO TO 42
   12 I=1
      EXP=-(GAM)/(GAM-1.0)
      CONV=(1.0+(GAM-1.0)/2.0*AMO**2)**EXP
      CONST=1.0/(GAM*AMO**2)
      XND=ND
      TAU=THETAL
      DELT=(BS-TAU)/XND
      XP=YC/TAN(TAU)
      YP=YC
      U1=UC
      V1=VC
      DEL1=-1.0/DELC
      DELU=DELU1
```

```
      IF (NDIM .GT. 2) GO TO 14
      CALL PSA(AMO,0.0,THETC)
      UP=UK
      VP=VK
      GO TO 16
   14 CALL CFF(XP,YP)
      UP=UA
      VP=VA
   16 CALL CFPR(XP,YP,UP,VP)
      PSI=DENS*Q*SIN(TAU-THETA)/SIN(TAU)
      PRHO=FACTR*PRES
      PRPO=PRHO/CONV
      PSIR(I)=PRPO
      WRITE (6,104)
      WRITE (6,106)
      WRITE (6,108) XP,YP,Q,THETA,UP,VP,AM,PRHO,PRPO
      IF (NDIM .GT. 2) GO TO 18
      STFR=PSI*YP
      GO TO 20
   18 STFR=PSI*YP**2/2.0
   20 SUM=0.0
      DO 36 I=1,ND
      TAU=TAU+DELT
      IF (NDIM .GT. 2) GO TO 22
      UQ=UP
      VQ=VP
      GO TO 24
   22 CALL CFF(YC/TAN(TAU),YC)
      UQ=UA
      VQ=VA
   24 CALL CFPR(YC/TAN(TAU),YC,UQ,VQ)
      PSI=DENS*Q*SIN(TAU-THETA)/SIN(TAU)
      IF (NDIM .GT. 2) GO TO 26
      YQ=STFR/PSI
      GO TO 28
   26 YQ=SQRT(2.0*STFR/PSI)
   28 XQ=YQ/TAN(TAU)
      CALL CFPR(XQ,YQ,UQ,VQ)
      PRHO=FACTR*PRES
      PRPO=PRHO/CONV
      PSIR(I+1)=PRPO
      WRITE (6,108) XQ,YQ,Q,THETA,UQ,VQ,AM,PRHO,PRPO
   30 PAVE=(PSIR(I+1)+PSIR(I)-2.0)
      IF (NDIM .GT. 2) GO TO 32
      REF=YP-YQ
      GO TO 34
   32 REF=YP**2-YQ**2
   34 PART=CONST*PAVE*REF
      SUM=SUM+PART
      YP=YQ
   36 CONTINUE
      IF (NDIM .GT. 2) GO TO 38
      CAPT=YQ
      GO TO 40
   38 CAPT=YQ**2
   40 WRITE (6,102) THETAL,CAPT,SUM
      WRITE (6,110)
      DELS=TAN(THETAL)
   42 RETURN
      END
```

```
$IBFTC DATUR    LIST,REF,DECK,DEBUG
C
C     STARTING SUBROUTINE FOR OFF DESIGN CALCULATIONS
C
      SUBROUTINE DATUM(START,DELQ)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C.D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//39X,46HINITIAL NET SPACING PARAMETER (START) TO SMALL//)
      K=1
      I=1
      XR=BODY(2)
      YR=S(1,1)+S(1,2)*XR+S(1,3)*XR**2+S(1,4)*XR**3
      CALL EQUK(K,I,XR,YR,UC,VC)
      SLOPE=A
      UCR=UC
      VCR=VC
      U1=UC
      V1=VC
      DEL1=-1.0/DELC
      DELU=DELU1
      REF=START/2.0
      KOUT=0
      DO 18 K=1,49
      IF (KOUT .EQ. 1) GO TO 20
      IF (K .LT. 49) GO TO 10
      WRITE (6,100)
      CALL EXIT
   10 DIST=START
      IF (K .LE. 2) DIST=REF
      ANGLE=ATAN(SLOPE)
      XR=X(K,I)+DIST*COS(ANGLE)
      YR=Y(K,I)+DIST*SIN(ANGLE)
      TANT=YR/XR
      IF (TANT .EQ. DELQ) KOUT=1
      IF (TANT .LE. DELQ) GO TO 12
      XR=(Y(K,I)-SLOPE*X(K,I))/(DELQ-SLOPE)
      YR=DELQ*XR
      KOUT=1
   12 IF (NDIM .GT. 2) GO TO 14
      UR=UCR
      VR=VCR
      GO TO 16
   14 CALL CFF(XR,YR)
      UR=UA
      VR=VA
   16 CALL EQUK(K+1,1,XR,YR,UR,VR)
      SLOPE=A
   18 CONTINUE
   20 RETURN
      END
```

```
$IBFTC SPIKEI  LIST,REF,DECK,DEBUG
C
C     CHARACTERISTIC ROUTINE FOR ISENTROPIC RAMP, NDIM=2
C     CHARACTERISTIC ROUTINE FOR ISENTROPIC SPIKE, NDIM=3
C
      SUBROUTINE SPIKE(KA,IA,M,START,AMF,SIGN,PRINT)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//43X,30HCONDITIONS ON THE RAMP CONTOUR//)
  102 FORMAT (//43X,31HCONDITIONS ON THE SPIKE CONTOUR//)
  104 FORMAT (//45X,28HCONDITIONS IN THE FLOW FIELD//)
  106 FORMAT (//8X,1HX,13X,1HY,13X,1HQ,11X,5HTHETA,11X,1HU,13X,1HV,9X,
     18HMACH NO.,8X,4HP/HO,10X,4HP/PO//)
  108 FORMAT (1P9E14.5//)
  110 FORMAT (1H1)
  112 FORMAT (1HJ)
  114 FORMAT (//32X,55HCALCULATION OF FLOW FIELD PERFORMED IN SUBROUTINE
     1 SPIKE//)
  116 FORMAT (//35X,46HINITIAL NET SPACING PARAMETER (START) TO SMALL//)
  118 FORMAT (//28X,52HPRESSURE RECOVERY (H/HO) ALONG THE CONTOUR SURFAC
     1E =1PE12.5//)
      K=KA
      I=IA
      CONVR=(1.0+(GAM-1.0)/2.0*AMO**2)**(-GAM/(GAM-1.0))
      IREV=IREV+1
      RECV(1,IREV)=RECV(1,IREV-1)
      RECV(2,IREV)=RECV(2,IREV-1)
      IF (NDIM .GT. 2) GO TO 10
      CALL PSA(AMO,0.0,THETC)
      U(K,I)=UK
      V(K,I)=VK
      UCR=UK
      VCR=VK
      GO TO 12
   10 CALL CFF(X(K,I),Y(K,I))
      U(K,I)=UA
      V(K,I)=VA
   12 CALL CFPR(X(K,I),Y(K,I),U(K,I),V(K,I))
      AMP=AM
      ANGLE=THETA
      PSIR(I)=PSI
      SLOPE=A
      UREF=UC
      VREF=VC
      DREF=-1.0/DELC
      DELU=-DELU1
      KOUT=0
      STFR=1.0
      DO 22 K=1,49
      IF (KOUT .EQ. 1) GO TO 24
      IF (K .LT. 49) GO TO 14
      WRITE (6,116)
      CALL CFPR(0.0,0.0,0.0,0.0)
   14 ALPHA=ATAN(SLOPE)
      XR=X(K,I)-START*COS(ALPHA)
```

```
      YR=Y(K,I)-START*SIN(ALPHA)
      TANT=YR/XR
      IF (TANT .EQ. DELC) KOUT=1
      IF (TANT .GT. DELC) GO TO 16
      XR=(Y(K,I)-SLOPE*X(K,I))/(DELC-SLOPE)
      YR=DELC*XR
      KOUT=1
   16 IF (NDIM .GT. 2) GO TO 18
      UR=UCR
      VR=VCR
      GO TO 20
   18 U1=UREF
      V1=VREF
      DEL1=DREF
      CALL CFF(XR,YR)
      UR=UA
      VR=VA
      UREF=U1
      VREF=V1
      DREF=DEL1
   20 CALL EQUK(K+1,I,XR,YR,UR,VR)
      SLOPE=A
      PSIAV=.5*(PSI+PSIR(I))
      EXP=NDIM-1
      REF=Y(K+1,I)**EXP-Y(K,I)**EXP
      STFR=STFR+PSIAV*REF
      PSIR(I)=PSI
   22 CONTINUE
   24 KREF=K
      IREF=1
      CALL EQUI(1,X(KREF,1),Y(KREF,1),U(KREF,1),V(KREF,1))
      XM= M
      DELAM= (AMP-AMF)/XM
      AMF= AMP
      MR=M+1
      DO 26 I=2,MR
      AMF= AMF-DELAM
      CALL PMER(AMP,ANGLE,AMF)
      CALL EQUK(1,I,X(1,1),Y(1,1),UK,VK)
      CALL CFPR(X(1,I),Y(1,I),U(1,I),V(1,I))
      PSIR(I)=PSI
   26 CONTINUE
      DO 38 I=1,M
      STF=1.0
      DO 36 K=1,KREF
      CALL CCRC(X(K,I+1),Y(K,I+1),U(K,I+1),V(K,I+1),X(K+1,I),Y(K+1,I),
     1U(K+1,I),V(K+1,I),SIGN)
      CALL EQUK(K+1,I+1,XK,YK,UK,VK)
      KREF=K+1
      IREF=I+1
      CALL CFPR(X(K+1,I+1),Y(K+1,I+1),U(K+1,I+1),V(K+1,I+1))
      PSIAV=.5*(PSIR(I+1)+PSI)
      IF (NDIM .GT. 2) GO TO 28
      REF=Y(K+1,I+1)-Y(K,I+1)
      GO TO 30
   28 REF=Y(K+1,I+1)**2-Y(K,I+1)**2
   30 STF=STF+PSIAV*REF
      PSIR(I+1)=PSI
      IF (STF-STFR) 34,32,36
   32 CALL EQUI(I+1,X(K+1,I+1),Y(K+1,I+1),U(K+1,I+1),V(K+1,I+1))
```

```
   GO TO 38
34 CALL CCSR(STFR,STF,X(K+1,I+1),Y(K+1,I+1),U(K+1,I+1),V(K+1,I+1),
  1X(K,I+1),Y(K,I+1),U(K,I+1),V(K,I+1))
   CALL EQUI(I+1,XK.YK,UK,VK)
   GO TO 38
36 CONTINUE
38 CONTINUE
40 CALL EQUK(KREF,IREF,XB(IREF),YB(IREF),UB(IREF),VB(IREF))
   WRITE (6,114)
   IF (NDIM .GT. 2) GO TO 42
   WRITE (6,100)
   WRITE (6,118) RECV(1,IREV)
   WRITE (6,106)
   GO TO 44
42 WRITE (6,102)
   WRITE (6,118) RECV(1,IREV)
   WRITE (6,106)
44 DO 46 I=1,IREF
   IF (XB(I) .EQ. 0.0) GO TO 46
   CALL CFPR(XB(I),YB(I),UB(I),VB(I))
   PRHO=FACTR*PRES
   PRPO=PRHO/CONVR
   WRITE (6,108) XB(I),YB(I),Q,THETA,UB(I),VB(I),AM,PRHO,PRPO
46 CONTINUE
   IF (PRINT .EQ. 0.0) GO TO 50
   WRITE (6,110)
   WRITE (6,104)
   WRITE (6,106)
   DO 48 K=1,50
   WRITE (6,112)
   DO 48 I=1,IREF
   IF (X(K,I) .EQ. 0.0) GO TO 48
   CALL CFPR(X(K,I),Y(K,I),U(K,I),V(K,I))
   PRHO=FACTR*PRES
   PRPO=PRHO/CONVR
   WRITE (6,108) X(K,I),Y(K,I),Q,THETA,U(K,I),V(K,I),AM,PRHO,PRPO
48 CONTINUE
50 KR=KREF+1
   IR=IREF-1
   DO 54 K=1,50
   DO 54 I=1,50
   IF (K .GT. KR) GO TO 52
   IF (I .GT. IR) GO TO 54
52 CALL EQUK(K,I,0.0,0.0,0.0,0.0)
   CALL EQUI(I,0.0,0.0,0.0,0.0)
54 CONTINUE
   KR=KREF+1
   DO 56 K=KR,50
   CALL EQUK(K,IREF,0.0,0.0,0.0,0.0)
56 CONTINUE
   DO 58 K=1,KREF
   KR=(KREF+1)-K
   CALL EQUK(K,1,X(KR,IREF),Y(KR,IREF),U(KR,IREF),V(KR,IREF))
   CALL EQUK(KR,IREF,0.0,0.0,0.0,0.0'
58 CONTINUE
   RETURN
   END
```

```
$IBFTC CONER    LIST,REF,DECK,DEBUG
C
C     SUBROUTINE FOR CALCULATING THE C-NET BEHIND A SHOCK
C
      SUBROUTINE CONE(KA,IA,START,ALPHA,SIGN,PRINT)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1      Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2      XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3      X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4      U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5      XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6      REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//38X,47HCONDITIONS IN THE VICINITY OF THE OBLIQUE SHOCK//
     1)
  102 FORMAT (//18X,8HTAN(Y/X),6X,8HMACH NO.,8X,5HTHETA,9X,4HBETA,10X,
     13HDEL,10X,5HP2/P1,10X,5HH2/H1//)
  104 FORMAT (14X,1P7E14.5//)
  106 FORMAT (//43X,30HCONDITIONS ON THE RAMP CONTOUR//)
  108 FORMAT (//43X,31HCONDITIONS ON THE SPIKE CONTOUR//)
  110 FORMAT (//45X,28HCONDITIONS IN THE FLOW FIELD//)
  112 FORMAT (//8X,1HX,13X,1HY,13X,1HQ,11X,5HTHETA,11X,1HU,13X,1HV,9X,
     18HMACH NO.,8X,4HP/H0,10X,4HP/P0//)
  114 FORMAT (1P9E14.5//)
  116 FORMAT (1H1)
  118 FORMAT (1HJ)
  120 FORMAT (//35X,54HCALCULATION OF FLOW FIELD PERFORMED IN SUBROUTINE
     1 CONE//)
  122 FORMAT (//28X,52HPRESSURE RECOVERY (H/H0) ALONG THE CONTOUR SURFAC
     1E =1PE12.5//)
  124 FORMAT (//39X,46HINITIAL NET SPACING PARAMETER (START) TO SMALL//)
      K=KA
      I=IA
      KSIGN=SIGN
      CONVR=(1.0+(GAM-1.0)/2.0*AM0**2)**(-GAM/(GAM-1.0))
      SUMR=0.0
      SUMY=0.0
      DELTA=ALPHA
      FACTP=FACTR
      CALL CFPR(1.0,DELC,UC,VC)
      AMR=AM
      THETAR=THETA
      DELP=TAN(THETAR+ALPHA)
      NREF=NS
      IF (NS .GT. 1) GO TO 10
      X(K,I)=1.0
      Y(K,I)=DELC
      CALL CURVE(0.0,0.0,DELC,X(K,I),0.0,0.0,1.0,NS)
      NS=NS+1
      XP=X(K,I)+6.0
      CALL CURVE(X(K,I),Y(K,I),DELP,XP,0.0,0.0,1.0,NS)
      NS=NS+1
   10 XS=BODY(2)
      YS=S(2,1)+S(2,2)*XS+S(2,3)*XS**2+S(2,4)*XS**3
      CALL PSA(AMR,THETC,DELTA)
      CALL EQUK(K,I,XS,YS,UK,VK)
      CALL EQUI(I,XS,YS,UK,VK)
      BETA(K)=BETAK
      SLOPE=TAN(BETA(K))
      BETAR=BETA(K)-THETAR
      CALL OSWR(AMR,BETAR)
```

