AD623 298

D1-82-0462

BOEING SCIENTIFIC RESEARCH LABORATORIES

An IBM 7090 Program for Computing Two-Dimensional and Axially-Symmetric Supersonic Flow of an Ideal Gas

Supplement I. Subroutines for Computing the Shock and Free Streamline Dividing Two Regions of Non-Uniform Supersonic Flow

F. Edward Ehlers

Mathematics Research

August 1965

AN IBM 7090 PROGRAM FOR COMPUTING TWO-DIMENSIONAL AND AXIALLY-SYMMETRIC SUPERSONIC FLOW OF AN IDEAL GAS

SUPPLEMENT I. SUBROUTINES FOR COMPUTING THE SHOCK AND FREE STREAMLINE DIVIDING TWO REGIONS OF NON-UNIFORM SUPERSONIC FLOW

by

F. Edward Ehlers

Mathematical Note No. 422

Mathematics Research Laboratory

BOEING SCIENTIFIC RESEARCH LABORATORIES

August 1965

ABSTRACT

Methods for computing a shock wave and also a free streamline dividing two regions of isoenergetic supersonic flow are explained with computed examples of both two-dimensional and axially-symmetric flows. The subroutines as coded in FORTRAN language for the IBM 7094 are listed and the procedures for their use are described.

I. INTRODUCTION

In Ref. [1], an IBM 7090 program for computing the supersonic flow of an ideal gas by the method of characteristics was discussed. A theory was also presented for a subroutine (SHOCK) for computing the next point along a shock wave provided the region of flow ahead of the shock is uniform. Another subroutine was also included for computing the next point along a free streamline bounding a supersonic flow and quiescient air. In this document, we develop the theory for computing the next point along the shock when the flow ahead of the shock is also non-uniform and for computing the next point along a free streamline bounding two non-uniform supersonic streams. Subroutines applying the theory have been coded in FORTRAN and tested in practice.

Both the shock and free streamline subroutines make use of interpolation in an upstream Mach net and a procedure is described for selecting the appropriate points. The routines required for computing the non-uniform flow over a two-dimensional or axially-symmetric body with attached shock are explained and a FORTRAN listing is given in Appendix I. To illustrate the method, the flows over a wedge in source flow and over a body consisting of two conical segments were computed and some typical results are given.

To compute the free boundary between a supersonic jet and an exterior supersonic flow, some of the subroutines described in Ref. [1]

are modified. The subprograms of Ref. [1] to be changed and subroutines to be added to the complete program are listed in Appendix II. With these changes, the program system is capable of computing all flows described in Ref. [1] except the free streamline boundary in still air and in addition can compute the exterior supersonic stream and upper free streamline boundary between jet and exterior flow. A portion of the flow field for an axially-symmetric jet embedded in supersonic source flow was computed to illustrate the method.

II. CONTINUITY CONDITIONS ACROSS A SHOCK

In a manner similar to the theory in Ref. [1], [2], we introduce the variables

$$\xi = \tan \omega \qquad \zeta = \tan \theta \qquad \beta = \sqrt{M^2 - 1}$$

where ω and θ are the local shock angle and flow angle, respectively, and M is the Mach number. The general conditions of continuity of mass, momentum, and energy across the shock wave are given by

$$\rho_0 V_0 \cdot N = \rho_1 V_1 \cdot N \tag{1}$$

$$V_0 \times N = V_1 \times N \tag{2}$$

$$p_0 + \rho_0 (V_0 \cdot N)^2 = p_1 + \rho_1 (V_1 \cdot N)^2 \tag{3}$$

$$i_0 + (V_0 \cdot N)^2/2 = i_1 + (V_1 \cdot N)^2/2 \tag{4}$$

where ρ, p, and V are the density, pressure, and velocity vector, respectively. The quantity i is the enthalpy, N the unit vector normal to the shock, and the subscripts 0 and 1 denote quantities ahead of and behind the shock, respectively. Equation (1) may be written

$$\frac{\rho_0}{\rho_1} = \frac{V_1 \cdot N}{V_0 \cdot N} = \frac{u_1 \sin\omega - v_1 \cos\omega}{u_0 \sin\omega - v_0 \cos\omega} = \frac{u_1}{u_0}\left(\frac{\xi - \zeta_1}{\xi - \zeta_0}\right) \tag{5}$$

where u and v are the x and y components of velocity (see Fig. 1). In terms of these variables, Eq. (2) becomes

$$u_0 \cos\omega + v_0 \sin\omega = u_1 \cos\omega + v_1 \sin\omega. \tag{6}$$

Solving for u_1/u_0 and substituting into Eq. (5) yields

$$\frac{\rho_0}{\rho_1} = \frac{(1 + \zeta_0\xi)(\xi - \zeta_1)}{(1 + \zeta_1\xi)(\xi - \zeta_0)} = \sigma_1. \tag{7}$$

With

$$\rho_0(V_0 \cdot N)^2/\gamma p_0 = M_0^2(\xi - \zeta_0)^2/(1 + \xi^2)(1 + \zeta_0^2) = \sigma_0,$$

Equation (3) takes the form

$$p_1/p_0 = 1 + \gamma\sigma_0(1 - \sigma_1).$$

Now $i = \gamma p/(\gamma - 1)\rho = c^2/(\gamma - 1) = \gamma RT/(\gamma - 1)$ where γ is the ratio of specific heats, c is the speed of sound, and T is the temperature. Thus Eq. (4) leads to

$$T_1/T_0 = 1 + (\gamma - 1)\sigma_0(1 - \sigma_1^2)/2. \tag{8}$$

From the ideal gas law, we also have

$$\frac{p_1}{p_0}\frac{\rho_0}{\rho_1} = \sigma_1[1 + \gamma\sigma_0(1 - \sigma_1)] = T_1/T_0. \tag{9}$$

Combining the last two equations and solving for σ_1 or σ_0 yield

$$\sigma_1 = 2/(\gamma + 1)\sigma_0 + k$$

or (10)

$$\sigma_0 = 2/(\gamma + 1)(\sigma_1 - k)$$

where $k = (\gamma - 1)/(\gamma + 1)$. Then

$$T_1/T_0 = \sigma_1(1 - k\sigma_1)/(\sigma_1 - k) = \tau \tag{11}$$

A relation for β_1 behind the shock results from using Eq. (11) and rewriting Eq. (6) as

$$(u_1/u_0)^2 = (q_1/q_0)^2(1 + \zeta_0^2)/(1 + \zeta_1^2) = (1 + \zeta_0\xi)^2/(1 + \zeta_1\xi)^2$$

where q is the velocity magnitude. Since

$$q_0^2T_1/q_1^2T_0 = (\beta_0^2 + 1)/(\beta_1^2 + 1)$$

we get

$$\beta_1^2 = (\beta_0^2 + 1)a/\tau - 1 \tag{12}$$

where

$$a = (1 + \zeta_0\xi)^2(1 + \zeta_1^2)/(1 + \zeta_1\xi)^2(1 + \zeta_0^2) \tag{13}$$

with σ_1 known, the velocity slope ζ_1 behind the shock is given by solving Eq. (7) for ζ_1. Thus

$$\zeta_1 = [(1 + \zeta_0\xi)\xi + \sigma_1(\zeta_0 - \xi)]/[(1 + \zeta_0\xi) + \sigma_1\xi(\xi - \zeta_0)].$$

The jump in entropy s across the shock is given by

$$s_1 - s_0 = \log\ (p_1/p_0) + \gamma\ \log\ (\rho_0/\rho_1)$$

or

$$s_1 - s_0 = (\gamma - 1)\ \log\ \sigma_1 + \log\ \tau. \tag{14}$$

Therefore, with β_0, ζ_0, s_0 and ξ known, conditions behind the shock are found by computing sequentially

$$\sigma_0 = (\beta_0^2 + 1)(\xi - \zeta_0)^2/(1 + \xi^2)(1 + \zeta_0^2) \tag{15}$$

$$\sigma_1 = 2/(\gamma + 1)\sigma_0 + k \tag{16}$$

$$\zeta_1 = [(1 + \zeta_0\xi)\xi + \sigma_1(\zeta_0 - \xi)]/[(1 + \zeta_0\xi) + \sigma_1\xi(\xi - \zeta_0)] \tag{17}$$

$$\tau = \sigma_1(1 - k\sigma_1)/(\sigma_1 - k) \tag{18}$$

$$a = (1 + \zeta_0\xi)^2(1 + \zeta_1^2)/(1 + \zeta_1\xi)^2(1 + \zeta_0^2) \tag{19}$$

$$\beta_1 = \sqrt{(\beta_0^2 + 1)a/\tau - 1} \tag{20}$$

$$s_1 = s_0 + (\gamma - 1)\ \log\ \sigma_1 + \log\ \tau \tag{21}$$

III. INTERACTION OF MACH LINES WITH THE SHOCK WAVE

The characteristics and their corresponding compatibility relations have been derived in Ref. [2]. In the notation of Ref. [2], we have

$$E(\zeta)d\zeta \mp F(\beta)d\beta \mp G(\beta)ds \pm jH^{\pm}dx/y = 0 \tag{22}$$

along the Mach lines

$$\frac{dy}{dx} = A^{\pm}(\beta,\zeta), \tag{23}$$

and

$$ds = 0 \tag{24}$$

along the streamline

$$dy/dx = \zeta, \tag{25}$$

where

$$\left.\begin{aligned} A^{\pm}(\beta,\zeta) &= (\beta\zeta \pm 1)/(\beta \mp \zeta) \\ E(\zeta) &= 1/(1 + \zeta^2) \\ F(\beta) &= 2\beta^2/\{(\beta^2 + 1)[(\gamma - 1)\beta^2 + (\gamma + 1)]\} \\ G(\beta) &= \beta/\gamma(\gamma - 1)(\beta^2 + 1) \\ H^{\pm}(\beta,\zeta) &= \zeta/(\beta \mp \zeta) \end{aligned}\right\} \tag{26}$$

and $j = 0$ or 1 depending upon whether the flow is two-dimensional or axially-symmetric. The plus sign corresponds to Mach lines running upward and to the right and the minus sign to Mach lines running downward to the right.