```
      KREV=(3+KSIGN)/2
      IREV=IREV+1
      RECV(KREV,IREV)=RECV(KREV,IREV-1)*RECOV
      RAVE=RECOV
      RECV1=RECOV
      FACTR=FACTP*RAVE
      WRITE (6,120)
      WRITE (6,122) RECV(KREV,IREV-1)
      WRITE (6,100)
      WRITE (6,102)
      WRITE (6,104) DELC,AMR,THETAR,BETAR,DELTA,RPRES,RECOV
      UREF=UC
      VREF=VC
      DREF=-1.0/DELC
      DELU=DELU1
      UCR=UC
      VCR=VC
      KREV=(3-KSIGN)/2
      REF=START/2.0
      KOUT=0
      EXP=NDIM-1
      DO 32 K=1,49
      IF (KOUT .EQ. 1) GO TO 33
      IF (K .LT. 49) GO TO 12
      WRITE (6,124)
      CALL EXIT
   12 DIST=START
      IF (K .LE. 2) DIST=REF
      ANGLE=ATAN(SLOPE)
      XR=X(K,1)+DIST*COS(ANGLE)
      YR=Y(K,1)+DIST*SIN(ANGLE)
      TANT=YR/XR
      IF (TANT .EQ. DELS) KOUT=1
      IF (TANT .LE. DELS) GO TO 14
      XR=(Y(K,1)-SLOPE*X(K,1))/(DELS-SLOPE)
      YR=DELS*XR
      TANT=YR/XR
      KOUT=1
   14 IF (NDIM .GT. 2) GO TO 16
      UR=UCR
      VR=VCR
      GO TO 18
   16 U1=UREF
      V1=VREF
      DEL1=DREF
      DELU=DELU1
      CALL CFF(XR,YR)
      UR=UA
      VR=VA
      UREF=U1
      VREF=V1
      DREF=DEL1
   18 CALL CFPR(XR,YR,UR,VR)
      AMR=AM
      THETAR=THETA
   20 IF (K .GT. 1) GO TO 22
      CALL CCRA(K,1,XR,YR,UR,VR,DELTA,SIGN)
      CALL EQUK(K+1,1,XK,YK,UK,VK)
      CALL EQUK(K+1,2,XJ,YJ,UJ,VJ)
      CALL EQUI(2,XJ,YJ,UJ,VJ)
```

```
      BETA(K+1)=BETAK
      SLOPE=TAN(BETA(K+1))
      BETAR=BETA(K+1)-THETAR
      DELTA=ATAN(V(K+1,1)/U(K+1,1))-THETAR
      IF (ABS(DELTA) .LE. ERROR) DELTA=0.0
      CALL OSWR(AMR,BETAR)
      RECV(KREV,IREV)=RECV(KREV,IREV-1)*RECOV
      RBAR=(RECOV+RECV1)/2.0
      YBAR=Y(K+1,1)**EXP-Y(K,1)**EXP
      SUMR=SUMR+RBAR*YBAR
      SUMY=SUMY+YBAR
      RAVE=SUMR/SUMY
      RECV1=RECOV
      FACTR=FACTP*RAVE
      WRITE (6,104) TANT,AMR,THETAR,BETAR,DELTA,RPRES,RECOV
      CALL CORE(K+1,2,SIGN)
      GO TO 32
   22 DO 24 J=2,50
      IF (X(K,J) .EQ. 0.0) GO TO 24
      CALL CCRB(K,J,XR,YR,UR,VR,DELTA,SIGN)
      TEST=(XJ-XK)/ABS(XJ-XK)
      IF (TEST .GT. 0.0) GO TO 26
   24 CONTINUE
   26 CALL EQUK(K+1,1,XK,YK,UK,VK)
      CALL EQUK(K+1,J,XJ,YJ,UJ,VJ)
      BETA(K+1)=BETAK
      SLOPE=TAN(BETA(K+1))
      BETAR=BETA(K+1)-THETAR
      DELTA=ATAN(V(K+1,1)/U(K+1,1))-THETAR
      IF (ABS(DELTA) .LE. ERROR) DELTA=0.0
      CALL OSWR(AMR,BETAR)
      RECV(KREV,IREV)=RECV(KREV,IREV-1)*RECOV
      RBAR=(RECOV+RECV1)/2.0
      YBAR=Y(K+1,1)**EXP-Y(K,1)**EXP
      SUMR=SUMR+RBAR*YBAR
      SUMY=SUMY+YBAR
      RAVE=SUMR/SUMY
      RECV1=RECOV
      FACTR=FACTP*RAVE
      WRITE (6,104) TANT,AMR,THETAR,BETAR,DELTA,RPRES,RECOV
      DO 28 I=J,50
      IF (X(K,I+1) .EQ. 0.0) GO TO 30
      CALL CCRC(X(K,I+1),Y(K,I+1),U(K,I+1),V(K,I+1),X(K+1,I),Y(K+1,I),
     1          U(K+1,I),V(K+1,I),-SIGN)
      CALL EQUK(K+1,I+1,XK,YK,UK,VK)
   28 CONTINUE
   30 CALL CCRE(X(K+1,I),Y(K+1,I),U(K+1,I),V(K+1,I),SIGN)
      CALL EQUK(K+1,I+1,XK,YK,UK,VK)
      CALL EQUI(I+1,XK,YK,UK,VK)
      CALL CORE(K+1,I+1,SIGN)
   32 CONTINUE
   33 KREF=K
      IREF=KREF
      IF (NREF .GT. 1) GO TO 38
      NS=NREF
      RATIO=1.0/Y(KREF,1)
      DO 34 I=1,50
      CALL EQUI(I,RATIO*XB(I),RATIO*YB(I),UB(I),VB(I))
      DO 34 K=1,50
      IF (X(K,I) .EQ. 0.0) GO TO 34
```

```
      CALL EQUK(K,I,RATIO*X(K,I),RATIO*Y(K,I),U(K,I),V(K,I))
   34 CONTINUE
      CALL CURVE(0.0,0.0,DELC,X(1,1),0.0,0.0,1.0,NS)
      NS=NS+1
      XP=X(KREF,IREF)+3.0
   36 CALL CURVE(X(1,1),Y(1,1),V(1,1)/U(1,1),XP,0.0,0.0,1.0,NS)
      NS=NS+1
   38 WRITE (6,116)
   40 IF (NDIM .GT. 2) GO TO 42
      WRITE (6,106)
      WRITE (6,112)
      GO TO 44
   42 WRITE (6,108)
      KREV=(3+KSIGN)/2
      WRITE (6,122) RECV(KREV,IREV)
      WRITE (6,112)
   44 DO 48 I=1,50
      IF (XB(I) .EQ. 0.0) GO TO 48
   46 CALL CFPR(XB(I),YB(I),UB(I),VB(I))
      PRHO=FACTR*PRES
      PRPO=PRHO/CONVR
      WRITE (6,114) XB(I),YB(I),Q,THETA,UB(I),VB(I),AM,PRHO,PRPO
   48 CONTINUE
      IF (PRINT .EQ. 0.0) GO TO 54
      WRITE (6,116)
      WRITE (6,110)
      WRITE (6,112)
   50 DO 52 K=1,50
      WRITE (6,118)
      DO 52 I=1,50
      IF (X(K,I) .EQ. 0.0) GO TO 52
      IF (I .GT. K) GO TO 52
      CALL CFPR(X(K,I),Y(K,I),U(K,I),V(K,I))
      PRHO=FACTR*PRES
      PRPO=PRHO/CONVR
      WRITE (6,114) X(K,I),Y(K,I),Q,THETA,U(K,I),V(K,I),AM,PRHO,PRPO
   52 CONTINUE
   54 RETURN
      END
```

```
$IBFTC PUTS    LIST,REF,DECK,DEBUG
C
C     SUBROUTINE FOR LOCATING THE COWL CONDITIONS
C
      SUBROUTINE PUT(XCOWL,YCOWL)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//25X,57HCOWL LIP POINT LIES OUTSIDE REGION OF FLOW FIELD,
     1 XCOWL =1PE12.5,2X,7HYCOWL =1PE12.5//)
      DO 10 K=1,50
      IF (X(K,1) .EQ. 0.0) GO TO 12
   10 CONTINUE
   12 KP=K-1
      DIST=Y(KP,1)-YCOWL
      IF (DIST .GE. 0.0) GO TO 14
      WRITE (6,100) XCOWL,YCOWL
      CALL CFPR(0.0,0.0,0.0,0.0)
   14 DO 16 I=1,KP
      IF (X(KP,I) .EQ. 0.0) GO TO 16
      DIST=X(KP,I)-XCOWL
      IF (DIST .GE. 0.0) GO TO 18
   16 CONTINUE
      WRITE (6,100) XCOWL,YCOWL
      CALL CFPR(0.0,0.0,0.0,0.0)
   18 RMIN=10.0
      DO 20 K=1,50
      DO 20 I=1,50
      IF (X(K,I) .EQ. 0.0) GO TO 20
      DELX=X(K,I)-XCOWL
      DELY=Y(K,I)-YCOWL
      RADIUS=SQRT(DELX**2+DELY**2)
      RMIN=AMIN1(RADIUS,RMIN)
   20 CONTINUE
      DO 22 K=1,50
      DO 22 I=1,50
      IF (X(K,I) .EQ. 0.0) GO TO 22
      DELX=X(K,I)-XCOWL
      DELY=Y(K,I)-YCOWL
      RADIUS=SQRT(DELX**2+DELY**2)
      IF (RADIUS .EQ. RMIN) GO TO 24
   22 CONTINUE
   24 IF (ABS(RADIUS) .GT. ERROR) GO TO 26
      KREF=KP
      IREF=IP
      GO TO 38
   26 KP=K
      IP=I
      KREF=K+1
      CALL CFPR(X(KP,IP),Y(KP,IP),U(KP,IP),V(KP,IP))
      TANA=A
      TANB=B
      DO 28 J=1,2
      KSIGN=(-1)**(J+1)
      SIGN=(-1)**(J+1)
      KREP=KP+KSIGN
```

```
      IF (X(KREP,IP) .EQ. 0.0) GO TO 28
      CALL BOUND(KP,IP,KREP,IP,XCOWL,YCOWL,TANB,1.0)
      TEST=(XK-X(KP,IP))/ABS(XK-X(KP,IP))
      IF (TEST .EQ. SIGN) GO TO 30
   28 CONTINUE
   30 XI=XK
      YI=YK
      UI=UK
      VI=VK
      DO 34 J=1,2
      IREP=IP
      ISIGN=(-1)**(J+1)
      IF (IP .EQ. 1) ISIGN=1
      SIGN=ISIGN
   31 IREP=IREP+ISIGN
      IF (X(KP,IREP) .GT. 0.0) GO TO 32
      GO TO 31
   32 CALL BOUND(KP,IP,KP,IREP,XCOWL,YCOWL,TANA,-1.0)
      TEST=(XK-X(KP,IP))/ABS(XK-X(KP,IP))
      IF (TEST .EQ. SIGN) GO TO 36
   34 CONTINUE
   36 CALL CCRC(XK,YK,UK,VK,XI,YI,UI,VI,1.0)
      KREF=KP
      IREF=IP
      CALL EQUK(KREF,IREF,XCOWL,YCOWL,UK,VK)
   38 RETURN
      END
```

```
$IBFTC SHAPS    LIST,REF,DECK,DEBUG
C
C     CHARACTERISTIC FIELD FOR THE BOUNDARY Y=A0+A1*Z+A2*Z**2+A3*Z**3
C
      SUBROUTINE SHAPE(KA,IA,SIGN,SET,PRINT)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1          O,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2          XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3          X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4          U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5          XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6          REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//30X,30HCOALESCENCE HAS OCCURED AT X =1PE12.5,2X,3HY =1PE
     112.5//)
  102 FORMAT (//43X,33HCONDITIONS ON THE CONTOUR SURFACE//)
  104 FORMAT (//45X,28HCONDITIONS IN THE FLOW FIELD//)
  106 FORMAT (//8X,1HX,13X,1HY,13X,1HQ,11X,5HTHETA,11X,1HU,13X,1HV,9X,
     18HMACH NO.,8X,4HP/H0,10X,4HP/P0//)
  108 FORMAT (1P9E14.5//)
  110 FORMAT (1H1)
  112 FORMAT (1HJ)
  114 FORMAT (//33X,55HCALCULATION OF FLOW FIELD PERFORMED IN SUBROUTINE
     1 SHAPE//)
```

```
  116 FORMAT (//28X,52HPRESSURE RECOVERY (H/HO) ALONG THE CONTOUR SURFAC
     1E =1PE12.5//)
      WRITE (6,110)
      WRITE (6,114)
      CONVR=(1.0+(GAM-1.0)/2.0*AMO**2)**(-GAM/(GAM-1.0))
      KSIGN=SIGN
      IREV=IREV+1
      RECV(1,IREV)=RECV(1,IREV-1)
      RECV(2,IREV)=RECV(2,IREV-1)
   12 K=KA
      I=IA
      CALL EQUI(I,X(K,I),Y(K,I),U(K,I),V(K,I))
      KREF=KA
      IR=IA+1
      DO 30 I=IR,40
      KREF=KREF+1
      K=KREF
      LP=I-1
      DO 14 L=1,LP
      J=I-L
      IF (X(K,J) .GT. 0.0) GO TO 16
   14 CONTINUE
      GO TO 32
   16 CALL CCRE(X(K,J),Y(K,J),U(K,J),V(K,J),SIGN)
      CALL EQUK(K,K,XK,YK,UK,VK)
      CALL EQUI(K,XK,YK,UK,VK)
      IREF=I
      DO 28 K=KREF,50
      JR=I-IA
      DO 18 J=1,JR
      IPT=I-J
      IF (X(K+1,IPT) .GT. 0.0) GO TO 20
   18 CONTINUE
      GO TO 30
   20 TEST=X(K+1,IPT)-X(K,IPT)
      IF (ABS(TEST) .GT. 0.0) GO TO 22
      CALL EQUK(K+1,I,X(K,I),Y(K,I),U(K,I),V(K,I))
      GO TO 24
   22 CALL CCRC(X(K+1,IPT),Y(K+1,IPT),U(K+1,IPT),V(K+1,IPT),
     1          X(K,I),Y(K,I),U(K,I),V(K,I),SIGN)
      CALL EQUK(K+1,I,XK,YK,UK,VK)
      RADII=X(K+1,I)-X(K,I)
      IF (RADII .GT. 0.0) GO TO 24
      WRITE (6,100) X(K+1,I),Y(K+1,I)
      IF (K .EQ. I) GO TO 30
      CALL EQUK(K,I,X(K+1,I),Y(K+1,I),U(K+1,I),V(K+1,I))
   24 RADIUS=X(K+1,I)-X(K+1,IPT)
      IF (RADIUS .GT. 0.0) GO TO 28
      WRITE (6,100) X(K+1,I),Y(K+1,I)
      IF (IPT .EQ. IA) GO TO 30
      KF=K+1
      DO 26 JPT=KF,50
      CALL EQUK(JPT,IPT,0.0,0.0,0.0,0.0)
   26 CONTINUE
   28 CONTINUE
```

```
30 CONTINUE
32 WRITE (6,102)
   KREV=(3+KSIGN)/2
   WRITE (6,116) RECV(KREV,IREV)
   WRITE (6,106)
   DO 34 I=1,50
   IF (XB(I) .EQ. 0.0) GO TO 34
   CALL CFPR(XB(I),YB(I),UB(I),VB(I))
   PRHO=FACTR*PRES
   PRPO=PRHO/CONVR
   WRITE (6,108) XB(I),YB(I),Q,THETA,UB(I),VB(I),AM,PRHO,PRPO
34 CONTINUE
   IF (PRINT .EQ. 0.0) GO TO 37
   WRITE (6,110)
   WRITE (6,104)
   WRITE (6,106)
   DO 36 K=1,KREF
   WRITE (6,112)
   DO 36 I=1,50
   IF (X(K,I) .EQ. 0.0) GO TO 36
   IF (I .GT. K) GO TO 36
   CALL CFPR(X(K,I),Y(K,I),U(K,I),V(K,I))
   PRHO=FACTR*PRES
   PRPO=PRHO/CONVR
   WRITE (6,108) X(K,I),Y(K,I),Q,THETA,U(K,I),V(K,I),AM,PRHO,PRPO
36 CONTINUE
37 DO 38 K=KA,50
   IF (X(K,IA) .EQ. 0.0) GO TO 40
38 CONTINUE
40 KREF=K-1
   IF (SET .EQ. 0.0) GO TO 56
   KP=KREF
   IP=1
   CALL MATRX
   CALL ENDS(KP,1,-SIGN)
   KREF=KP
   IREF=IP
   CALL EQUK(KREF,KREF,XB(KREF),YB(KREF),UB(KREF),VB(KREF))
   K=0
   DO 46 J=1,50
   IF (X(J,KREF) .EQ. 0.0) GO TO 46
   K=K+1
   CALL EQUK(K,1,X(J,KREF),Y(J,KREF),U(J,KREF),V(J,KREF))
46 CONTINUE
48 KREF=K
   DO 50 K=1,50
   DO 50 I=2,50
   CALL EQUK(K,I,0.0,0.0,0.0,0.0)
50 CONTINUE
   CALL EQUI(1,X(1,1),Y(1,1),U(1,1),V(1,1))
   DO 52 I=2,50
   CALL EQUI(I,0.0,0.0,0.0,0.0)
52 CONTINUE
56 RETURN
   END
```

```
$IBFTC SKOCKS  LIST,REF,DECK,DEBUG
C
C     SUBROUTINE FOR CALCULATING THE C-NET BEHIND A SHOCK
C
C     FOR INTERNAL ISEN. COMP. ON DESIGN SETP=0.0,SETQ=0.0
C     FOR INTERNAL ISEN. COMP. OFF DESIGN SETP=1.0,SETQ=1.0
C     FOR INTERNAL REFLCT. SHOCK ON DESIGN SETP=1.0,SETQ=0.0
C     FOR INTERNAL REFLCT. SHOCK OFF DESIGN SETP=1.0,SETQ=0.0
C
      SUBROUTINE SHOCK(KA,IA,DELTA,SIGN,SETP,SETQ)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1      Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2      XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3      X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4      U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5      XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6      REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//38X,47HCONDITIONS IN THE VICINITY OF THE OBLIQUE SHOCK//
     1)
  102 FORMAT (//15X,1HX,12X,1HY,10X,8HMACH NO.,8X,5HTHETA,9X,4HBETA,10X,
     13HDEL,10X,5HP2/P1,10X,5HH2/H1//)
  104 FORMAT (7X,1P8E14.5//)
  106 FORMAT (//8X,1HX,13X,1HY,13X,1HQ,11X,5HTHETA,11X,1HU,13X,1HV,9X,
     18HMACH NO.,8X,4HP/HO,10X,4HP/PO//)
  108 FORMAT (1P9E14.5//)
  110 FORMAT (//43X,33HCONDITIONS ON THE CONTOUR SURFACE//)
  112 FORMAT (//45X,28HCONDITIONS IN THE FLOW FIELD//)
  114 FORMAT (1H1)
  116 FORMAT (1HJ)
  118 FORMAT (//34X,55HCALCULATION OF FLOW FIELD PERFORMED IN SUBROUTINE
     1 SHOCK//)
  120 FORMAT (//28X,52HPRESSURE RECOVERY (H/HO) ALONG THE CONTOUR SURFAC
     1E =1PE12.5//)
      WRITE (6,114)
      WRITE (6,118)
      WRITE (6,100)
      WRITE (6,102)
      CONVR=(1.0+(GAM-1.0)/2.0*AMO**2)**(-GAM/(GAM-1.0))
      KSIGN=SIGN
      SUMR=0.0
      SUMY=0.0
      DEL=DELTA
      FACTP=FACTR
      CALL CFPR(X(IA,KA),Y(IA,KA),U(IA,KA),V(IA,KA))
      CALL PSA(AM,THETA,DEL)
      CALL EQUK(1,1,X(IA,KA),Y(IA,KA),UK,VK)
      CALL EQUI(1,X(IA,KA),Y(IA,KA),UK,VK)
      CALL CFPR(X(IA,KA),Y(IA,KA),U(IA,KA),V(IA,KA))
      AMR=AM
      BETA(1)=-SIGN*BETAK+(1.0+SIGN)*THETA
      SLOPE=TAN(BETA(1))
      BETAR=BETA(1)-THETA
      KREF=KA
      IREF=IA
      CALL OSWR(AMR,BETAR)
      KREV=(3+KSIGN)/2
      IREV=IREV+1
      RECV(KREV,IREV)=RECV(KREV,IREV-1)*RECOV
      RAVE=RECOV
      RECV1=RECOV
```

```
      FACTR=FACTP*RAVE
      WRITE (6,104) X(IREF,KREF),Y(IREF,KREF),AMR,THETA,BETAR,DEL,RPRES,
     1RECOV
      KREV=(3-KSIGN)/2
      EXP=NDIM-1
      DO 30 K=2,50
      IF (KREF .EQ. 50) GO TO 32
      IF (ABS(DELTA) .GT. 0.001) GO TO 10
      IREF=IREF+1
      IF (X(IREF,KREF) .EQ. 0.0) GO TO 32
      GO TO 14
   10 DO 12 L=1,2
      REG=-REG
      CALL SEEK(KREF,IREF,X(IREF,KREF),Y(IREF,KREF),SLOPE,SIGN)
      CALL EQUK(IREF,KREF,XK,YK,UK,VK)
      REG=-REG
      IF (K .EQ. 2) GO TO 12
      IF (KREF .EQ. 50) GO TO 14
      DELX=X(K-1,1)-X(IREF,KREF)
      DELY=Y(K-1,1)-Y(IREF,KREF)
      DIST=SQRT(DELX**2+DELY**2)
      IF (ABS(DIST) .GT. 0.000) GO TO 14
   12 CONTINUE
   14 IF (K .GT. 2) GO TO 20
   16 CALL CCRA(K-1,1,X(IREF,KREF),Y(IREF,KREF),U(IREF,KREF),
     1           V(IREF,KREF),DEL,SIGN)
      CALL EQUK(K,1,XK,YK,UK,VK)
      CALL EQUK(K,2,XJ,YJ,UJ,VJ)
      CALL EQUI(2,XJ,YJ,UJ,VJ)
      CALL CFPR(X(IREF,KREF),Y(IREF,KREF),U(IREF,KREF),V(IREF,KREF))
      AMR=AM
      BETA(K)=-SIGN*BETAK+(1.0+SIGN)*THETA
      SLOPE=TAN(BETA(K))
      BETAR=BETA(K)-THETA
      DEL=ATAN(V(K,1)/U(K,1))-THETA
      IF (ABS(DEL) .LE. ERROR) DEL=0.0
      CALL OSWR(AMR,BETAR)
      RECV(KREV,IREV)=RECV(KREV,IREV-1)*RECOV
      RBAR=(RECOV+RECV1)/2.0
      YBAR=ABS(Y(K,1)**EXP-Y(K-1,1)**EXP)
      SUMR=SUMR+RBAR*YBAR
      SUMY=SUMY+YBAR
      RAVE=SUMR/SUMY
      RECV1=RECOV
      FACTR=FACTP*RAVE
      WRITE (6,104) X(IREF,KREF),Y(IREF,KREF),AMR,THETA,BETAR,DEL,RPRES,
     1RECOV
      CALL CORE(K,2,SIGN)
      GO TO 30
   20 DO 22 J=2,50
      IF (X(K-1,J) .EQ. 0.0) GO TO 22
      CALL CCRB(K-1,J,X(IREF,KREF),Y(IREF,KREF),U(IREF,KREF),
     1           V(IREF,KREF),DEL,SIGN)
      TEST=(XJ-XK)/ABS(XJ-XK)
      IF (TEST .GT. 0.0) GO TO 24
   22 CONTINUE
   24 CALL EQUK(K,1,XK,YK,UK,VK)
      CALL EQUK(K,J,XJ,YJ,UJ,VJ)
      KP=K
      CALL CFPR(X(IREF,KREF),Y(IREF,KREF),U(IREF,KREF),V(IREF,KREF))
```

```
      AMR=AM
      BETA(K)=-SIGN*BETAK+(1.0+SIGN)*THETA
      SLOPE=TAN(BETA(K))
      BETAR=BETA(K)-THETA
      DEL=ATAN(V(K,1)/U(K,1))-THETA
      IF (ABS(DEL) .LE. ERROR) DEL=0.0
      CALL OSWR(AMR,BETAR)
      RECV(KREV,IREV)=RECV(KREV,IREV-1)*RECOV
      RBAR=(RECOV+RECV1)/2.0
      YBAR=ABS(Y(K,1)**EXP-Y(K-1,1)**EXP)
      SUMR=SUMR+RBAR*YBAR
      SUMY=SUMY+YBAR
      RAVE=SUMR/SUMY
      RECV1=RECOV
      FACTR=FACTP*RAVE
      WRITE (6,104) X(IREF,KREF),Y(IREF,KREF),AMR,THETA,BETAR,DEL,RPRES,
     1RECOV
      DO 26 I=J,50
      IF ((K-1) .EQ. I) GO TO 28
      CALL CCRC(X(K,I),Y(K,I),U(K,I),V(K,I),X(K-1,I+1),Y(K-1,I+1),
     1          U(K-1,I+1),V(K-1,I+1),SIGN)
      CALL EQUK(K,I+1,XK,YK,UK,VK)
   26 CONTINUE
   28 CALL CCRE(X(K,I),Y(K,I),U(K,I),V(K,I),SIGN)
      CALL EQUK(K,I+1,XK,YK,UK,VK)
      CALL EQUI(I+1,XK,YK,UK,VK)
      CALL CORE(K,I+1,SIGN)
      IP=I+1
   30 CONTINUE
   32 KREF=KP
      IREF=IP
      IF (SETP .GT. 0.0) GO TO 34
      CALL LOCUS(1,SIGN)
      STFR=PSI
   34 WRITE (6,114)
      WRITE (6,110)
      KREV=(3+KSIGN)/2
      WRITE (6,120) RECV(KREV,IREV)
      WRITE (6,106)
      DO 36 I=1,IREF
      IF (XB(I) .EQ. 0.0) GO TO 36
      CALL CFPR(XB(I),YB(I),UB(I),VB(I))
      PRHO=FACTR*PRES
      PRPO=PRHO/CONVR
      WRITE (6,108) XB(I),YB(I),Q,THETA,UB(I),VB(I),AM,PRHO,PR
   36 CONTINUE
      WRITE (6,114)
      WRITE (6,112)
      WRITE (6,106)
      DO 38 K=1,KREF
      WRITE (6,116)
      DO 38 I=1,50
      IF (X(K,I) .EQ. 0.0) GO TO 38
      IF (I .GT. K) GO TO 38
      CALL CFPR(X(K,I),Y(K,I),U(K,I),V(K,I))
      PRHO=FACTR*PRES
      PRPO=PRHO/CONVR
      WRITE (6,108) X(K,I),Y(K,I),Q,THETA,U(K,I),V(K,I),AM,PRHO,PRPO
   38 CONTINUE
      KF=KREF+1
```

```
      DO 46 K=KF,50
      DO 46 I=1,50
      CALL EQUK(K,I,0.0,0.0,0.0,0.0)
   46 CONTINUE
      DO 50 I=1,50
      IF (I .GT. KREF) GO TO 48
      CALL EQUI(I,X(I,I),Y(I,I),U(I,I),V(I,I))
      GO TO 50
   48 CALL EQUI(I,0.0,0.0,0.0,0.0)
   50 CONTINUE
      IF (SETP .EQ. 0.0) GO TO 62
      KP=KREF
      IP=IREF
      CALL MATRX
      CALL ENDS(KP,1,-SIGN)
      KREF=KP
      IREF=IP
      CALL EQUK(KREF,KREF,XB(KREF),YB(KREF),UB(KREF),VB(KREF))
      IF (SETQ .EQ. 0.0) GO TO 62
      K=0
      DO 54 J=1,50
      IF (X(J,KREF) .EQ. 0.0) GO TO 54
      IF (J .EQ. 1) GO TO 52
      DELX=X(J,KREF)-X(K,1)
      DELY=Y(J,KREF)-Y(K,1)
      DIST=SQRT(DELX**2+DELY**2)
      IF (ABS(DIST) .LT. 0.000) GO TO 54
   52 K=K+1
      CALL EQUK(K,1,X(J,KREF),Y(J,KREF),U(J,KREF),V(J,KREF))
   54 CONTINUE
      KF=K
      DO 58 K=1,50
      DO 58 I=2,50
   56 CALL EQUK(K,I,0.0,0.0,0.0,0.0)
   58 CONTINUE
      CALL EQUI(1,X(1,1),Y(1,1),U(1,1),V(1,1))
      DO 60 I=2,50
      CALL EQUI(I,0.0,0.0,0.0,0.0)
   60 CONTINUE
   62 PSI=STER
      DELA=DEL
      RETURN
      END
```

```
$IBFTC SEEKS    LIST,REF,DECK,DEBUG
C
C     SUBROUTINE FOR LOCATING SPECIFIC POINTS IN A C-NET
C
      SUBROUTINE SEEK(KA,IA,XR,YR,SLOPE,SIGN)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      DIMENSION NPT(2)
      K=KA
      I=IA
      IF (K .EQ. I) GO TO 30
      JF=KA-IA
      DO 10 J=1,JF
      I=IA+J
      IF (X(I,K) .GT. 0.0) GO TO 12
   10 CONTINUE
   12 IREF=I
      XF=XR
      YF=YR
      DO 24 J=1,2
      RMIN=10.0
      DO 14 K=IREF,KA
      IF (X(IREF,K) .EQ. 0.0) GO TO 14
      RADIUS=SQRT((X(IREF,K)-XF)**2+(Y(IREF,K)-YF)**2)
      RMIN=AMIN1(RADIUS,RMIN)
   14 CONTINUE
      DO 16 K=IREF,KA
      RADIUS=SQRT((X(IREF,K)-XF)**2+(Y(IREF,K)-YF)**2)
      IF (RADIUS .EQ. RMIN) GO TO 18
   16 CONTINUE
   18 KREF=K
      IF (IREF .LT. KREF) GO TO 20
      K=KREF+1
      IF (X(IREF,K) .EQ. 0.0) GO TO 30
      GO TO 22
   20 CALL BOUND(IREF,KREF,IREF,KREF-1,XR,YR,SLOPE,SIGN)
      XREF=XK
      TEST=XREF-X(IREF,KREF)
      KSIGN=TEST/ABS(TEST)
      K=KREF+KSIGN
      IF (IREF .GT. K) GO TO 30
      IF (X(IREF,K) .EQ. 0.0) GO TO 30
   22 CALL BOUND(IREF,KREF,IREF,K,XR,YR,SLOPE,SIGN)
      XF=XK
      YF=YK
      TEST=XF-X(KREF,IREF)
      KSIGN=TEST/ABS(TEST)
      K=KREF+KSIGN
      IF (IREF .GT. K) GO TO 30
   24 CONTINUE
      CALL FIND(XK)
      IF (REG .EQ. -1.0) GO TO 26
      C0=R(ICOWL,1)
      C1=R(ICOWL,2)
      C2=R(ICOWL,3)
```