To facilitate understanding of the procedure for finding the next point along a shock wave, we briefly describe the steps in the calculation, referring to Fig. 2. With the data given at point 2 and at the shock point denoted by subscripts b and s, we estimate the coordinates of point 1 by the linear extension of the shock wave and the upward running mach line 2-1. Then using the data ahead of the shock at the previous point as the first estimate for the new point along the shock, we interpolate to find β, ζ, and s at points m, n, and q, which are the intersections with the known upstream Mach net of the two Mach lines and of the streamline through the estimated shock point. By means of the compatibility conditions along these three new lines, a better estimate of β_0, ζ_0, s_0 ahead of the shock at the new point is found.

Then from the compatibility condition along the Mach line from the downstream point 2 and the continuity conditions across the shock, we obtain an estimate for the shock inclination at the new point by Newton's method, and simultaneously calculate the values of the flow quantities behind the shock. With these estimated values of the data on both sides of the shock at the new shock point, we go through the calculations again with improved coefficients, repeating until the desired accuracy is achieved.

Assume that the values of β, ζ and s are known at one point on both sides of the shock wave, at one point 2 in the flow behind the shock, and at the points, 3, 4, and 5, along two characteristics as shown in Fig. 2. Denote by subscript 1, the quantities on the downstream side at the next point along the shock and by the subscript 0 the quantities on the upstream side. For the first estimate of the values of β_0 and ζ_0, we take the values at the known point b.

The first step in the calculation is to obtain an estimate of x_1, y_1 by the intersection of the straight line extension of the shock and the upward running Mach line from the point 2. This yields

$$x_1 = [y_2 - y_s + \xi_s x_s - A_2^+ x_2]/(\xi_s - A_2^+) \tag{27}$$

$$y_1 = y_2 + A_2^+(x_2 - x_1) \tag{28}$$

where the subscript denotes the point at which the quantity is evaluated. To improve the estimate of the upstream quantities at the new point on the shock, we interpolate the points m, n, and q, which are the points of intersection of the two Mach lines and the streamline running upstream from the point 1 with the Mach lines 3-4 and 5-3. The first estimate of x_m, x_n, and x_q are

$$\begin{aligned} x_m &= (y_1 - y_3 + D_1 x_3 - A_0^- x_1)/(D_1 - A_0^-) \\ x_n &= (y_1 - y_3 + D_1 x_3 - \zeta_0 x_1)/(D_1 - \zeta_0) \\ x_q &= (y_1 - y_3 + D_2 x_3 - A_0^+ x_1)/(D_2 - A_0^+) \end{aligned} \tag{29}$$

where

$$D_1 = (y_3 - y_4)/(x_3 - x_4)$$

and

$$D_2 = (y_5 - y_3)/(x_5 - x_3). \tag{30}$$

The quantities β, ζ, and s are found at these points by linear interpolation; for example,

$$\zeta_m = [\zeta_3(x_m - x_4) + \zeta_4(x_3 - x_m)]/(x_3 - x_4),$$

and since entropy is constant along streamlines,

$$s_0 = s_n = [s_3(x_n - x_4) + s_4(x_3 - x_n)]/(x_3 - x_4).$$

Interpolation of point n is indicated as being on the segment 3-4. For a more contracted net, the streamline may intersect the segment 5-3 instead. A test must be made to see which situation exists.

A better estimate of x_m, x_n, and x_q is found by replacing A_0^-, ζ_0, and A_0^+ by $(A_0^- + A_m^-)/2$, $(\zeta_0 + \zeta_n)/2$, and $(A_0^+ + A_q^+)/2$, respectively. The process is repeated until the quantities are determined to the desired degree of accuracy at the points m, n, and q.

By using the characteristics through the points m and q, we obtain an estimate for β_0 and ζ_0 from the compatibility conditions along these Mach lines. Thus

$$E_m(\zeta_0 - \zeta_m) + F_m(\beta_0 - \beta_m) + G_m(s_0 - s_m) - H_m^-(x_1 - x_m)/y_m = 0$$

$$E_q(\zeta_0 - \zeta_q) - F_q(\beta_0 - \beta_q) - G_q(s_0 - s_q) + H_q^+(x_1 - x_q)/y_q = 0.$$

Solving for β_0, ζ_0 leads to

$$\zeta_0 = (N_1F_q + N_2F_m)/(E_mF_q + E_qF_m) \tag{33}$$

$$\beta_0 = (N_1 - E_m\zeta_0)/F_m \tag{34}$$

where

$$N_1 = E_m\zeta_m + F_m\beta_m + G_m(s_m - s_0) + H_m^-(x_1 - x_m)/y_m \tag{35}$$

$$N_2 = E_q\zeta_q - F_q\beta_q - G_q(s_q - s_0) - H_q^+(x_1 - x_q)/y_q. \tag{36}$$

Similarly, these values of β_0 and ζ_0 can be improved by replacing, for example, E_m by $(E_m + E_0)/2$ and F_q by $(F_q + F_0)/2$ in Eqs. (33) and (34).

With β_0, ζ_0 estimated we now seek an approximation to ξ, the slope of the shock at the new point. Since β_1, ζ_1 and s_1 are functions of ξ and of the quantities ahead of the shock and must also satisfy the compatibility relation along the Mach line 2-1, ξ is then a root of the equation

$$F(\xi) = E_2(\zeta_1 - \zeta_2) - F_2(\beta_1 - \beta_2) - G_2(s_1 - s_2) + H_2^+(x_1 - x_2)/y_2 = 0.$$

The shock slope ξ then is given by applying Newton's method with ξ_s as a starting value. Thus

$$\xi = \xi - F(\xi)/F'(\xi), \tag{37}$$

where

$$F'(\xi) = E_2\zeta_1' - F_2\beta_1' - G_2s_1'. \tag{38}$$

The function $F(\xi)$ and $F'(\xi)$ are found by computing sequentially Eqs. (15) through (21) and in addition, the quantities

$$\sigma_0' = 2\sigma_0(1 + \zeta_0\xi)/(\xi - \zeta_0)(1 + \xi^2) \tag{39}$$

$$\sigma_1' = -2\sigma_0'/(\gamma + 1)\sigma_0^2 \tag{40}$$

$$\tau' = \sigma_1'[1/\sigma_1 + (k^2 - 1)/(\sigma_1 - k)(1 - k\sigma_1)]\tau \tag{41}$$

$$\begin{aligned}\zeta_1' = \zeta_1\{[(1 - \sigma_1) + 2\xi\zeta_0 + \sigma_1'(\zeta_0 - \xi)]/[\xi(1 + \zeta_0\xi)\\ + \sigma_1(\zeta_0 - \xi)] - [\zeta_0(1 - \sigma_1) + 2\sigma_1\xi + \xi(\xi - \zeta_0)\sigma_1']/[1\\ + \xi\zeta_0 + \sigma_1\xi(\xi - \zeta_0)]\end{aligned} \tag{42}$$

$$\begin{aligned}a' = 2a[\zeta_0/(1 + \zeta_0\xi) + (\zeta_1 - \xi)\zeta_1'/(1 + \zeta_1^2)(1 + \zeta_1\xi)\\ - \zeta_1/(1 + \zeta_1\xi)]\end{aligned} \tag{43}$$

$$\beta_1' = (a'/a - \tau'/\tau)(\beta_1^2 + 1)/2\beta_1 \tag{44}$$

$$s_1' = (\gamma - 1)\sigma_1'/\sigma_1 + \tau'/\tau. \tag{45}$$

The form of the derivatives used here are obtained by differentiating the logarithms of the quantities.

A better approximation to x_1, y_1 is now obtained by replacing A_2^+ and ξ_s in Eqs. (27) and (28) by

$$(A_2^+ + A_1^+)/2$$

and

$$(\xi_s + \xi)/2,$$

respectively. The remaining calculations are repeated with E_2 replaced by

$$(E_2 + E_1)/2$$

and similarly for A_2^+, G_2, and H_2^+. The iteration is continued until the desired accuracy is attained.

For the sake of clarity we summarize the computations below:

I. Locate new point on shock.

$$x_1 = [y_2 - y_s + \xi_s x_s - A_2^+ x_2]/(\xi_s - A_2^+)$$

$$y_1 = y_2 + A_2^+(x_2 - x_1)$$

II. Locate intersection of streamline and the two upstream Mach lines from the new point 1.

$$x_m = (y_1 - y_3 + D_1 x_1 - A_0^- x_1)/(D_1 - A_0^-)$$

$$x_n = (y_1 - y_3 + D_1 x_3 - \zeta_0 x_1)/(D_1 - \zeta_0)$$

$$x_q = (y_1 - y_3 + D_2 x_3 - A_0^+ x_1)/(D_2 - A_0^+)$$

where $D_1 = (y_3 - y_4)/(x_3 - x_4)$ and $D_2 = (y_5 - y_3)/(x_5 - x_3)$

III. Data at points m, n, q are interpolated by relations of the form

$$\zeta_m = [\zeta_3(x_m - x_4) + \zeta_4(x_3 - x_m)]/(x_3 - x_4)$$

and x_m, x_n, and x_q improved by repeating step II after replacing

$$A_0^- \text{ by } (A_0^- + A_m^-)/2$$

and similarly for A_0^+ and ζ_0.

IV. Data β_0, ζ_0, s_0 estimated by using the compatibility relations along streamline 1-n and along Mach lines 1-m and 1-q. Thus

$$\zeta_0 = (N_1F_q + N_2F_m)/(E_mF_q + E_qF_m)$$

$$\beta_0 = (N_1 - E_m\zeta_0)/F_m$$

$$s_0 = s_n = [s_3(x_n - x_4) + s_4(x_3 - x_n)]/(x_3 - x_4)$$

where

$$N_1 = E_m\zeta_m + F_m\beta_m + G_m(s_m - s_0) + H_m^-(x_1 - x_m)/y_m$$

$$N_2 = E_q\zeta_q - F_q\beta_q - G_q(s_q - s_0) - H_q^+(x_1 - x_q)/y_q$$

values of β_0, ζ_0, s_0 are improved by replacing

$$F_q \text{ by } (F_q + F_0)/2, \text{ etc.,}$$

in the preceding relations and repeating step IV.

V. New value of ξ is estimated by iterating

$$\xi = \xi - F(\xi)/F'(\xi)$$

where

$$F(\xi) = E_2(\zeta_1 - \zeta_2) - F_2(\beta_1 - \beta_2) - G_2(s_1 - s_2) + H_2^+(x_1 - x_2)/y_2$$

is the compatibility conditions along Mach line 2-1. The quantities

ζ_1, β_1 and s_1 are given as functions of ξ by Eq. (15) through (21) and their derivatives by Eq. (39) through (45). As a by-product of the iteration, β_1, ζ_1, and s_1 are also obtained.

VI. Steps I to IV are repeated with

$$A_2^+ \text{ replaced by } (A_2^+ + A_1^+)/2$$

and

$$\xi_s \text{ replaced by } (\xi_s + \xi)/2.$$

The calculations described above are coded in Fortran as subroutine SHOCK and are listed on pages 31 through 33 in the Appendix I.

IV. PROCEDURE FOR SELECTING DATA FROM THE UPSTREAM MACH NET FOR COMPUTING THE NEXT POINT ALONG THE SHOCK

In the preceding section, we showed how a new point is computed along a shock using data at three points of the upstream Mach net, at the preceding shock point, and at a point downstream of the new shock point. We now explain one method for selecting the appropriate points in the upstream Mach net for each point along the shock. Consider the upstream Mach net illustrated in Fig. 3. In this example, there are upward to the left running Mach lines containing 5 points. Each of these Mach lines is approximated by a parabola

$$y = ax^2 + bx + c \tag{46}$$

which passes through points 1, 3, and 5 of the Mach line (see Fig. 3). The coefficients a, b, and c are given by is positive or negative. When this quantity is negative then x_0 lies between the two Mach lines and we then test the value of y at $x = x_{t1}$ on the first Mach line, denoted by y_{t1}, to find which two of the five points lie on each side of it. That is, we investigate

$$(y_{t1} - y_j)(y_{t1} - y_{j+1}), \quad i=1,2,3,4$$

to see if it is positive or negative. If it is negative for particular j, then we have located the quadrilateral region in which the estimated point x_0, y_0 of the shock lies. From this quadrilateral we select the points 3, 4, and 5 of Fig. 2 to be used in the subroutine SHOCK described in the preceding section. If Eq. (48) should be positive then we try the second and third Mach lines and continue until the proper lines are found.

This procedure was programmed to compute the flow over a wedge or conical segment and a Fortran listing is given on page 30 of Appendix I. As a two dimensional example, the flow field was computed for a wedge inserted in a diverging channel in which the flow velocity was assumed to be radial and the Mach number constant along arcs of circles (two dimensional source flow). The upstream Mach net and the shock wave are shown in Fig. 4. The shock, downstream Mach net, and wedge surface are shown in Fig. 5. Figure 6 presents β along the wedge and ξ as functions of x. Figures 7 and 8 are graphs of β and ξ ahead of and behind the shock.

$$a = [y_1(x_3 - x_5) + y_3(x_5 - x_1) + y_5(x_1 - x_3)]/d$$

$$b = [y_1(x_5^2 - x_3^2) + y_3(x_1^2 - x_5^2) + y_5(x_3^2 - x_1^2)]/d$$

$$c = y_1 - ax_1^2 - bx_1$$

where $$d = x_1^2(x_3 - x_5) + x_3^2(x_5 - x_1) + x_5^2(x_1 - x_3).$$

We now find the intersection of the Mach line

$$y = y_0 + A_0^-(x - x_0)$$

with the parabola of Eq. (46). Equating values of y and solving for x the resulting quadratic

$$ax^2 + (b - A_0^-)x + (c - y_0 + A_0^- x_0) = 0$$

yields

$$x = (A_0^- - b)/2a \pm Q$$

where

$$Q = \sqrt{(A_0^- - b)^2/4a^2 - (c - y_0 + A_0^- x_0)/a}.$$

The sign of radical must be chosen to make the smallest positive value for x. If $(A_0^- - b)/a > 0$, then we select the minus sign in Eq. (47) when x is positive. If $(A_0^- - b)/a < 0$, then we use the positive sign. Let this value of x be denoted by x_{t1}, then we follow the same procedure for the adjacent set of 5 points. Denoting this value of x by x_{t2}, we check to see if

$$(x_{t1} - x_0)(x_{t2} - x_0) \tag{48}$$

As an axially-symmetric example, the flow induced by a change in the cone angle for the flow through a coaxial diverging channel was calculated. The flow in the upstream portion was assumed to be radial and the Mach number constant along spherical surfaces (spherical source flow). The upstream Mach net together with the shock wave and the cone boundary is shown in Fig. 9, while the downstream Mach net is presented in Fig. 10. Figures 11, 12, and 13 show the values of β along the cone and of β and ζ ahead of and behind the shock and of ξ as functions of x.

In each step of the calculation of the downstream Mach net for both examples, an additional point was interpolated between the boundary point and the adjacent point along the Mach line. The halfway point was chosen and data was found by averaging the data at the two adjacent points. For all Mach lines behind the shock except the first one a better approximation of the data at this point could be obtained by fitting a parabola to the three adjacent points along the Mach line at the boundary.

V. PROCEDURE FOR USING THE PROGRAM IN APPENDIX I TO COMPUTE A SHOCK

The examples of a wedge and cone were chosen to check and to illustrate the method of computing the shock and the associated downstream flow field. However, by using for the FORTRAN statements numbered 20 in WALL(X,K) and WALPR(X,K) on page 36 of Ref. [1] the coding for the y coordinate and its derivative for any chosen contour, the flows over more general shaped bodies may also be calculated using this program.

Besides the coding of the body shape, the appropriate parameters for the flow must be loaded by two data cards according to the Format

(3I3,3E15.8/E15.8,I3).

The data are as follows:

1. N — The number of downward to the right running characteristics in upstream Mach net.
2. KODE — The flow code according to Table I, page 4, of Ref. [1]. Since these flows have variable entropy, this is either 1 or 3 depending upon whether the flow is axially symmetric or two dimensional.
3. M — The number of upward to the right running Mach lines in the upstream Mach net data (5 in examples of Figs. 4 and 9).
4. C — Maximum number of iterations allowed.
5. EPS — Maximum fractional error (not percentage as in the program of Ref. [1]).
6. Gamma — Ratio of specific heats, γ
7. XI — Tangent ξ of shock angle at initial shock point.
8. NK — Number of points in shock to be computed.

The Mach net data points are punched according to the Format on page 6 and 7 of Ref. [1]. The data are read beginning with the upstream data at the first shock point which is also the lowest point of the first downward to the right running characteristic. The matrix of data is read upward along each of these Mach lines beginning at the lowest Mach line moving toward the top. The second from the last pair of data cards is the data just downstream of the initial shock point and the last pair of cards, the first data point on the boundary downstream of initial shock points. In the examples given, the data was conveniently chosen as the same as the beginning of the shock.