```
      C3=R(ICOWL,4)
      GO TO 28
   26 CO=S(IBODY,1)
      C1=S(IBODY,2)
      C2=S(IBODY,3)
      C3=S(IBODY,4)
   28 YSURF=CO+C1*XK+C2*XK**2+C3*XK**3
      TEST=(YK-YSURF)/ABS(YK-YSURF)
      IF (TEST .EQ. SIGN) GO TO 56
   30 NPT(2)=0
      XREF=XR
      DO 54 L=1,2
      CALL FIND(XREF)
      IF (REG .EQ. -1.0) GO TO 32
      CO=R(ICOWL,1)
      C1=R(ICOWL,2)
      C2=R(ICOWL,3)
      C3=R(ICOWL,4)
      POINT=COWL(ICOWL)
      NPT(L)=ICOWL
      GO TO 34
   32 CO=S(IBODY,1)
      C1=S(IBODY,2)
      C2=S(IBODY,3)
      C3=S(IBODY,4)
      POINT=BODY(IBODY)
      NPT(L)=IBODY
   34 IF (NPT(1) .EQ. NPT(2)) GO TO 56
      IF (ABS(SLOPE) .GT. ERROR) GO TO 36
      XK=XR
      GO TO 46
   36 ALPHA=C3
      ETA=C2
      SIGA=C1
      DELTA=SIGA-SLOPE
      GAMMA=(CO-YR)+SLOPE*XR
      CALL ROOT3(ALPHA,ETA,DELTA,GAMMA)
      M=0
      DO 38 I=1,3
      J=2*I-1
      K=2*I
      P(J)=P(J)-POINT
      IF (P(J) .LE. 0.0) GO TO 38
      IF (ABS(P(K)) .GT. ERROR) GO TO 38
      M=M+1
      PSIR(M)=P(J)
```

```
38 CONTINUE
   XMIN=10.0
   DO 40 I=1,M
   XMIN=AMIN1(XMIN,PSIR(I))
40 CONTINUE
44 XK=XMIN+POINT
46 YK=C0+C1*XK+C2*XK**2+C3*XK**3
   SIGMA=C1+2.0*C2*XK+3.0*C3*XK**2
   RMIN=10.0
   DO 48 I=1,50
   IF (XB(I) .EQ. 0.0) GO TO 48
   TEST=SQRT((XB(I)-XK)**2+(YB(I)-YK)**2)
   RMIN=AMIN1(TEST,RMIN)
48 CONTINUF
   DO 50 I=1,50
   IF (XB(I) .EQ. 0.0) GO TO 50
   TEST=SQRT((XB(I)-XK)**2+(YB(I)-YK)**2)
   IF (RMIN .EQ. TEST) GO TO 52
50 CONTINUE
52 TEST=XK-XB(I)
   ISIGN=TEST/ABS(TEST)
   J=I+ISIGN
   ANGLE=ATAN(SIGMA)
   QI=SQRT(UB(I)**2+VB(I)**2)
   QJ=SQRT(UB(J)**2+VB(J)**2)
   DS=SQRT((XB(J)-XB(I))**2+(YB(J)-YB(I))**2)
   DELS=SQRT((XK-XB(I))**2+(YK-YB(I))**2)
   DQ=QJ-QI
   DQDS=DQ/DS
   QK=QI+DQDS*DELS
   UK=QK*COS(ANGLE)
   VK=QK*SIN(ANGLE)
   KREF=50
   IREF=I+1
   XREF=XK
54 CONTINUE
56 RETURN
   END
```

```
$IBFTC LOCI    LIST,REF,DECK,DEBUG
C
C     SUBROUTINE FOR LOCATING A SPECIFIC CHARACTERISTIC
C
      SUBROUTINE LOCUS(KA,SIGN)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
      DO 8 K=KA,50
      IF (X(K,1) .EQ. 0.0) GO TO 10
    8 CONTINUE
   10 KREF=K
      DO 12 I=2,50
      IF (X(KREF-1,I) .GT. 0.0) GO TO 14
   12 CONTINUE
   14 IREF=I
      KP=KREF-1
      REG=-REG
      XP=X(KP,1)+3.0
      TANS=V(KP,1)/U(KP,1)
      CALL CURVE(X(KP,1),Y(KP,1),TANS,XP,0.0,0.0,1.0,NS)
      NS=NS+1
      CALL CCRE(X(KP,IREF),Y(KP,IREF),U(KP,IREF),V(KP,IREF),-SIGN)
      CALL EQUK(KREF,IREF,XK,YK,UK,VK)
      CALL CORE(KREF,IREF,-1.0)
      REG=-REG
      BODY(NS)=X(KREF,IREF)
      CALL CFPR(X(KREF,IREF),-Y(KREF,IREF),U(KREF,IREF),-V(KREF,IREF))
      PSIR(IREF)=PSI
      STFR=1.0
      KP=KREF-1
      DO 22 KK=1,KP
      K=KREF-KK
      IF (IREF .GT. K) GO TO 24
      CALL CFPR(X(K,IREF),-Y(K,IREF),U(K,IREF),-V(K,IREF))
      PSIAV=.5*(PSIR(IREF)+PSI)
      PSIR(IREF)=PSI
      IF (NDIM .GT. 2) GO TO 18
      REF=Y(K,IREF)-Y(K+1,IREF)
      GO TO 20
   18 REF=Y(K,IREF)**2-Y(K+1,IREF)**2
   20 STFR=STFR+PSIAV*REF
   22 CONTINUE
   24 DO 28 K=1,50
      DO 28 I=1,50
      IF (I .GT. K) GO TO 26
      IF (K .GT. KREF) GO TO 26
      IF (I .LE. IREF) GO TO 28
   26 CALL EQUK(K,I,0.0,0.0,0.0,0.0)
   28 CONTINUE
      DO 30 I=1,50
      IF (I .LE. IREF) GO TO 30
      CALL EQUI(I,0.0,0.0,0.0,0.0)
   30 CONTINUE
      PSI=STFR
      RETURN
      END
```

```
$IBFTC SURFS    LIST,REF,DECK,DEBUG
C
C     CONDITIONS IN THE ISENTROPIC COMPRESSION REGION
C
      SUBROUTINE SURF(KS,IS,NN,AMT,THROAT,ANGLE,STFR,SIGN)
      COMMON AM0,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//8X,1HX,13X,1HY,13X,1HQ,11X,5HTHETA,11X,1HU,13X,1HV,9X,
     18HMACH NO.,8X,4HP/H0,10X,4HP/P0//)
  102 FORMAT (1P9E14.5//)
  104 FORMAT (//43X,33HCONDITIONS ON THE CONTOUR SURFACE//)
  106 FORMAT (//45X,28HCONDITIONS IN THE FLOW FIELD//)
  108 FORMAT (//44X,3HAMT,11X,3HTHR,11X,3HANG//)
  110 FORMAT (37X,1P3E14.5//)
  112 FORMAT (//28X,46HISENTROPIC COMPRESSION REGION INITIATES AT X =1PE
     112.5,2X,3HY =1PE12.5////)
  114 FORMAT (1H1)
  116 FORMAT (1HJ)
  118 FORMAT (//33X,54HCALCULATION OF FLOW FIELD PERFORMED IN SUBROUTINE
     1 SURF//)
  120 FORMAT (//29X,50HPRESSURE RECOVERY (H/H0) ALONG THE UPPER CONTOUR
     1=1PE12.5//)
  122 FORMAT (//29X,50HPRESSURE RECOVERY (H/H0) ALONG THE LOWER CONTOUR
     1=1PE12.5//)
      K=KS
      I=IS
      MP=K
      MPF=MP+NN
      XNN=NN
      POINT=1.0
      CONVR=(1.0+(GAM-1.0)/2.0*AM0**2)**(-GAM/(GAM-1.0))
      IREV=IREV+1
      RECV(1,IREV)=RECV(1,IREV-1)
      RECV(2,IREV)=RECV(2,IREV-1)
      CALL CFPR(X(K.I),Y(K,I),U(K,I),V(K,I))
      AMR=AM
      THETAR=THETA
      SIGI=TAN(THETAR)
      IP=I+1
      P(I)=1.0
      DELAM=(AMR-AMT)/XNN
      DELX=THROAT/XNN
      TAU=TAN(ANGLE)-TAN(THETAR)
      X0=X(KS,IS)
      A0=Y(KS,IS)
      A1=V(KS,IS)/U(KS,IS)
      IF (THROAT .EQ. 0.0) GO TO 10
      A2=TAU/(2.0*THROAT)
   10 AMF=AMR
      NNN=NN+IP-1
      DO 20 I=IP,NNN
      IF (THROAT .GT. 0.0) GO TO 12
      P(I)=1.0
      GO TO 14
   12 P(I)=1.0+P(I-1)
```

```
14 AMF=AMF-DELAM
   KP=P(I)
   KT=P(I)-P(I-1)
   K=MP-1+KP
   KK=K-KT
   KL=K
   XK=X(KK,I-1)+DELX
   IF (ABS(ANGLE) .GT. 0.0) GO TO 16
   CALL PMER(AMR,-THETAR,AMF)
   SIGK=-VK/UK
   SIGC=0.5*(SIGI+SIGK)
   SIGI=SIGK
   YK=Y(KK,I-1)+SIGC*DELX
   EXP=3.0
   GO TO 18
16 YK=A0+A1*(XK-X0)+A2*(XK-X0)**2
   SIGK=A1+2.0*A2*(XK-X0)
   PHI=ATAN(SIGK)
   QK=SQRT(AMF**2/(1.0+SIGSQ*(AMF**2-1.0)))
   UK=QK*COS(PHI)
   VK=-QK*SIN(PHI)
   EXP=2.0
18 CALL EQUK(K,I,XK,YK,UK,-VK)
   KREF=K
   IREF=I
   CALL CFPR(X(K,I),-Y(K,I),U(K,I),-V(K,I))
   PSIR(I)=PSI
20 CONTINUE
   DO 36 I=IP,NNN
   STF=1.0
   DO 34 KK=1,MPF
   KP=P(I)-1.0
   K=MP+1+KP-KK
22 CALL CCRC(X(K-1,I-1),Y(K-1,I-1),U(K-1,I-1),V(K-1,I-1),X(K,I),
  1            Y(K,I),U(K,I),V(K,I),SIGN)
   CALL EQUK(K-1,I,XK,YK,UK,VK)
24 CALL CFPR(X(K-1,I),-Y(K-1,I),U(K-1,I),-V(K-1,I))
   PSIAV=.5*(PSIR(I)+PSI)
   IF (NDIM .GT. 2) GO TO 26
   REF=Y(K-1,I)-Y(K,I)
   GO TO 28
26 REF=Y(K-1,I)**2-Y(K,I)**2
28 STF=STF+PSIAV*REF
   PSIR(I)=PSI
   IF (STFR-STF) 30,32,34
30 CALL CCSR(STFR,STF,X(K-1,I),-Y(K-1,I),U(K-1,I),-V(K-1,I),X(K,I),
  1            -Y(K,I),U(K,I),-V(K,I))
   CALL EQUI(I,XK,-YK,UK,-VK)
   GO TO 36
32 CALL EQUI(I,X(K-1,I),Y(K-1,I),U(K-1,I),V(K-1,I))
   GO TO 36
34 CONTINUE
36 CONTINUE
   DO 38 J=1,KS
   K=(KS+1)-J
   IF (X(K,IS) .EQ. 0.0) GO TO 40
38 CONTINUE
40 KR=K+1
   IR=IS
   REG=-1.0
```

```
      CALL CURVE(X(KS,IS),Y(KS,IS),V(KS,IS)/U(KS,IS),X(KREF,IREF),
     1           Y(KREF,IREF),V(KREF,IREF)/U(KREF,IREF),EXP,NS)
      NS=NS+1
      REG=1.0
      CALL CURVE(XB(IS),YB(IS),VB(IS)/UB(IS),XB(IREF),YB(IREF),
     1           VB(IREF)/UB(IREF),3.0,NR)
      NR=NR+1
      WRITE (6,114)
      WRITE (6,118)
      WRITE (6,108)
      WRITE (6,110) AMT,THROAT,ANGLE
      WRITE (6,120) RECV(2,IREV)
      WRITE (6,122) RECV(1,IREV)
      WRITE (6,104)
      WRITE (6,100)
      DO 42 I=IR,IREF
      IF (XB(I) .EQ. 0.0) GO TO 42
      CALL CFPR(XB(I),YB(I),UB(I),VB(I))
      PRHO=FACTR*PRES
      PRPO=PRHO/CONVR
      WRITE (6,102) XB(I),YB(I),Q,THETA,UB(I),VB(I),AM,PRHO,PRPO
   42 CONTINUE
      WRITE (6,114)
      WRITE (6,106)
      WRITE (6,100)
      DO 46 K=KR,KREF
      WRITE (6,116)
      DO 46 I=IR,IREF
      IF (X(K,I) .EQ. 0.0) GO TO 46
      TEST=ABS(X(K,I)-X(KS,IS))
      IF (TEST .GT. 0.0) GO TO 44
      IF (POINT .EQ. 0.0) GO TO 44
      WRITE (6,112) X(K,I),Y(K,I)
      POINT=0.0
   44 CALL CFPR(X(K,I),Y(K,I),U(K,I),V(K,I))
      PRHO=FACTR*PRES
      PRPO=PRHO/CONVR
      WRITE (6,102) X(K,I),Y(K,I),Q,THETA,U(K,I),V(K,I),AM,PRHO,PRPO
   46 CONTINUE
      IF (XB(IREF) .LT.X(KS,IS)) GO TO 47
      CALL ENDI(KS,IS,SIGN)
   47 DO 48 J=1,KREF
      K=(KREF+1)-J
      IF (X(K,IREF) .EQ. 0.0) GO TO 50
   48 CONTINUE
   50 KP=K+1
      DO 52 J=1,50
      K=(KP-1)+J
      IF (X(K,IREF) .EQ. 0.0) GO TO 54
      CALL EQUK(J,1,X(K,IREF),Y(K,IREF),U(K,IREF),V(K,IREF))
   52 CONTINUE
   54 KREF=J-1
      CALL EQUI(1,XB(IREF),YB(IREF),UB(IREF),VB(IREF))
      IREF=1
      DO 58 K=1,50
      DO 58 I=1,50
      IF (K .GT. KREF) GO TO 56
      IF (I .EQ. 1) GO TO 58
   56 CALL EQUK(K,I,0.0,0.0,0.0,0.0)
   58 CONTINUE
      DO 60 I=2,50
      CALL EQUI(I,0.0,0.0,0.0,0.0)
   60 CONTINUE
      RETURN
      END
```

```
$IBFTC ENDIR   LIST,REF,DECK,DEBUG
C
C
      SUBROUTINE ENDI(KA,IA,SIGN)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1       Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2       XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3       X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4       U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5       XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6       REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//38X,46HCONDITIONS IN THE VICINITY OF THE NORMAL SHOCK//)
  102 FORMAT (//29X,1HX,12X,1HY,10X,8HMACH NO.,8X,5HTHETA,9X,5HP2/P1,9X,
     14HH/HO//)
  104 FORMAT (21X,1P6E14.5//)
  200 FORMAT (1H1)
      PI=3.1415927
      CONVP=(1.0+(GAM-1.0)/2.0)**(-GAM/(GAM-1.0))
      RAVE=FACTR/CONVP
      KR=KA-1
      CALL CFPR(X(KA,IA),Y(KA,IA),U(KA,IA),V(KA,IA))
      SLOPE=TAN(PI/2.0+THETA)
      CALL OSWR(AM,PI/2.0)
      RECOV=RAVE*RECOV
      WRITE (6,200)
      WRITE (6,100)
      WRITE (6,102)
      WRITE (6,104) X(KA,IA),Y(KA,IA),AM,THETA,RPRES,RECOV
      XREF=X(KA,IA)
      YREF=Y(KA,IA)
      DO 18 KK=1,KR
      K=KA-KK
      XR=XREF
      YR=YREF
      DO 16 J=1,2
      RMIN=10.0
      DO 10 I=1,IREF
      IF (X(K,I) .EQ. 0.0) GO TO 10
      RADIUS=SQRT((X(K,I)-XR)**2+(Y(K,I)-YR)**2)
      RMIN=AMIN1(RMIN,RADIUS)
   10 CONTINUE
      DO 12 I=1,IREF
      IF (X(K,I) .EQ. 0.0) GO TO 12
      RADIUS=SQRT((X(K,I)-XR)**2+(Y(K,I)-YR)**2)
      IF (RMIN .EQ. RADIUS) GO TO 14
   12 CONTINUE
   14 IF (X(K,I+1) .EQ. 0.0) GO TO 20
      CALL BOUND(K,I,K,I+1,XREF,YREF,SLOPE,SIGN)
      TEST=XK-X(K,I)
      ISIGN=TEST/ABS(TEST)
      IP=I+ISIGN
      IF (X(K,IP) .EQ. 0.0) GO TO 20
      CALL BOUND(K,I,K,IP,XREF,YREF,SLOPE,SIGN)
      XR=XK
      YR=YK
   16 CONTINUE
      XREF=XR
      YREF=YR
      CALL CFPR(XK,YK,UK,VK)
      SLOPE=TAN(PI/2.0+THETA)
```

```
      CALL OSWR(AM,PI/2.0)
      RECOV=RAVE*RECOV
      WRITE (6,104) XK,YK,AM,THETA,RPRES,RECOV
   18 CONTINUE
   20 RMIN=10.0
      DO 22 I=1,IREF
      RADIUS=SQRT((XB(I)-XREF)**2+(YB(I)-YREF)**2)
      RMIN=AMIN1(RMIN,RADIUS)
   22 CONTINUE
      DO 24 I=1,IREF
      RADIUS=SQRT((XB(I)-XREF)**2+(YB(I)-YREF)**2)
      IF (RMIN .EQ. RADIUS) GO TO 26
   24 CONTINUE
   26 TEST=XREF-XB(I)
      ISIGN=TEST/ABS(TEST)
      J=I+ISIGN
      DU=UB(J)-UB(I)
      DV=VB(J)-VB(I)
      DX=XB(J)-XB(I)
      DY=YB(J)-YB(I)
      DS=SQRT(DX**2+DY**2)
      DUDS=DU/DS
      DYDX=DY/DX
      XK=(YB(I)-YREF+SLOPE*XREF-DYDX*XB(I))/(SLOPE-DYDX)
      YK=YB(I)+DYDX*(XK-XB(I))
      DELS=SQRT((XK-XB(I))**2+(YK-YB(I))**2)
      UK=UB(I)+DUDS*DELS
      VK=UK*DYDX
      CALL CFPR(XK,YK,UK,VK)
      SLOPE=TAN(PI/2.0+THETA)
      CALL OSWR(AM,PI/2.0)
      RECOV=RAVE*RECOV
      WRITE (6,104) XK,YK,AM,THETA,RPRES,RECOV
      RETURN
      END
```