VI. CALCULATION OF FREE STREAMLINE BOUNDING TWO SUPERSONIC STREAMS

Assume that at one point of the free streamline the values of β and s on both sides of the streamline, the value of the streamline slope ζ, and the coordinates of the point (see Fig. 14) are known. Assume that the values of β, ζ, s, x and y are known at two points (3 and 4) along an upward to the right running characteristic above the streamline and at one point (2) below the streamline. We then seek the coordinates and the flow data of the next point along the streamline. Let 0 and 1 denote the quantities above and below the free streamline at the new point. The first estimate of the coordinate x_0, y_0 is given by the intersection of the Mach line from the point 2 with the linear extension of the streamline. This yields

$$x_0 = [y_2 - y_3 + \zeta_3 x_3 - A_2 x_2]/[\zeta_3 - A_2^+]$$

$$y_0 = y_2 + A_2^+(x_0 - x_2). \qquad (49)$$

Then using $\zeta_0 = \zeta_3$ and $\beta_0 = \beta_u$, where u denotes the upper value at point 3, the negative characteristic from x_0, y_0 is computed and x_m, y_m, β_m, ζ_m and s_m are interpolated in the same way as in the shock program (Eqs. (29) and (30)).

With x_m, y_m, ζ_m, β_m and s_m known, then the compatibility condition along the characteristic or Mach line yields one relation in β_0, ζ_0 or ζ_1. Thus

$$E_m(\zeta_1 - \zeta_m) + F_m(\beta_0 - \beta_m) + G_m(s_u - s_m) - H_m^-(x_0 - x_m)/y_m = 0$$

since $\zeta_1 = \zeta_0$ and $s_0 = s_u$. The compatibility condition along the characteristics connecting the points 1 and 2 provides a relation in ζ_1 and β_1, namely,

$$E_2(\zeta_1 - \zeta_2) - F_2(\beta_1 - \beta_2) - G_2(s_\ell - s_2) + H_2^+(x_0 - x_2)/y_2.$$

where ℓ denotes the lower value at point 3.

The remaining relation in the quantities at point 1 is supplied by the condition that the pressure be continuous across the free streamline. Along each streamline the pressure satisfies the following differential equation, since entropy is constant,

$$dp/p = -2\gamma\beta d\beta/[(\gamma - 1)\beta^2 + (\gamma + 1)] = N(\beta)d\beta.$$

This is derived from Eq. (7), page 87 of Ref. [2]. In difference form this relation leads to

$$(p_u - p_0)/p_u = N_u(\beta_u - \beta_0)$$

on the upper part of the streamline and

$$(p_\ell - p_1)/p_\ell = N_\ell(\beta_\ell - \beta_1),$$

on the lower side of the streamline. Since $p_\ell = p_u$ and $p_0 = p_1$ we must have

$$N_u(\beta_u - \beta_0) = N_\ell(\beta_\ell - \beta_1)$$

or

$$N_\ell\beta_1 - N_u\beta_0 = N_\ell\beta_\ell - N_u\beta_u = C_1. \tag{51}$$

Eliminating ζ_1 between the compatibility equations yields

$$E_2F_m\beta_0 + E_mF_2\beta_1 = C_2 \tag{52}$$

where

$$\begin{aligned} C_2 = {} & E_2[E_m\zeta_m + F_m\beta_m + G_m(s_m - s_u) + H_m^-(x_0 - x_m)/y_m] \\ & - E_m[E_2\zeta_2 - F_2\beta_2 - G_2(s_2 - s_\ell) + H_2^-(x_2 - x_0)/y_2]. \end{aligned} \tag{53}$$

Equations (51) and (52) solved for β_1 and β_0 yield

$$\beta_1 = (E_2F_mC_1 + N_uC_2)/(E_2F_mN_\ell + E_mF_2N_u] \tag{54}$$

and

$$\beta_0 = (N_\ell\beta_1 - C_1)/N_u. \tag{55}$$

Then

$$\zeta_1 = [E_2\zeta_2 + F_2(\beta_1 - \beta_2) + G_2(s_\ell - s_2) - H_2^+(x_0 - x_2)/y_2]/E_2. \tag{56}$$

The values of x_0, y_0, β_1, ζ_1 and β_0 are improved by repeating the calculations with the coefficients replaced by average values, such as

$$F_2 \text{ replaced by } (F_1 + F_2)/2$$

$$F_m \text{ replaced by } (F_m + F_0)/2, \text{ etc.}$$

The iteration is carried out to the desired degree of accuracy.

For the sake of clarity, the procedure for computing the next point along a free streamline is summarized below.

I. Estimate new point by linear extension of Mach line and streamline.

$$x_0 = (y_2 - y_3 + \zeta_3 x_3 - A_2^+ x_2)/(\zeta_3 - A_2^+)$$

$$y_0 = y_2 + A_2^+(x_0 - x_2)$$

II. Interpolate point m using steps similar to II, III, and IV for the SHOCK subroutine.

III. Combine compatibility relations along the two characteristics through the point x_0, y_0, with the condition of continuous pressure and solve for β_0, ζ_0 and β_1.

$$\beta_1 = (E_2 F_m C_1 + N_u C_2)/(E_2 F_m N_\ell + E_m F_2 N_u)$$

$$\beta_0 = (N_\ell \beta_1 - C_1)/N_u$$

$$\zeta_0 = \zeta_1 = [E_2 \zeta_2 + F_2(\beta_1 - \beta_2) + G_2(s \ \ - s_2) - H_2^+(x_0 - x_2)/y_2]$$

where C_1 and C_2 are given by Eqs. (51) and (53).

IV. With the estimates of β_0, β_1, ζ_0, repeat steps I, II, and III with

$$\zeta_3 \text{ replaced by } (\zeta_3 + \zeta_0)/2$$

$$A_2^+ \text{ replaced by } (A_2^+ + A_1^+)/2$$

$$F_m \text{ replaced by } (F_m + F_0)/2$$

$$N_u \text{ replaced by } (N_u + N_0)/2$$

and similarly for the other coefficients.

This procedure is coded as subroutine STREAM and the FORTRAN listing is given on page 41 of Appendix II.

VI. PROGRAM FOR COMPUTING A JET EXHAUSTING INTO A SUPERSONIC STREAM

The complete program described in Ref. [1] was modified to make it capable of computing a jet issuing into a supersonic outside stream. The only subroutines which were changed are READ, which, besides reading the usual initial data for the jet stream, must include the data along a Mach line in the outer stream; WRITE2(K), which is modified to include the printing of the data along each newly computed exterior Mach line; and FREE(M) which computes each new streamline point using the subroutine STREAM and computes also the next exterior Mach line. The modified subroutines are listed in the appendix and may be used in place of the originals to calculate the other flows described in Ref. [1] as well.

The subroutine FREE(M) selects the points 3 and 4 to be used in the STREAM subroutine in a manner similar to the procedure for the shock program. Let the subscript 0 denote the quantities at the new streamline point and x_j, y_j, i=1,2,3,...,N the coordinates of the points on the exterior Mach line. The x coordinate of the intersection of the straight line through the two points x_j, y_j and x_{j+1}, y_{j+1} with the upward to the left running Mach line from the new streamline point is

$$x = (y_0 - y_j + Dx_j - A_0^- x_0)/(D - A_0^-)$$

where

$$D = (y_j - y_{j+1})/(x_j - x_{j+1}).$$

When

$$(x - x_j)(x - x_{j+1})$$

is negative, then x lies between x_j and x_{j+1}. When it is positive, the program continues and tests similarly the points x_{j+1}, y_{j+1} and x_{j+2}, y_{j+2}. Once the values of x_0, y_0, β_1, ζ_1, s_1, β_0, s_0 are computed the upward to the right running Mach line is computed using the values x_0, y_0, β_0, ζ_0, s_0 and the upward data point 4 of Fig. 14 to start the step by step calculations through the Mach line. This is seen from the example shown in Fig. 15 for an axially-symmetric jet. In this example both exterior and interior initial data were derived by assuming a source flow in which the Mach number is constant along spherical surfaces and the flow direction radial. The flow could not be continued further without introducing a shock wave because of the coalescing of the Mach lines in the jet.

VII. PROCEDURE FOR USING THE PROGRAM IN APPENDIX II TO COMPUTE A JET BOUNDARY IN A SUPERSONIC EXTERIOR G FLOW

With the subroutines in Appendix II replacing the correspondingly named programs in Ref. [1] the program is capable of calculating all flows described in Ref. [1] except a free streamline boundary in quiescent air. The first two data cards follow the order of the data on pages 5 and 6 of Ref. [1]. For code IU=3, a third card is added according to FORMAT(E10.0). This contains the data CN, the number of points in the initial line in the jet plus the number of data points in the initial Mach line in the exterior flow with the streamline point being considered as two points.

The data points for the initial line in the jet are loaded in the same way as described in Ref. [1]. The exterior initial data is loaded beginning with the point farthest away from the streamline boundary. The final pair of cards will be the data for the exterior flow at the initial streamline point.

REFERENCES

[1] Ehlers, F. Edward. An IBM Program for Computing Two-Dimensional and Axially-Symmetric Supersonic Flow of an Ideal Gas. Boeing document D1-82-0204, Seattle, Washington, November 1962.

[2] Ehlers, F. Edward. The Method of Characteristics for Isoenergetic Supersonic Flows Adapted to High-Speed Digital Computers. *J. Soc. Indust. Appl. Math.*, Vol. 7, p. 85, March 1959.

APPENDIX I

Listing of Program for Computing Shock in Non-Uniform Flow

```
C      MAIN CONTROL PROGRAM FOR SHOCK IN NON-UNIFORM FLOW
       DIMENSION X(1000),Y(1000),BETA(1000),ZETA(1000),S(1000)



       COMMON N,CODE,IL,IU,C,EPS,GAMMA,X,Y,BETA,ZETA,S,BETA0,XI,STOP,
      1 MXSTP
     1 FORMAT(3I3,3E15.8/E15.8,I3)
     2 FORMAT(4E15.8/E15.8)
     4 FORMAT (1X,5E15.8)
     5 FORMAT (1X,3I5)
C
C
C      FUNCTIONS
C
       APLMIF(A,B,M)=(A*B+(-1.)**M)/(A-(-1.)**M*B)
C
C
     3 READ  INPUT TAPE 5,1,N,KODE,M,C,EPS,GAMMA,XI,NK
       CODE=KODE
       K=1
       N1=M*N
       J4=N1+1
       N2=N1+2
       READ  INPUT TAPE 5,2,(X(I),Y(I),BETA(I),ZETA(I),S(I),I=1,N2)
       BETAZ=BETA(1)
       ZETAZ=ZETA(1)
       SZ=S(1)
       DO 140 II=1,NK
       A2=APLMIF(BETA(J4+1),ZETA(J4+1),0)
       XZ=(Y(J4+1)-Y(J4)+XI*X(J4)-A2*X(J4+1))/(XI-A2)
       YZ=Y(J4+1)+A2*(XZ-X(J4+1))
       APZ=APLMIF(BETAZ,ZETAZ,0)
       IFT=1
       DO 60 J=1,N
       J1=1+(J-1)*M
       J2=(M+1)/2+(J-1)*M
       J3=J*M
       C1=(X(J2)-X(J3))*Y(J1)+(X(J3)-X(J1))*Y(J2)+(X(J1)-X(J2))*Y(J3)
       D1=(X(J2)-X(J3))*X(J1)**2+(X(J3)-X(J1))*X(J2)**2+
      1(X(J1)-X(J2))*X(J3)**2
       C2=(X(J3)**2-X(J2)**2)*Y(J1)+(X(J1)**2-X(J3)**2)*Y(J2)+
      1(X(J2)**2-X(J1)**2)*Y(J3)
       C3=C1/D1
       C4=C2/D1
       C5=Y(J1)-C3*X(J1)**2-C4*X(J1)
       C6=C4-APZ
       C7=C5-YZ+APZ*XZ
       C8=SQRTF((C6/C3)**2-4.*C7/C3)
       XP=(-C6/C3+C8)/2.
       XM=XP-C8
       IF(XM)10,10,20
    10 XT=XP
       GO TO 30
    20 XT=XM
```

```
   30 GO TO (40,50),IFT
   40 IFT=2
      GO TO 60
   50 TEST=D*(XT-XZ)
      IF(TEST)70,70,60
   60 D=XT-XZ
      WRITE OUTPUT TAPE 6,1000
 1000 FORMAT(28HSHOCK POINT OUTSIDE MACH NET)
      CALL EXIT
   70 XT=D+XZ
      YT=YZ+APZ*D
      DO 90 I=1,M
      J5=J1-M+I
      TEST=(YT-Y(J5-1))*(YT-Y(J5))
      IF(TEST)80,90,90
   80 J6=J5-1
      J7=J5+M
      J8=J5
      WRITE OUTPUT TAPE 6,5,J6,J7,J8
      GO TO 100
   90 CONTINUE
      WRITE OUTPUT TAPE 6,1000
      CALL EXIT
  100 XIS=XI
      BS=BETAZ
      ZS=ZETAZ
      SS=SZ
      CALL SHOCK(X(J4),Y(J4),BS,ZS,SS,XIS,X(J6),Y(J6),BETA(J6),ZETA(J6),
     1S(J6),X(J7),Y(J7),BETA(J7),ZETA(J7),S(J7),X(J8),Y(J8),BETA(J8),ZET
     2A(J8),S(J8),X(J4+1),Y(J4+1),BETA(J4+1),ZETA(J4+1),S(J4+1),X(J4),Y(
     3J4),BETA(J4),ZETA(J4),S(J4),XI,BETAZ,ZETAZ,SZ)
      WRITE OUTPUT TAPE 6,4,XI,BETAZ,ZETAZ,SZ
      N3=II-1
      IF(N3)130,130,110
  110 DO 120 JJ=1,N3
      J9=J4+JJ
  120 CALL ITER1(J9+1,J9-1,J9)
  130 MXSTP=J4+II
      CALL FIXED(K)
      J10=J4+II
      J11=J10+1
      X(J11)=X(J10)
      Y(J11)=Y(J10)
      BETA(J11)=BETA(J10)
      ZETA(J11)=ZETA(J10)
      S(J11)=S(J10)
      X(J10)=(X(J10)+X(J10-1))/2.
      Y(J10)=(Y(J10)+Y(J10-1))/2.
      BETA(J10)=(BETA(J10)+BETA(J10-1))/2.
      ZETA(J10)=(ZETA(J10)+ZETA(J10-1))/2.
      S(J10)=(S(J10)+S(J10-1))/2.
  140 WRITE OUTPUT TAPE 6,4,(X(I),Y(I),BETA(I),ZETA(I),S(I),I=J4,J11)
      GO TO 3
      END
```

```
      SUBROUTINE SHOCK (XS,YS,BB,ZB,SB,XIS,X5,Y5,B5,Z5,S5,X4,
     1  Y4,B4,Z4,S4,X3,Y3,B3,Z3,S3,X2,Y2,B2,Z2,S2,X1,Y1,B1,Z1,S1,
     2  XI,BZ,ZZ,SZ)
      COMMON N,CODE,IL,IU,C,EPS,GAMMA,X,Y,BETA,ZETA,S,BETAO,XI,STOP,
     1 MXSTP
C
C
C     FUNCTIONS
      APLMIF(A,B,M)=(A*B+(-1.)**M)/(A-(-1.)**M*B)
      E1E2F(A)=1./(A**2+1.)
      F1F2F(A)=2.*A**2/((A**2+1.)*((GAMMA-1.)*A**2+GAMMA+1.))
      G1G2F(A)=A/(GAMMA*(GAMMA-1.)*(A**2+1.))
      HPLMIF(A,B,M)=B/(A-(-1.)**M*B)
C
C
      ZZ=ZB
      BZ=BB
      XI=XIS
      XIT=XI
      BZT=BB
      ZZT=ZB
      ZM=Z3
      XT=XS
      YT=YS
      BT=BB
      ST=S2
      ZT=ZB
      M=C
      XT1=+0.1E+07
      XT2=+0.1E+07
      XT3=+0.1E+07
      A2=APLMIF(B2,Z2,0)
      SA2=A2
      E2=E1E2F(Z2)
      SE2=E2
      F2=F1F2F(B2)