```
$IBFTC ENDSR   LIST,REF,DECK,DEBUG
C
C
      SUBROUTINE ENDS(KA,IA,SIGN)
      COMMON AMO,GAM,THETC,DELB,DELU1,DELU,UC,VC,DELC,BS,DELS,ERROR,
     1      Q,AM,AMU,THETA,A,B,C,D,E,F,G,PRES,DENS,AREA,DYNP,CP,PSI,
     2      XK,YK,UK,VK,XJ,YJ,UJ,VJ,XA,YA,UA,VA,DELA,BETAK,SIGK,SIGSQ,
     3      X(50,50),Y(50,50),U(50,50),V(50,50),BETA(50),PSIR(50),
     4      U1,V1,DEL1,DELY,RPRES,RDENS,RECOV,ENTP,NDIM,KREF,IREF,
     5      XB(50),YB(50),UB(50),VB(50),P(50),RECV(2,10),FACTR,IREV,
     6      REG,R(25,4),S(25,4),COWL(25),BODY(25),NR,NS,ICOWL,IBODY
  100 FORMAT (//38X,46HCONDITIONS IN THE VICINITY OF THE NORMAL SHOCK//)
  102 FORMAT (//29X,1HX,12X,1HY,10X,8HMACH NO.,8X,5HTHETA,9X,5HP2/P1,9X,
     14HH/HO//)
  104 FORMAT (21X,1P6E14.5//)
  200 FORMAT (1H1)
      TEST=0.01745329
      PI=3.1415927
      CONVP=(1.0+(GAM-1.0)/2.0)**(-GAM/(GAM-1.0))
      RAVE=FACTR/CONVP
      KREF=KA
      IREF=IA
      XK=X(IREF,KREF)
      YK=Y(IREF,KREF)
      UK=U(IREF,KREF)
      VK=V(IREF,KREF)
      XP=XK
      YP=YK
      CALL CFPR(XK,YK,UK,VK)
      IF (ABS(THETA) .GT. TEST) GO TO 6
      SLOPE=0.0
      GO TO 8
    6 SLOPE=TAN(PI/2.0+THETA)
    8 CALL OSWR(AM,PI/2.0)
      RECOV=RAVE*RECOV
      WRITE (6,200)
      WRITE (6,100)
      WRITE (6,102)
      WRITE (6,104) XK,YK,AM,THETA,RPRES,RECOV
      IF=KA-1
      DO 14 I=1,IF
      IF (KREF .EQ. 50) GO TO 16
      CALL SEEK(KREF,IREF,XP,YP,SLOPE,SIGN)
      CALL CFPR(XK,YK,UK,VK)
      IF (ABS(THETA) .GT. TEST) GO TO 10
      SLOPE=0.0
      GO TO 12
   10 SLOPE=TAN(PI/2.0+THETA)
   12 CALL OSWR(AM,PI/2.0)
      RECOV=RAVE*RECOV
      WRITE (6,104) XK,YK,AM,THETA,RPRES,RECOV
      XP=XK
      YP=YK
   14 CONTINUE
   16 RETURN
      END
```

REFERENCES

1. Courant, Richard; and Friedrichs, K. O.: Supersonic Flow and Shock Waves. Interscience Publ., 1948.
2. Sorensen, Virginia L.: Computer Program for Calculating Flow Fields in Supersonic Inlets. NASA TN D-2897, 1965.
3. Kennedy, E. C.: Calculation of the Flow Fields Around a Series of Bi-Cone Bodies of Revolution Using the Method of Characteristics as Applied to Supersonic Rotational Flow. Rep. OAL/CM 873, Consolidated Vultee Aircraft Co., June 1956.
4. Kennedy, Ernest C.: Calculation of Axisymmetric Isentropic Spike Surfaces. J. Aero/Space Sci., vol. 25, no. 7, July 1958, pp. 463-464.