      SF2=F2
      G2=G1G2F(B2)
      SG2=G2
      IF (CODE-3.) 3,2,2
```

```
    2 H2=0.
      GO TO 4
    3 H2=HPLMIF(B2,Z2,0)
      SH2=H2
    4 FG=GAMMA-1.
      FK=FG/(GAMMA+1.)
      DO 150 I=1,M
      X1=(Y2-YS+(XI+XIS)*XS/2.-A2*X2)/((XI+XIS)/2.-A2)
      Y1=Y2+A2*(X1-X2)
      AZ=APLMIF(BZ,ZZ,1)
      SAZ=AZ
      D=(Y3-Y4)/(X3-X4)
      XTM=Y1-Y3+D*X3
      DO 20 J=1,M
      XM=(XTM-AZ*X1)/(D-AZ)
      ZM  =(Z3*(XM-X4)+Z4*(X3-XM))/(X3-X4)
      BM=(B3*(XM-X4)+B4*(X3-XM))/(X3-X4)
      IF (ABSF((XT1-XM)/XT1)-EPS)30,10,10
   10 XT1=XM
   20 AZ=(SAZ+APLMIF(BM,ZM,1))/2.
      PRINT 1
    1 FORMAT(28HITERATION 1 DID NOT CONVERGE   )
   30 SM=(S3*(XM-X4)+S4*(X3-XM))/(X3-X4)
      YM=Y1+AZ*(XM-X1)
      ZN=ZZ
      DO 40 K=1,M
      XN=(XTM-(ZN+ZZ)*X1/2.)/(D-(ZZ+ZN)/2.)
      ZN  =(Z4*(XN-X3)+Z3*(X4-XN))/(X4-X3)
      IF (ABSF((XT2-XN)/XT2)-EPS) 50,40,40
   40 XT2=XN
      PRINT 2000
 2000 FORMAT(28HITERATION 2 DID NOT CONVERGE   )
   50 SZ=(S4*(XN-X3)+S3*(X4-XN))/(X4-X3)
      D=(Y5-Y3)/(X5-X3)
      IF ((XN-X3)*(XN-X4)) 330,330,300
  300 ZN=ZZ
      XTM=Y1-Y3+D*X3
      DO 310 J=1,M
      XN=(XTM-(ZN+ZZ)*X1/2.)/(D-(ZZ+ZN)/2.)
      ZN=(Z5*(XN-X3)+Z3*(X5-XN))/(X5-X3)
```

```
      IF (ABSF((XT2-XN)/XT2)-EPS) 320,310,310
 310  XT2=XN
 320  SZ=(S5*(XN-X3)+S3*(X5-XN))/(X5-X3)
 330  AP=APLMIF(BZ,ZZ,0)
      SAP=AP
      DO 210 J=1,M
      XQ=(Y1-Y3+D*X3-AP*X1)/(D-AP)
      IF (ABSF((XT3-XQ)/XT3)-EPS) 220,200,200
 200  XT3=XQ
      ZQ=(Z3*(XQ-X5)+Z5*(X3-XQ))/(X3-X5)
      BQ=(B3*(XQ-X5)+B5*(X3-XQ))/(X3-X5)
 210  AP=(SAP+APLMIF(BQ,ZQ,0))/2.
      PRINT 3000
3000  FORMAT(28HITERATION 3 DID NOT CONVERGE   )
 220  SQ=(S3*(XQ-X5)+S5*(X3-XQ))/(X3-X5)
      YQ=(Y3*(XQ-X5)+Y5*(X3-XQ))/(X3-X5)
      EM=E1E2F(ZM)
      FM=F1F2F(BM)
      GM=G1G2F(BM)
      EQ=E1E2F(ZQ)
      FQ=F1F2F(BQ)
      GQ=G1G2F(BQ)
      IF (CODE-3.) 52,51,51
  51  HM=0.
      HZ=0.
      HQ=0.
      HPZ=0.
      GO TO 53
  52  HM=HPLMIF(BM,ZM,1)
      HZ=HPLMIF(BZ,ZZ,1)
      HQ=HPLMIF(BQ,ZQ,0)
      HPZ=HPLMIF(BZ,ZZ,0)
  53  EZ=E1E2F(ZZ)
      FZ=F1F2F(BZ)
      GZ=G1G2F(BZ)
      FN1=(EM+EZ)*ZM+(FM+FZ)*BM+(GM+GZ)*(SM-SZ)+(HM/YM+HZ/Y1)*(X1-XM)
      FN2=(EQ+EZ)*ZQ-(FQ+FZ)*BQ-(GQ+GZ)*(SQ-SZ)-(HQ/YQ+HPZ/Y1)*(X1-XQ)
      ZZ=(FN1*(FQ+FZ)+FN2*(FM+FZ))/((EM+EZ)*(FQ+FZ)+(EQ+EZ)*(FM+FZ))
      BZ=(FN1-(EM+EZ)*ZZ)/(FM+FZ)
      DO 80 L=1,M
```

```
      SIZ=(BZ**2+1.)*((XI-ZZ)**2)/((1.+XI**2)*(1.+ZZ**2))
      SI1=2./((GAMMA+1.)*SIZ)+FK
      TAU=SI1*(1.-FK*SI1)/(SI1-FK)
      Z1=(XI*(1.+ZZ*XI)-SI1*(XI-ZZ))/(1.+ZZ*XI+SI1*XI*(XI-ZZ))
      A1=((1.+ZZ*XI)**2)*(1.+Z1**2)/((1.+ZZ**2)*(1.+Z1*XI)**2)
      B1=SQRTF(-1.+A1*(BZ**2+1.)/TAU)
      S1=SZ+FG*LOGF(SI1)+LOGF(TAU)
      SIZP=2.*SIZ*(1.+ZZ*XI)/((XI-ZZ)*(1.+XI**2))
      SI1P=-2.*SIZP/((GAMMA+1.)*SIZ**2)
      TAUP=SI1P*(1./SI1+(FK**2-1.)/((SI1-FK)*(1.-FK*SI1)))*TAU
      Z1P=Z1*((1.-SI1+2.*XI*ZZ+SI1P*(ZZ-XI))/(XI*(1.+XI*ZZ)+SI1*(ZZ-
     1 XI))-(ZZ*(1.-SI1)+2.*SI1*XI+XI*(XI-ZZ)*SI1P)/(1.+ZZ*XI+SI1*
     2 XI*(XI-ZZ)))
      AP=2.*A1*(ZZ/(1.+ZZ*XI)+(Z1-XI)*Z1P/((1.+Z1**2)*(1.+Z1*XI))-Z1/(1.
     1 +Z1*XI))
      B1P=(AP/A1-TAUP/TAU)*(B1**2+1.)/(2.*B1)
      S1P=FG*SI1P/SI1+TAUP/TAU
      FXI=E2*(Z1-Z2)-F2*(B1-B2)-G2*(S1-S2)+H2*(X1-X2)/Y2
      FXIP=E2*Z1P-F2*B1P-G2*S1P
      XIN=XI-FXI/FXIP
      IF(ABSF((XI-XIN)/XI)-EPS)90,80,80
   80 XI=XIN
      PRINT 4000
 4000 FORMAT(28HITERATION 4 DID NOT CONVERGE   )
   90 XI=XIN
      IF(ABSF((X1-XT)/XT)-EPS)100,140,140
  100 IF(ABSF((Y1-YT)/YT)-EPS)110,140,140
  110 IF(ABSF((Z1-ZT)/ZT)-EPS)120,140,140
  120 IF(ABSF((B1-BT)/BT)-EPS)130,140,140
  130 IF(ABSF((S1-ST)/ST)-EPS)131,140,140
  131 IF(ABSF((XI-XIT)/XIT)-EPS)132,140,140
  132 IF(ABSF((BZT-BZ)/BZT)-EPS)133,140,140
  133 IF(ABSF((ZZT-ZZ)/ZZT)-EPS)160,140,140
  140 E2=(SE2+E1E2F(Z1))/2.
      F2=(SF2+F1F2F(B1))/2.
      G2=(SG2+G1G2F(B1))/2.
      IF (CODE-3.) 141,142,142
  141 H2=(SH2+HPLMIF(B1,Z1,0)*Y2/Y1)/2.
  142 A2=(SA2+APLMIF(B1,Z1,0))/2.
      XT=X1


      YT=Y1
      ZT=Z1
      BT=B1
      XIT=XI
      BZT=BZ
      ZZT=ZZ
  150 ST=S1
      PRINT 5
    5 FORMAT(26HITERATION DID NOT CONVERGE   )
  160 RETURN
      END
```