5. Ames Research Staff: Equations, Tables, and Charts for Compressible Flow. NACA Rept. 1135, 1953.
6. Calogeras, James E.; and Meleason, Edward T.: Wind Tunnel Investigation of Techniques for Reducing Cowl Drag of an Axisymmetric External-Compression Inlet at Mach 2.49. NASA TM X-1516, 1968.
7. Connors, James F.; and Woollett, Richard R.: Characteristics of Flow About Axially Symmetric Isentropic Spikes for Nose Inlets at Mach Number 3.85. NACA RM RM E54F08, 1954.
8. Cubbison, Robert W.; Meleason, Edward T.; and Johnson, David F.: Effect of Porous Bleed in a High-Performance Axisymmetric, Mixed-Compression Inlet at Mach 2.50. NASA TM X-1692, 1968.
9. Evvard, John C.; and Maslen, Stephen H.: Three-Dimensional Supersonic Nozzles and Inlets of Arbitrary Exit Cross Section. NASA TN 2688, 1952.

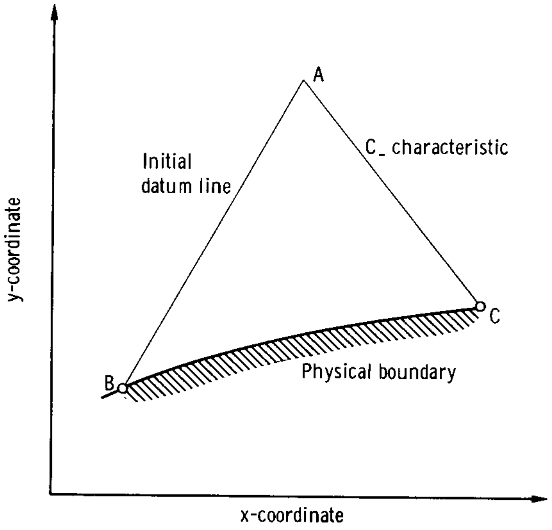


Figure 1. - Domain in the x, y plane in which solution can be established.

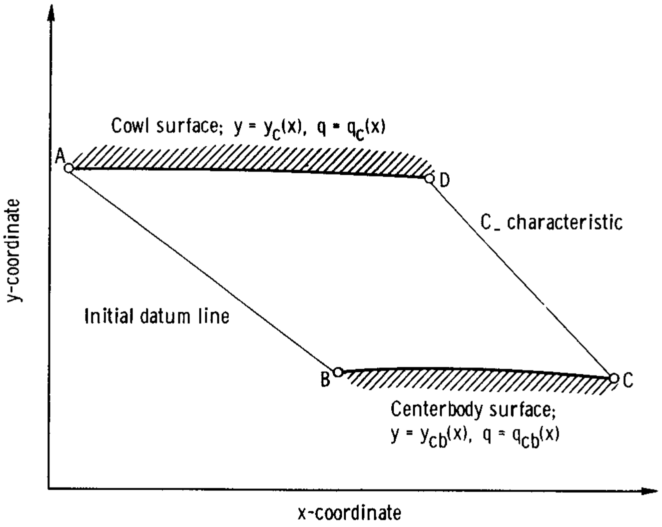


Figure 2. - Domain in x, y plane in which solution can be established for free-boundary problem.

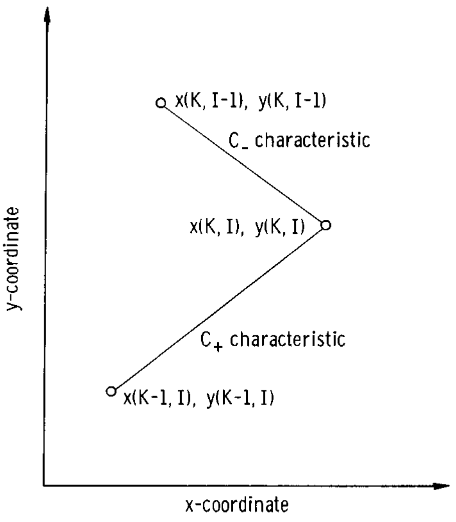


Figure 3. - Location of basic field point.

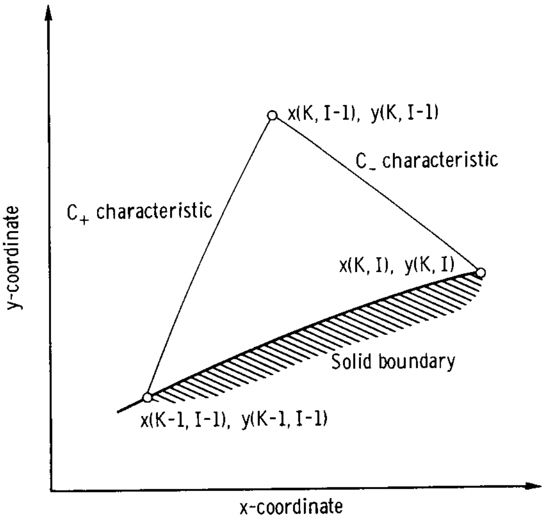


Figure 4. - Location of solid boundary point.

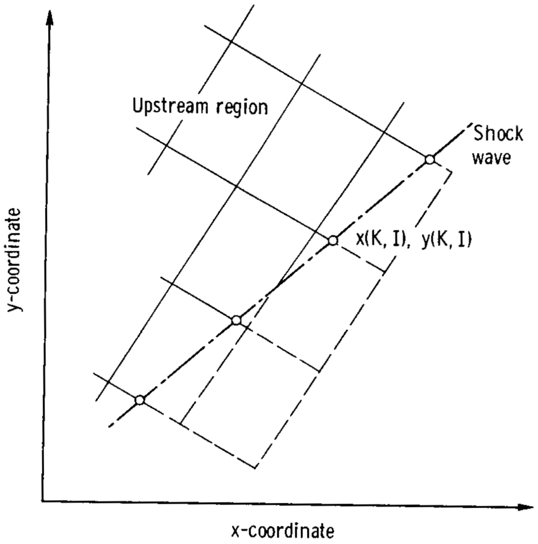


Figure 5. - Location of shock points in upstream region.

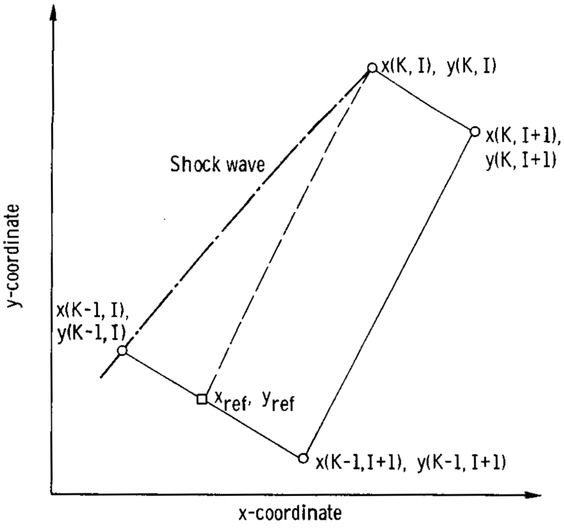


Figure 6. - Location of shock-field points.

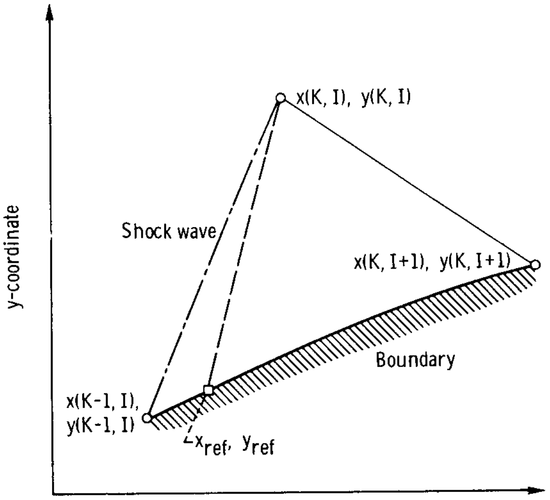


Figure 7. - Location of shock-boundary points.

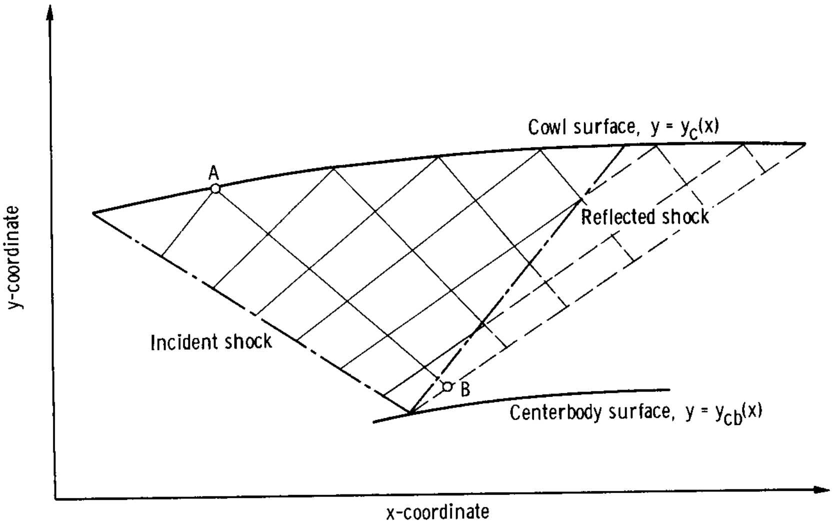


Figure 8. - Fixed-boundary problem.

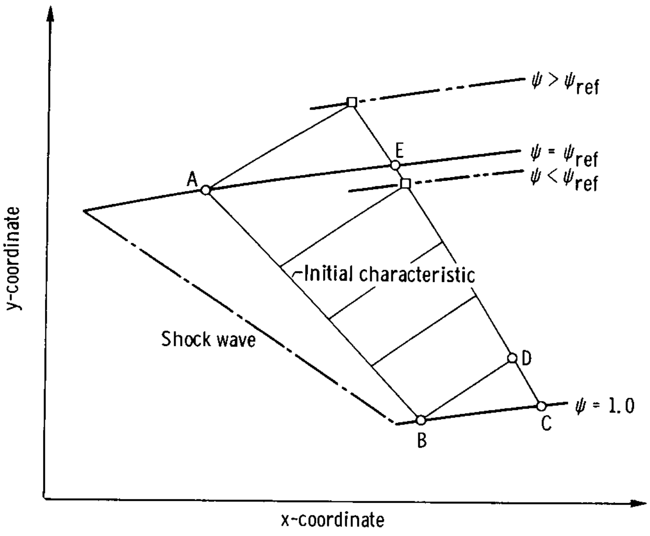


Figure 9. - Free-boundary problem.

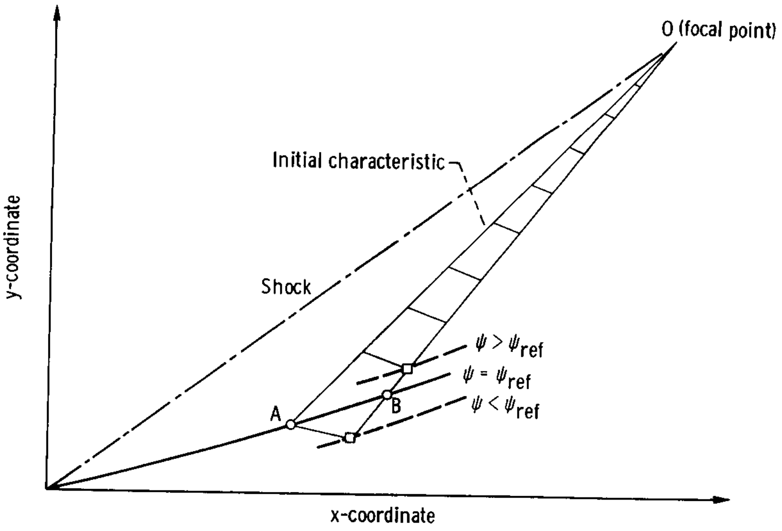
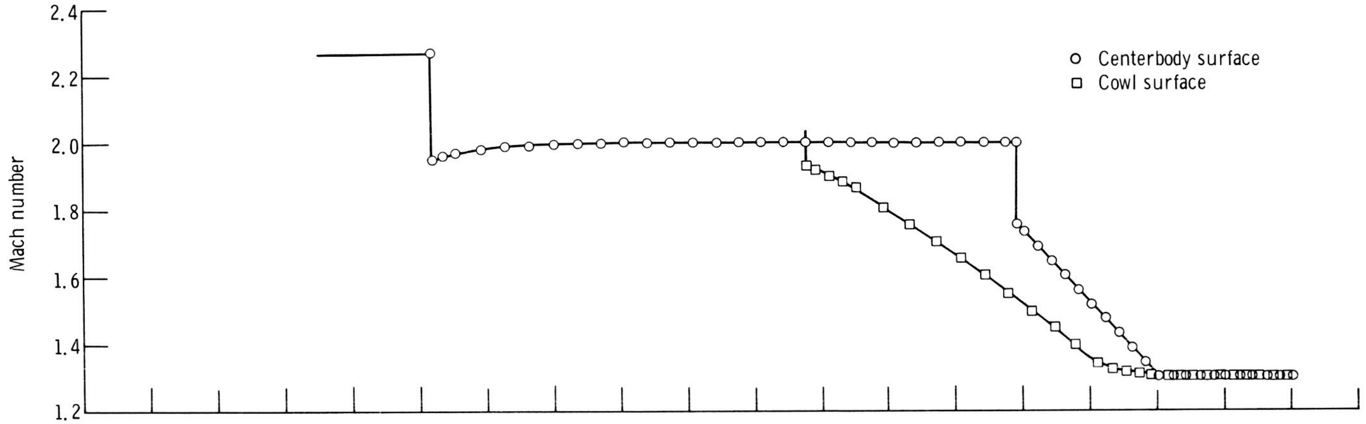
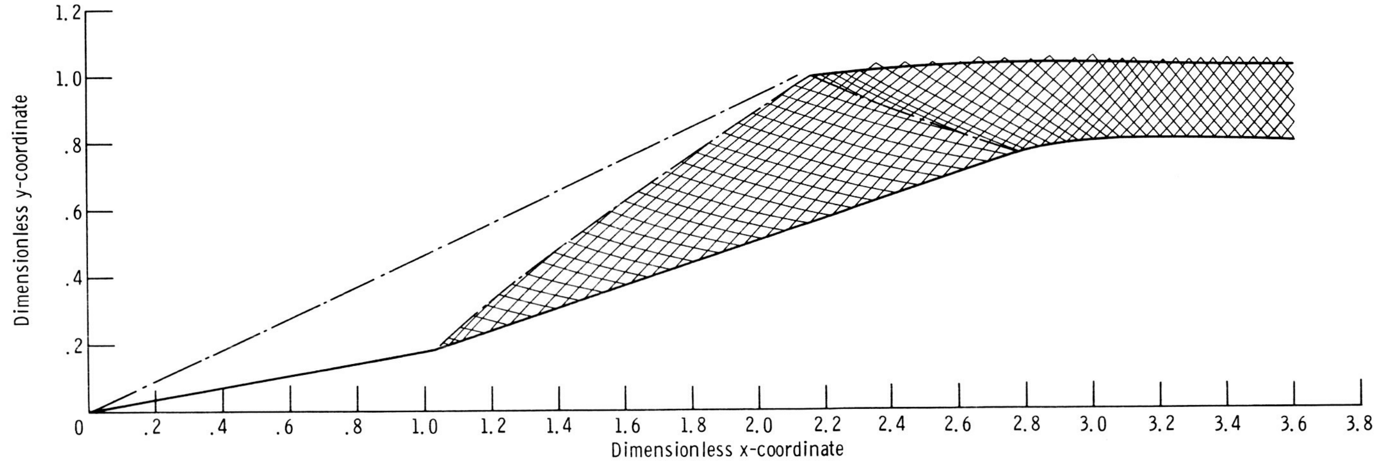


Figure 10. - Isentropic spike problem.

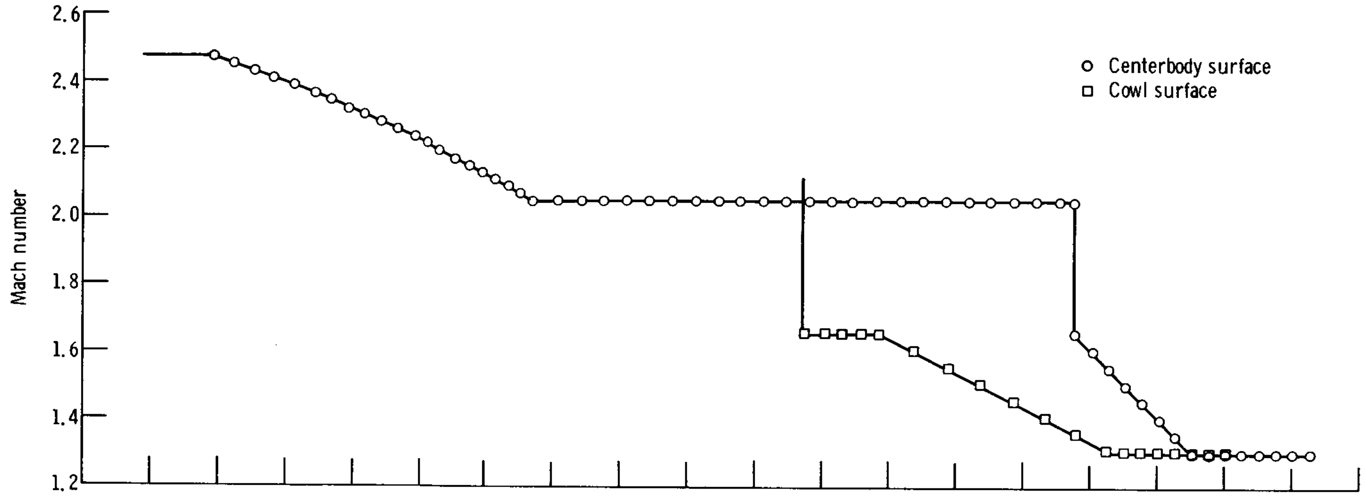


(a) Mach number distribution.

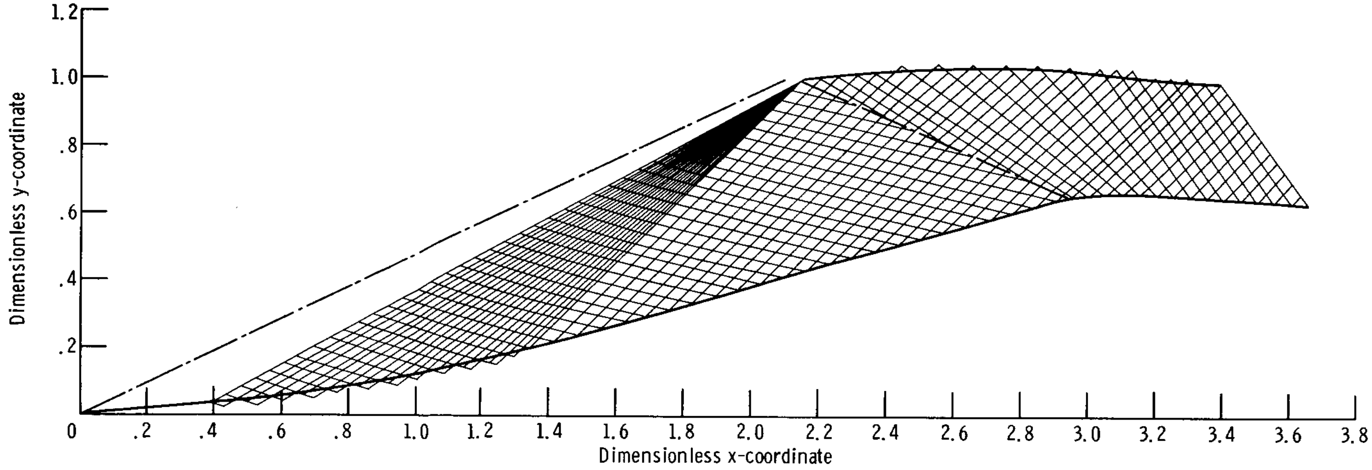


(b) Characteristic solution.

Figure 11. - Bicone mixed-compression inlet (10.0°-18.5°) designed for free-stream Mach number of 2.500. Cowl angle, 5.0°; throat Mach number, 1.300.

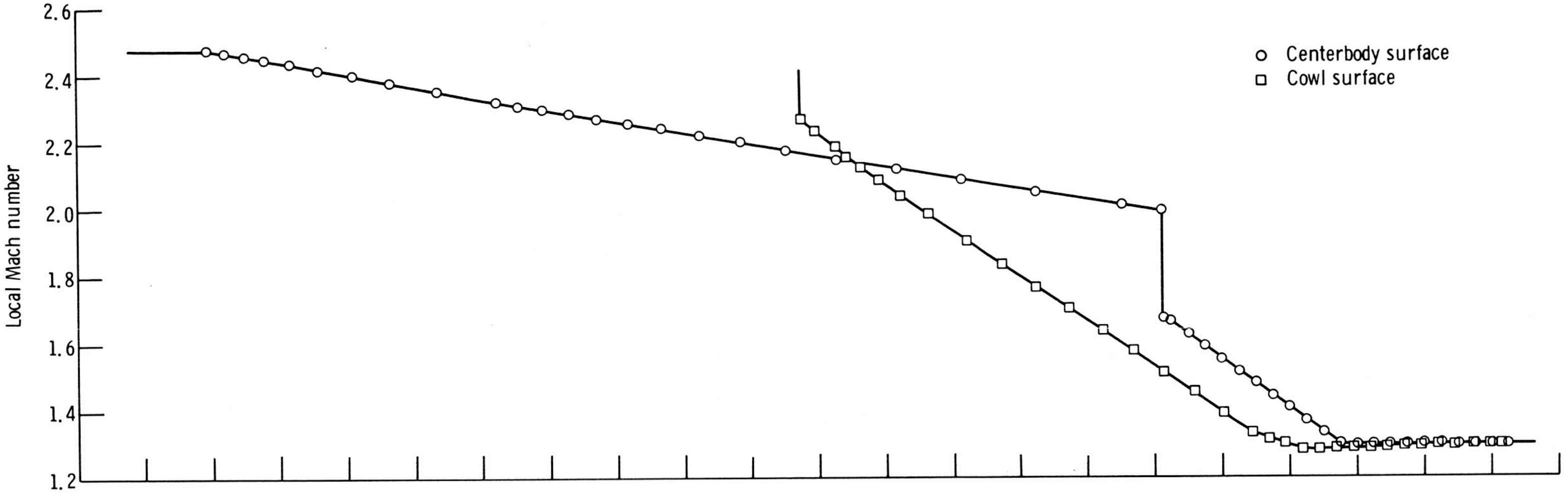


(a) Mach number distribution.

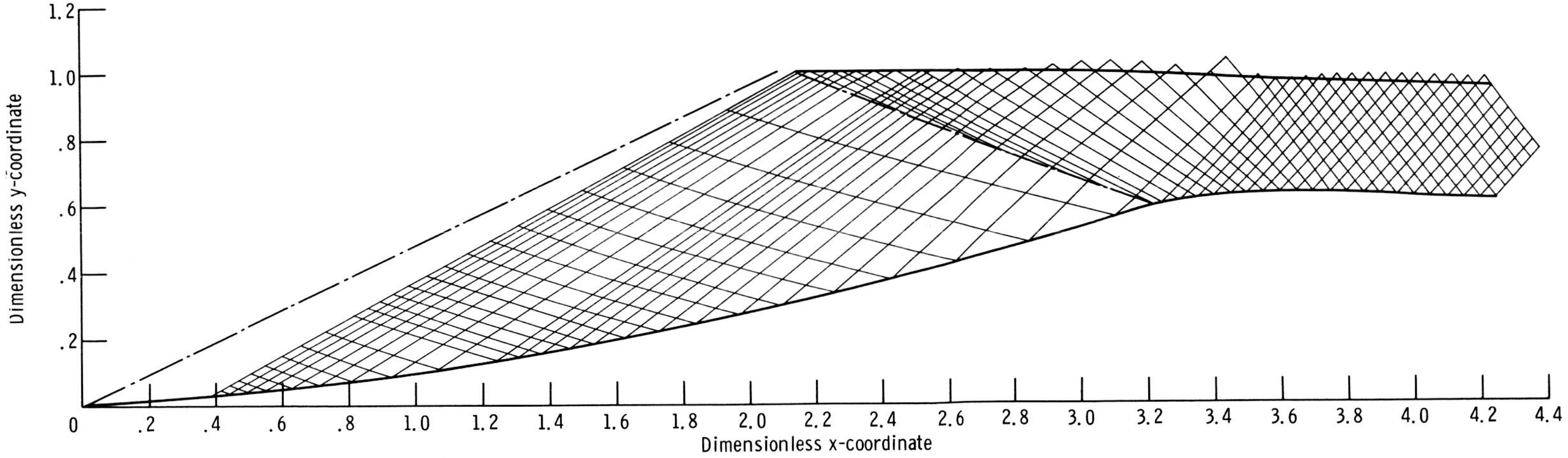


(b) Characteristic solution.

Figure 12. - Isentropic ramp mixed-compression inlet designed for free-stream Mach number of 2.700. Initial ramp angle, 5.0°; cowl angle, 5.0°; throat Mach number, 1.300.