Additional subroutines required by MAIN CONTROL PROGRAM FOR SHOCK IN NON-UNIFORM FLOW but not listed here

SUBROUTINE ITER (page 27, Ref. [1])

SUBROUTINE ITER (page 29, Ref. [1])

FIXED(K) The subroutine listed on page 35 of Ref. [1] was modified for use here by replacing instruction 20 and the one following with

```
20 J1 = MXSTP - 1
   J3 = MXSTP
```

FUNCTION WALL(X,K) (page 36, Ref. [1])

FUNCTION WALPR(X,K) (page 36, Ref. [1])

(Instruction 10 gives upper wall formula and instruction 20 for lower wall.)

APPENDIX II

Modified Subroutines of Program of Ref. [1] Used to Compute Free Streamline Boundary between Two Supersonic Streams

Subroutines READ, WRITE2(K), FREE(M) listed here replace correspondingly named subroutines in Ref. [1] and the subroutine STREAM is added.

```
      SUBROUTINE READ


      DIMENSION X(1000), Y(1000), BETA(1000), ZETA(1000), S(1000)
      COMMON N,CODE,IL,IU,A,EPS,GAMMA,X,Y,BETA,ZETA,S,BETAO,XI,STOP
     1,MXSTP,NF
    1 FORMAT(6I3,4E13.0)
    2 FORMAT(4E15.8/E15.8)
    3 FORMAT(E10.0)
      READ INPUT TAPE 5,1,M,ICODE,IU,IL,MAX,MXSTP,EPS,GAMMA,FREEM,ANGLE
      READ INPUT TAPE 5,3,STOP
      IF (IU-3) 210,200,210
  200 READ INPUT TAPE 5,3,CN
      NF=CN
  210 EPS = EPS/100.
      N=M-1
      CODE = ICODE
      A = MAX
      IF (GAMMA)10,10,15
   10 GAMMA = 1.4
   15 DO 100 I=1,N
  100 READ INPUT TAPE 5,2,X(I),Y(I),BETA(I),ZETA(I),S(I)
      N=M+1
      READ INPUT TAPE 5,2,X(N),Y(N),BETA(N),ZETA(N),S(N)
      IF (IU-3) 310,300,310
  300 NF=NF+1
      NB=N+1
      READ INPUT TAPE 5,2,(X(I),Y(I),BETA(I),ZETA(I),S(I),I=NB,NF)
  310 RETURN
      END
```

```
      SUBROUTINE WRITE2(K)
      DIMENSION X(1000), Y(1000), BETA(1000), ZETA(1000), S(1000)


      COMMON N,CODE,IL,IU,C,EPS,GAMMA,X,Y,BETA,ZETA,S,BETA0,XI,STOP,
     1 MXSTP,NF
      I = 1.5+.5*(-1.)**K
   10 WRITE OUTPUT TAPE 6, 1, K
      M=N-2
      DO 1000 J=1,M
      II=I+J-1
  750 L=J
 1000 WRITE OUTPUT TAPE 6,2,L,X(II),Y(II),BETA(II),ZETA(II),S(II)
      IF ((-1)**K) 2000,2000,2050
 2000 L=N-1
      WRITE OUTPUT TAPE 6,2,L,X(N),Y(N),BETA(N),ZETA(N),S(N)
      IF (IU-3) 210,200,210
  200 NA=N+1
      DO 100 J=NA,NF
      L=J-1
  100 WRITE OUTPUT TAPE 6,2, L,X(J),Y(J),BETA(J),ZETA(J),S(J)
  210 IF (IU-4) 2050,2020,2050
 2020 ANGLE = ATANF(XI)
      WRITE OUTPUT TAPE 6, 4, ANGLE
 2050 RETURN
    1 FORMAT(1H1/39X,42HCHARACTERISTIC SOLUTION OF SUPERSONIC FLOW,///54
     1X,8HSTEP NO.,I4////,16X,1HX,22X,1HY,18X,9HBETA      ,13X,11HZETA
     2     ,14X,7HENTROPY,/120X)
    2 FORMAT(120X/I5,5E23.8)
    4 FORMAT(120X/41H THE SHOCK ANGLE AT THE BOUNDARY POINT IS,E20.8)
      END
```

```
      SUBROUTINE FREE(M)


      DIMENSION X(1000), Y(1000), BETA(1000), ZETA(1000), S(1000)
      COMMON N,CODE,IL,IU,C,EPS,GAMMA,X,Y,BETA,ZETA,S,BETAO,XI,STOP,
     1 MXSTP,NF
C     FUNCTION
      APLMIF(A,B,K)=(A*B+(-1.)**K)/(A-(-1.)**K*B)
C
      IF(M)1000,10,1000
   10 AN=APLMIF(BETA(N-1),ZETA(N-1),0)
      XT=(Y(N-1)-Y(N)+ZETA(N)*X(N)-AN*X(N-1))/(ZETA(N)-AN)
      YT=Y(N)+ZETA(N)*(XT-X(N))
      A2=APLMIF(BETA(NF),ZETA(NF),1)
      NE=NF-N-1
      DO 20 J=1,NE
      J2=NF-J
      J1=J2+1
      SL=(Y(J1)-Y(J2))/(X(J1)-X(J2))
      X1=(YT-Y(J1)+SL*X(J1)-A2*XT)/(SL-A2)
      IF((X(J1)-X1)*(X(J2)-X1))30,30,20
   20 CONTINUE
      WRITE OUTPUT TAPE 6,2
      CALL EXIT
    2 FORMAT (23H1POINT OUTSIDE MACH NET)
   30 J3=J2
      J2=J3+1
      CALL STREAM(X(NF),Y(NF),BETA(NF),ZETA(NF),S(NF),
     1BETA(N),S(N),X(N-1),Y(N-1),BETA(N-1),ZETA(N-1),S(N-1),X(J2),
     2Y(J2),BETA(J2),ZETA(J2),S(J2),X(J3),Y(J3),BETA(J3),ZETA(J3),
     3S(J3),XZ,YZ,BZ,ZZ,SZ,X1,Y1,B1,Z1,S1)
      NF=J2
      X(NF)=X1
      Y(NF)=Y1
      BETA(NF)=BZ
      ZETA(NF)=ZZ
      S(NF)=SZ
      X(N)=X1
      Y(N)=Y1
      BETA(N)=B1
      ZETA(N)=Z1
      NE=NF-N-1
      DO 40 I=1,NE
      I2=NF-I
      I1=I2+1
   40 CALL ITER1(I1,I2,I2)
      RETURN
 1000 WRITE OUTPUT TAPE 6,1
      CALL EXIT
    1 FORMAT (33H1ERROR EXIT FROM FREE SUBROUTINE.)
      END
```

```
      SUBROUTINE STREAM(XU,YU,BU,ZU,SU,BL,SL,X2,Y2,B2,Z2,S2,X3,Y3,B3,Z3,
     1S3,X4,Y4,B4,Z4,S4,XZ,YZ,BZ,ZZ,SZ,X1,Y1,B1,Z1,S1)


      COMMON N,CODE,IL,IU,C,EPS,GAMMA,X,Y,BETA,ZETA,S,BETA0,XI,STOP,
     1 MXSTP,NF
C     FUNCTIONS
      APLMIF(A,B,M)=(A*B+(-1.)**M)/(A-(-1.)**M*B)
      E1E2F(A)=1./(A**2+1.)
      F1F2F(A)=2.*A**2/((A**2+1.)*((GAMMA-1.)*A**2+GAMMA+1.))
      FNFF(A)=2.*GAMMA*A/((GAMMA-1.)*A**2+GAMMA+1.)
      G1G2F(A)=A/(GAMMA*(GAMMA-1.)*(A**2+1.))
      HPLMIF(A,B,M)=B/(A-(-1.)**M*B)
C
      M=C
      BZ=BU
      Z1=ZU
      ZM=Z3
      XT=XU
      YT=YU
      BZT=BU
      B1T=BL
      Z1T=ZU
      XMT=X3
      A2=APLMIF(B2,Z2,0)
      SA2=A2
      E2=E1E2F(Z2)
      SE2=E2
      F2=F1F2F(B2)
      SF2=F2
      G2=G1G2F(B2)
      SG2=G2
      D=(Y3-Y4)/(X3-X4)
      IF (CODE-3.) 4,3,3
    3 H2=0.
      GO TO 5
    4 H2=HPLMIF(B2,Z2,0)
      SH2=H2
    5 FNL=FNFF(BL)
      SNL=FNL
      FNU=FNFF(BU)
      SNU=FNU
      DO 90 I=1,M
      X1=(Y2-YU+(Z1+ZU)*XU/2.-A2*X2)/((Z1+ZU)/2.-A2)
      Y1=Y2+A2*(X1-X2)
      AZ= APLMIF(BZ,Z1,1)
      SAZ=AZ
      XTM=Y1-Y3+D*X3
      DO 20 J=1,M
      XM=(XTM-AZ*X1)/(D-AZ)
      IF(ABSF((XM-XMT)/XMT)-EPS)30,10,10
   10 XMT=XM
      BM=(B3*(XM-X4)+B4*(X3-XM))/(X3-X4)
      ZM=(Z3*(XM-X4)+Z4*(X3-XM))/(X3-X4)
```

```
 20 AZ=(SAZ+APLMIF(BM,ZM,1))/2.
    WRITE OUTPUT TAPE 6,1
  1 FORMAT(28HITERATION 1 DID NOT CONVERGE   )
 30 SM=(S3*(XM-X4)+S4*(X3-XM))/(X3-X4)
    YM=Y1+AZ*(XM-X1)
    ZM=(Z3*(XM-X4)+Z4*(X3-XM))/(X3-X4)
    BM=(B3*(XM-X4)+B4*(X3-XM))/(X3-X4)
    EM=E1E2F(ZM)
    FM=F1F2F(BM)
    GM=G1G2F(BM)
    E1=E1E2F(Z1)
    FZ=F1F2F(BZ)
    GZ=G1G2F(BZ)
    IF (CODE-3.) 32,31,31
 31 HM=0.
    HZ=0.
    GO TO 33
 32 HM=HPLMIF(BM,ZM,1)
    HZ=HPLMIF(BZ,Z1,1)
 33 EN=(EM+E1)/2.
    FN=(FM+FZ)/2.
    GN=(GM+GZ)/2.
    HN=(HM+HZ*YM/Y1)/2.
    C1=FNL*BL-FNU*BU
    C2=E2*(EN*(ZM-Z2)+FN*BM+GN*(SM-SU)+HN*(X1-XM)/YM)
   1+EN*(F2*B2+G2*(S2-SL)+H2*(X1-X2)/Y2)
    B1=(E2*FN*C1+FNU*C2)/(E2*FN*FNL+EN*F2*FNU)
    Z1=(E2*Z2+F2*(B1-B2)+G2*(SL-S2)+H2*(X2-X1)/Y2)/E2
    BZ=(FNL*B1-C1)/FNU
    IF(ABSF((XT-X1)/XT)-EPS)40,80,80
 40 IF(ABSF((YT-Y1)/YT)-EPS)50,80,80
 50 IF(ABSF((BZT-BZ)/BZT)-EPS)60,80,80
 60 IF(ABSF((B1T-B1)/B1T)-EPS)70,80,80
 70 IF (Z1T) 71,80,71
 71 IF(ABSF((Z1T-Z1)/Z1T)-EPS)100,80,80
 80 XT=X1
    YT=Y1
    BZT=BZ
    B1T=B1
    Z1T=Z1
    A2=(SA2+APLMIF(B1,Z1,0))/2.
    E2=(SE2+E1E2F(Z1))/2.
    F2=(SF2+F1F2F(B1))/2.
    G2=(SG2+G1G2F(B1))/2.
    IF (CODE-3.) 81,82,82
 81 H2=(SH2+HPLMIF(B1,Z1,0)*Y2/Y1)/2.
 82 FNU=(SNU+FNFF(BZ))/2.
 90 FNL=(SNL+FNFF(B1))/2.
    WRITE OUTPUT TAPE 6,2
  2 FORMAT(26HITERATION DID NOT CONVERGE   )
100 S1=SL
    SZ=SU
    XZ=X1
    YZ=Y1
    RETURN
```