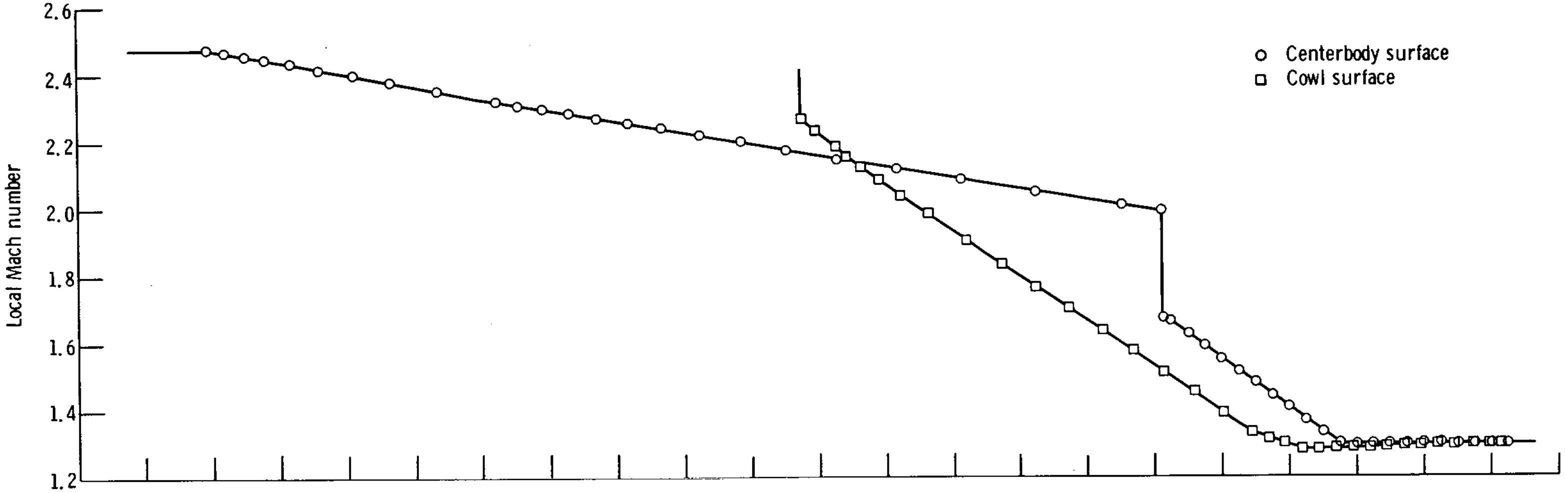


(a) Mach number distribution.

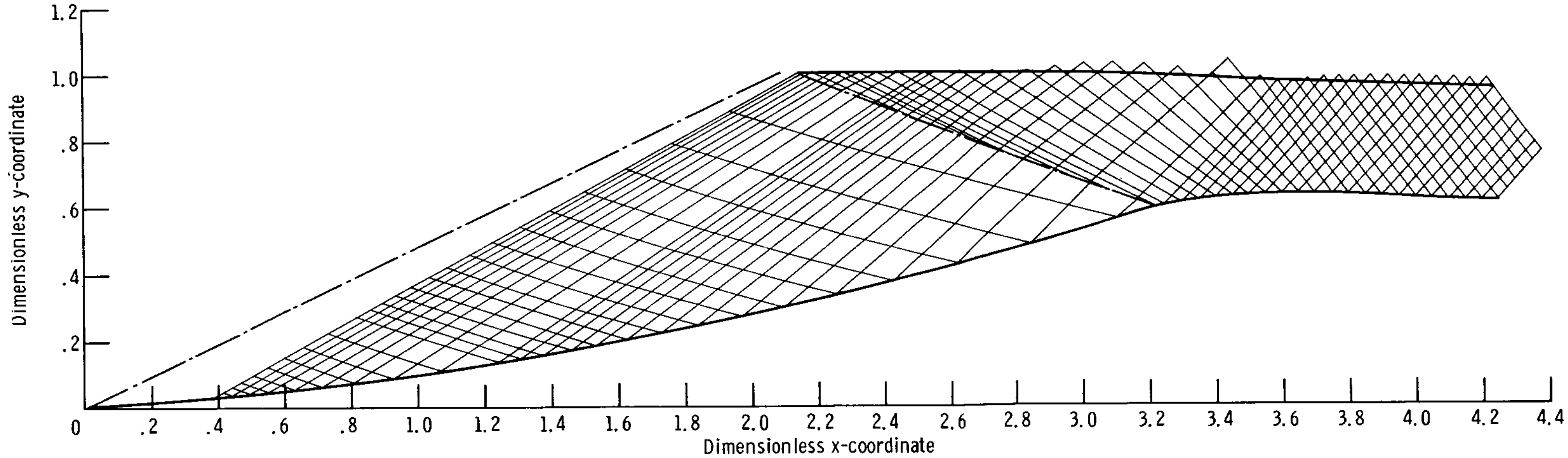


(b) Characteristic solution.

Figure 13. - Two-dimensional isentropic ramp inlet designed for free-stream Mach number of 2.700. Initial ramp angle, 5.0°; cowl angle, 0°; throat Mach number, 1.300.

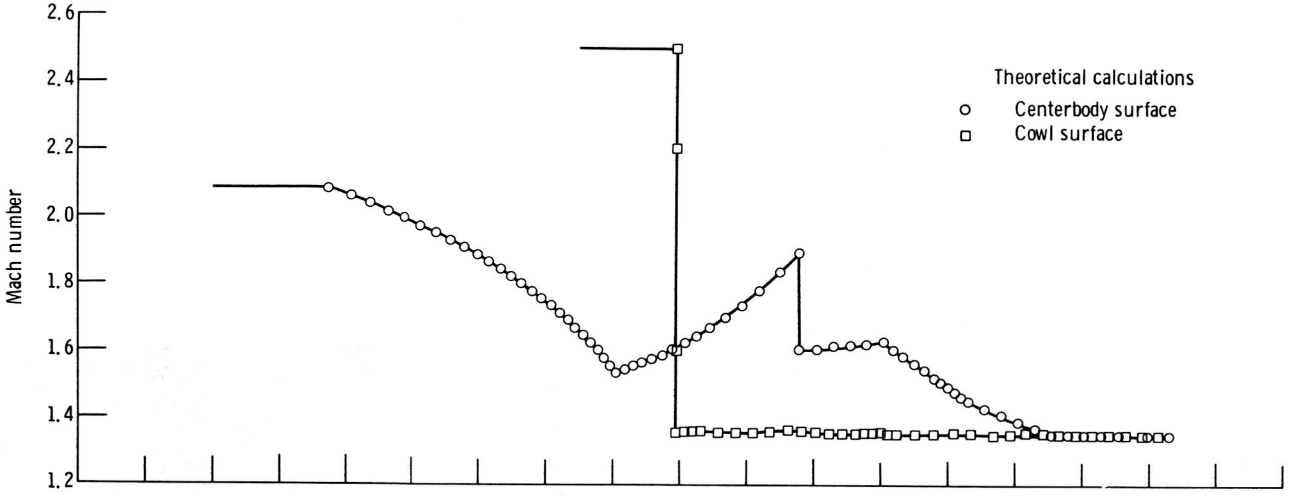


(a) Mach number distribution.

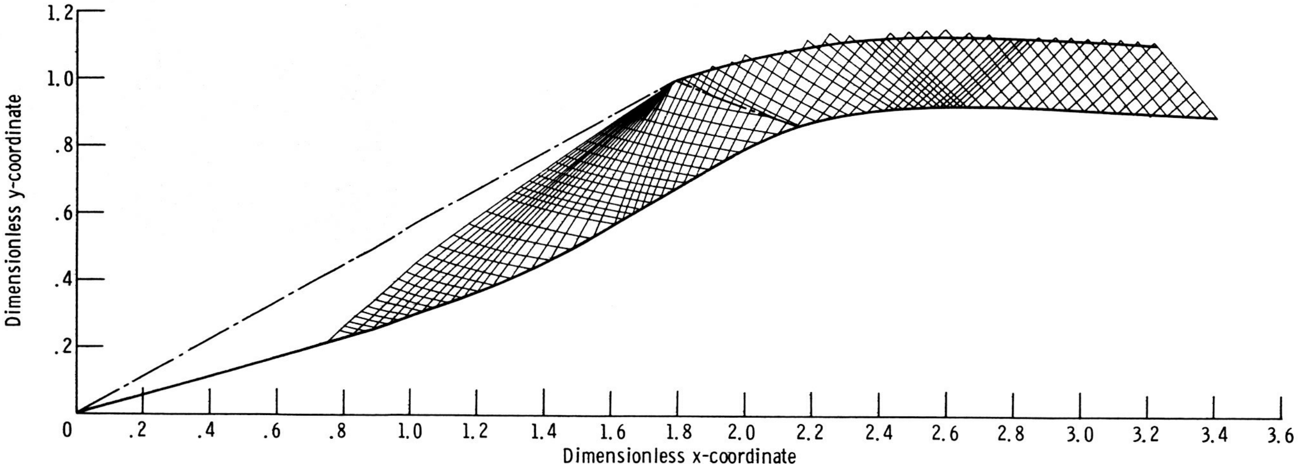


(b) Characteristic solution.

Figure 13. - Two-dimensional isentropic ramp inlet designed for free-stream Mach number of 2.700. Initial ramp angle, 5.0°; cowl angle, 0°; throat Mach number, 1.300.

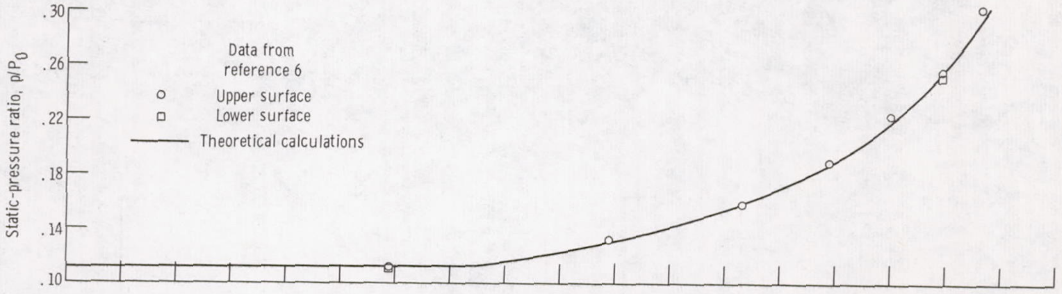


(a) Mach number distribution.

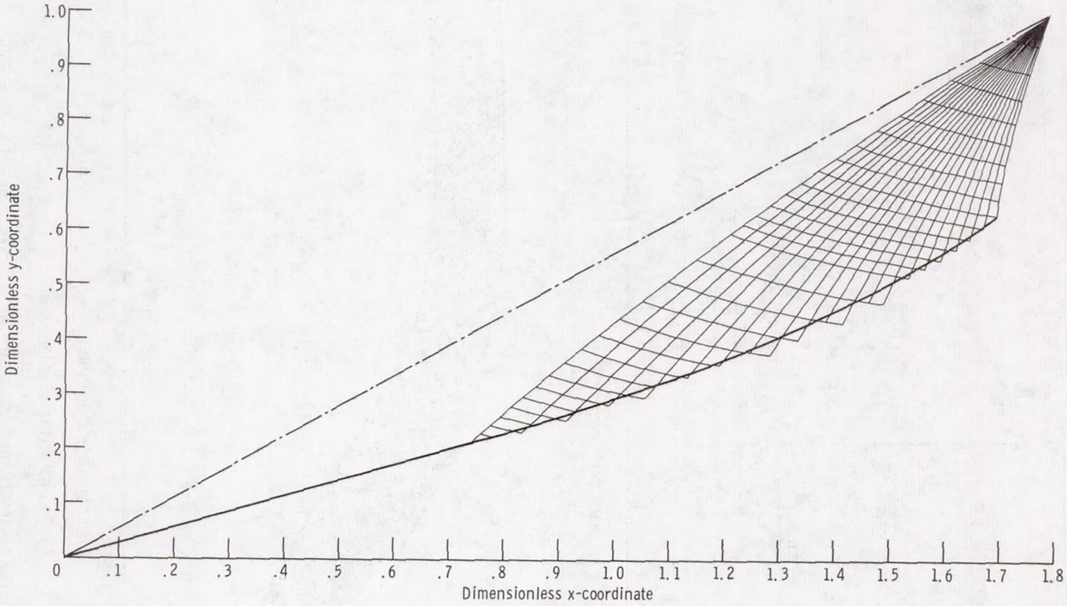


(b) Characteristic solution.

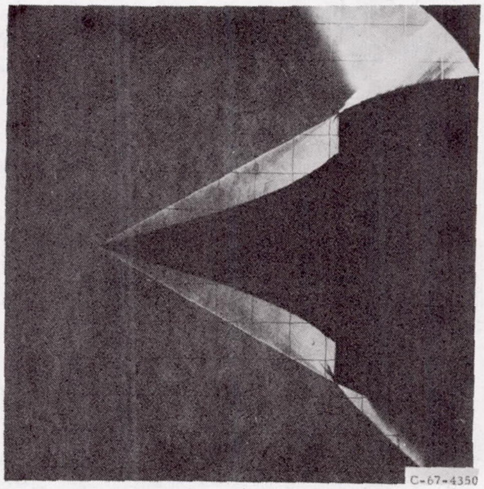
Figure 14. - Isentropic spike self-starting inlet designed for free-stream Mach number of 2.500. Initial half cone angle, 16.0°; cowl angle, 17.0°; throat Mach number, 1.35.



(a) Static-pressure distribution.

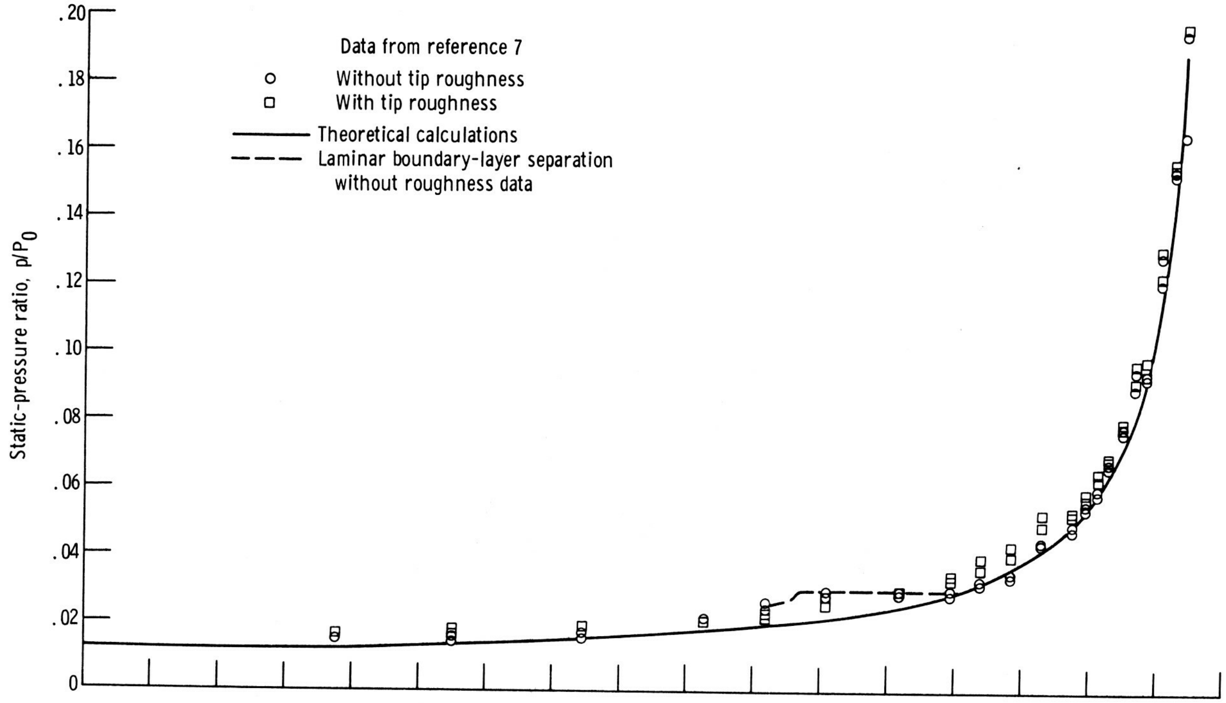


(b) Characteristic solution.

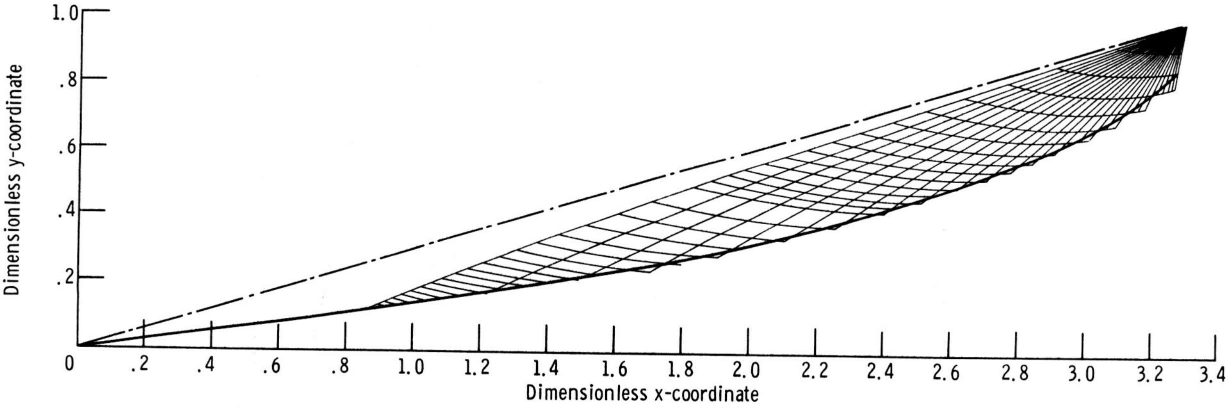


(c) Schlieren photograph showing isentropic spike cone.

Figure 15. - Isentropic spike configuration designed for free-stream Mach number of 2.49 and focal point Mach number of 1.46. Initial cone angle, 16.0°.

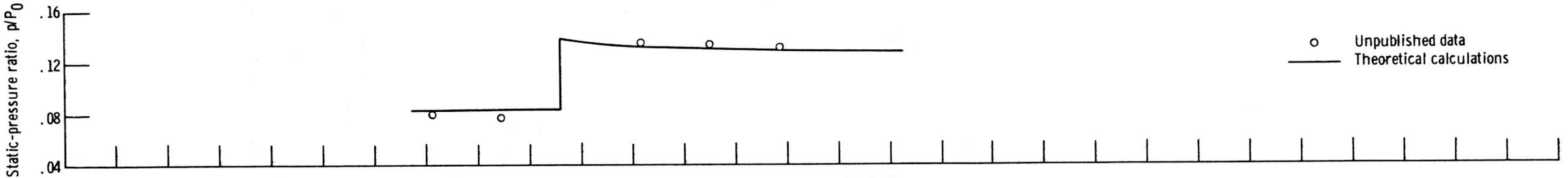


(a) Static-pressure distribution.

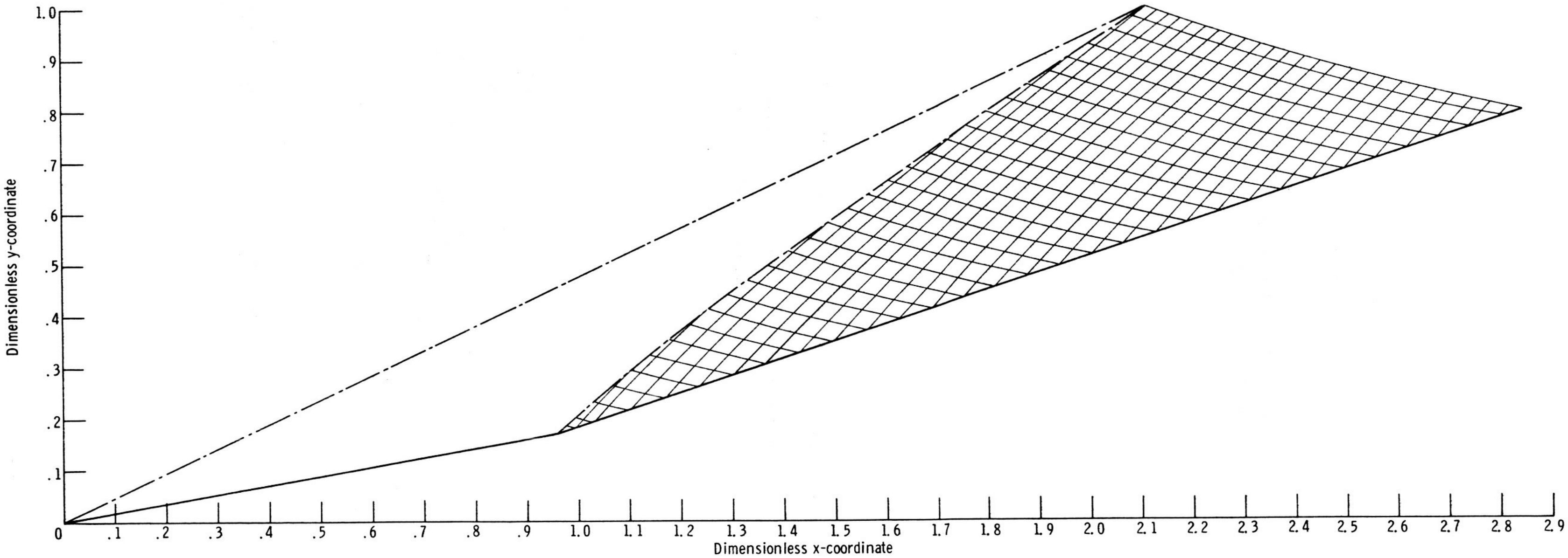


(b) Characteristic solution.

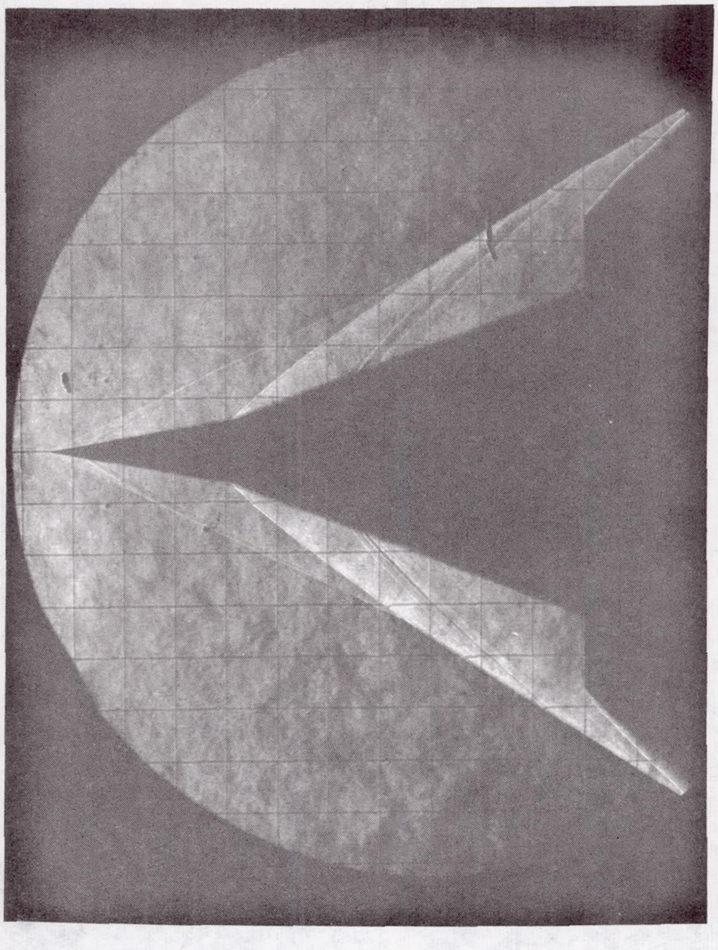
Figure 16. - Isentropic spike configuration designed for free-stream Mach number of 3.85 and focal point Mach number of 1.75. Initial cone angle, 8.0°.



(a) Static-pressure distribution.

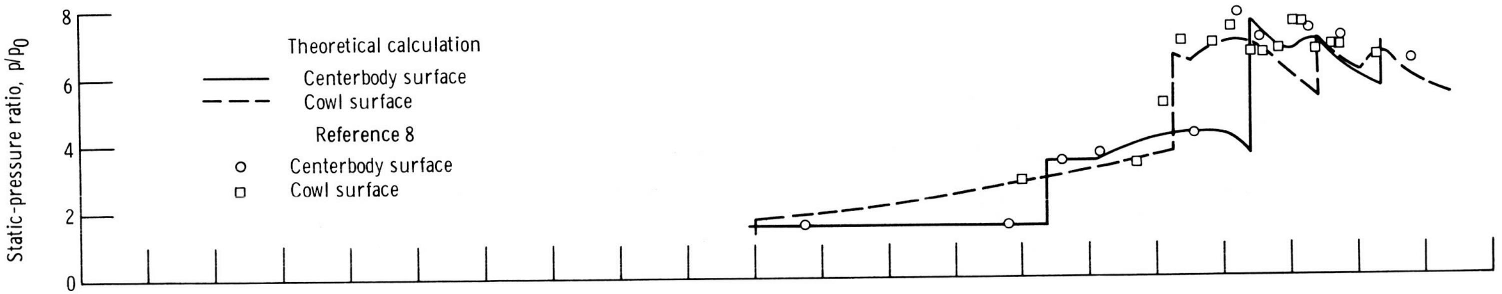


(b) Characteristic solution.

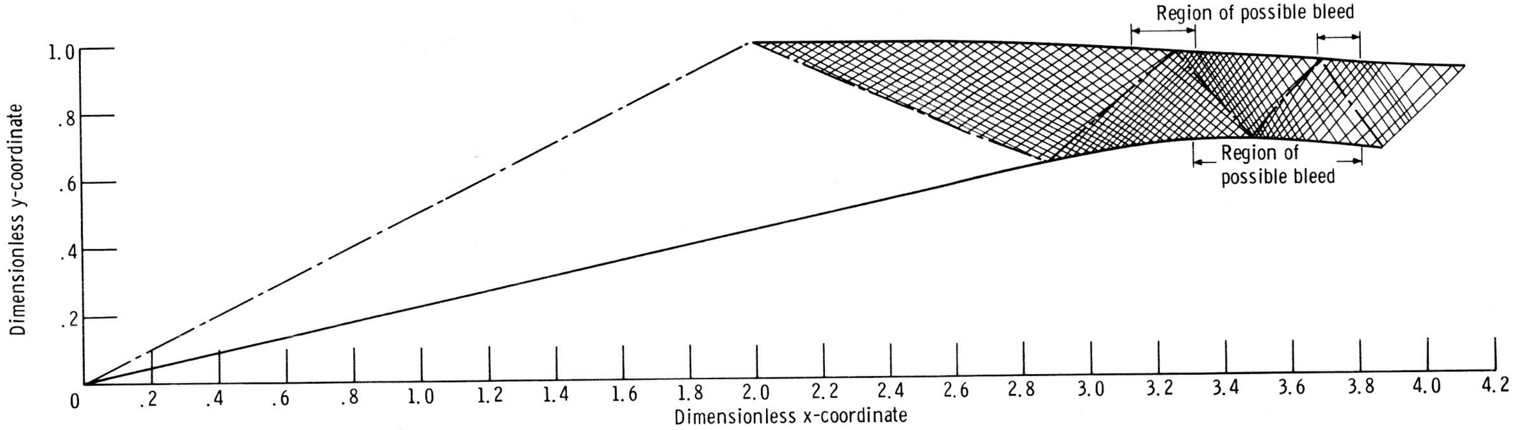


(c) Schlieren photograph of bicone configuration.

Figure 17. - Bicone forebody (10°-18.5°) configuration for free-stream Mach number of 2.49.



(a) Static-pressure distribution.



(b) Characteristic solution.

Figure 18. - Single-cone (12.5°) mixed-compression inlet designed for free-stream Mach number of 2.500.

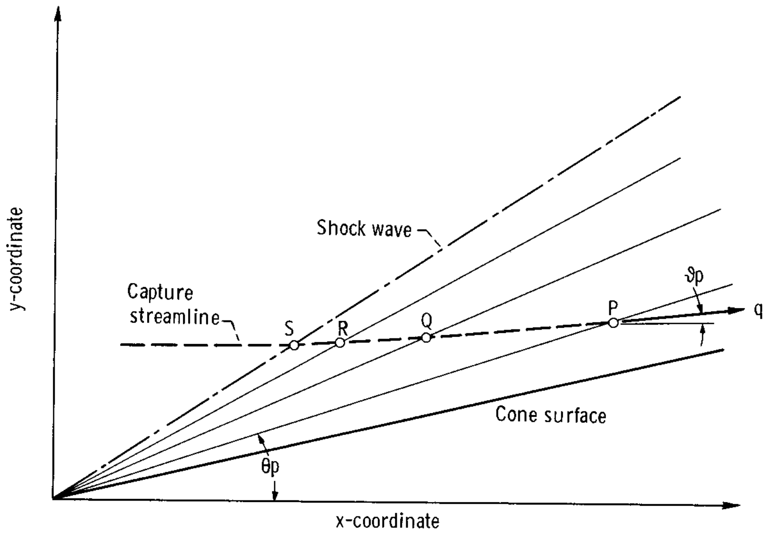


Figure 19. - Capture stream line used for additive drag calculations.