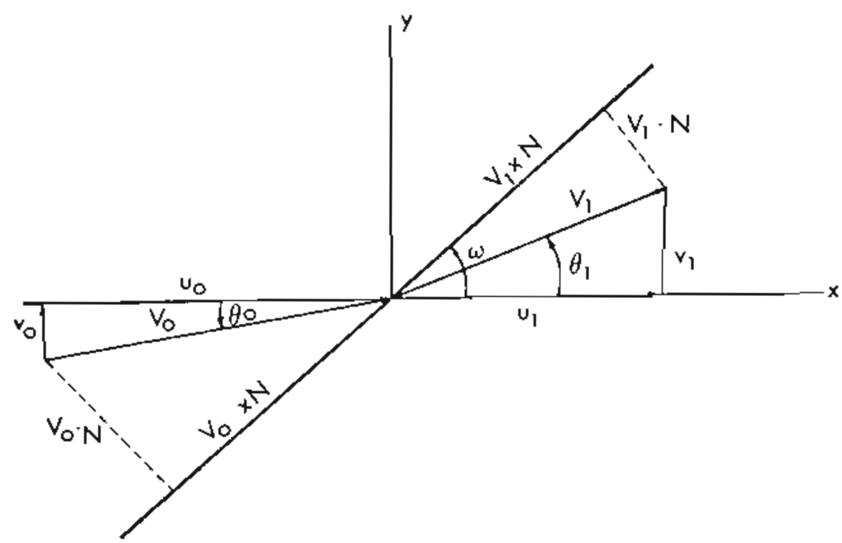

Fig. 1. Velocity components ahead of and behind the shock wave.

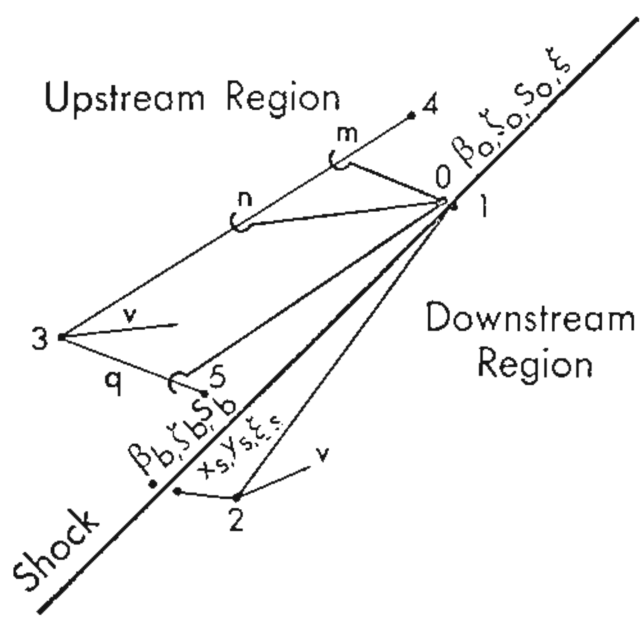

Fig. 2. Schematic diagram illustrating procedure for computing the next point along a shock wave in non-uniform flow.

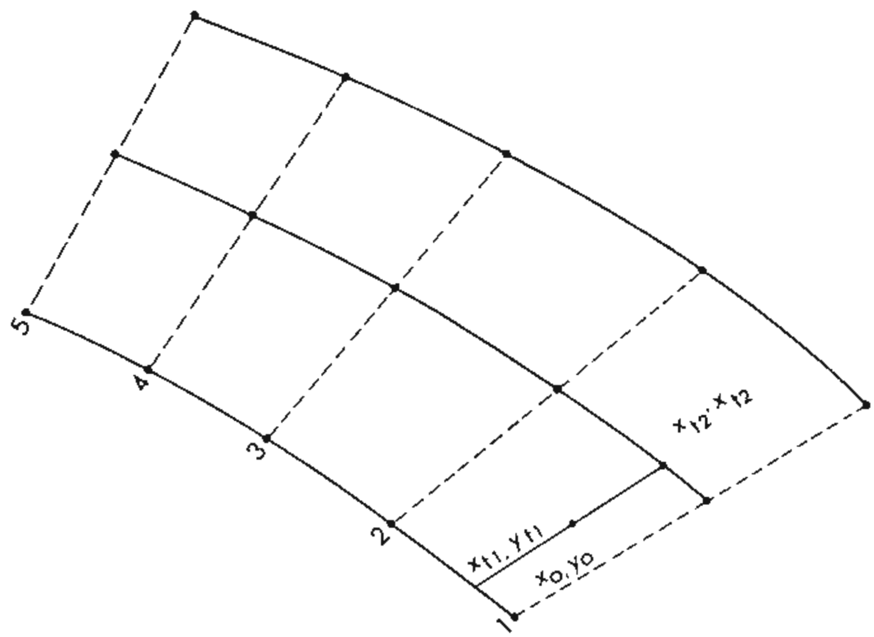

Fig. 3. Schematic diagram for selecting points 3, 4, and 5 in Fig. 2.

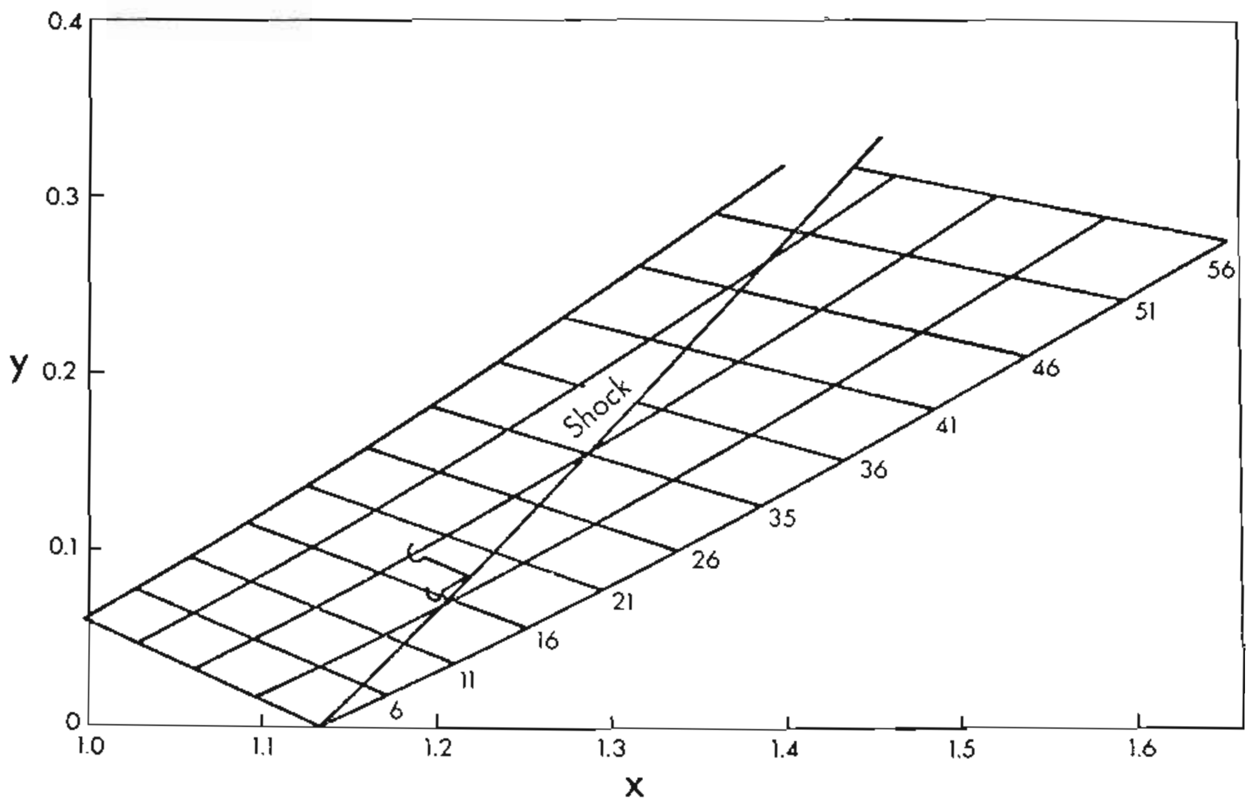

Fig. 4. Upstream Mach net and shock wave for wedge in diverging channel.

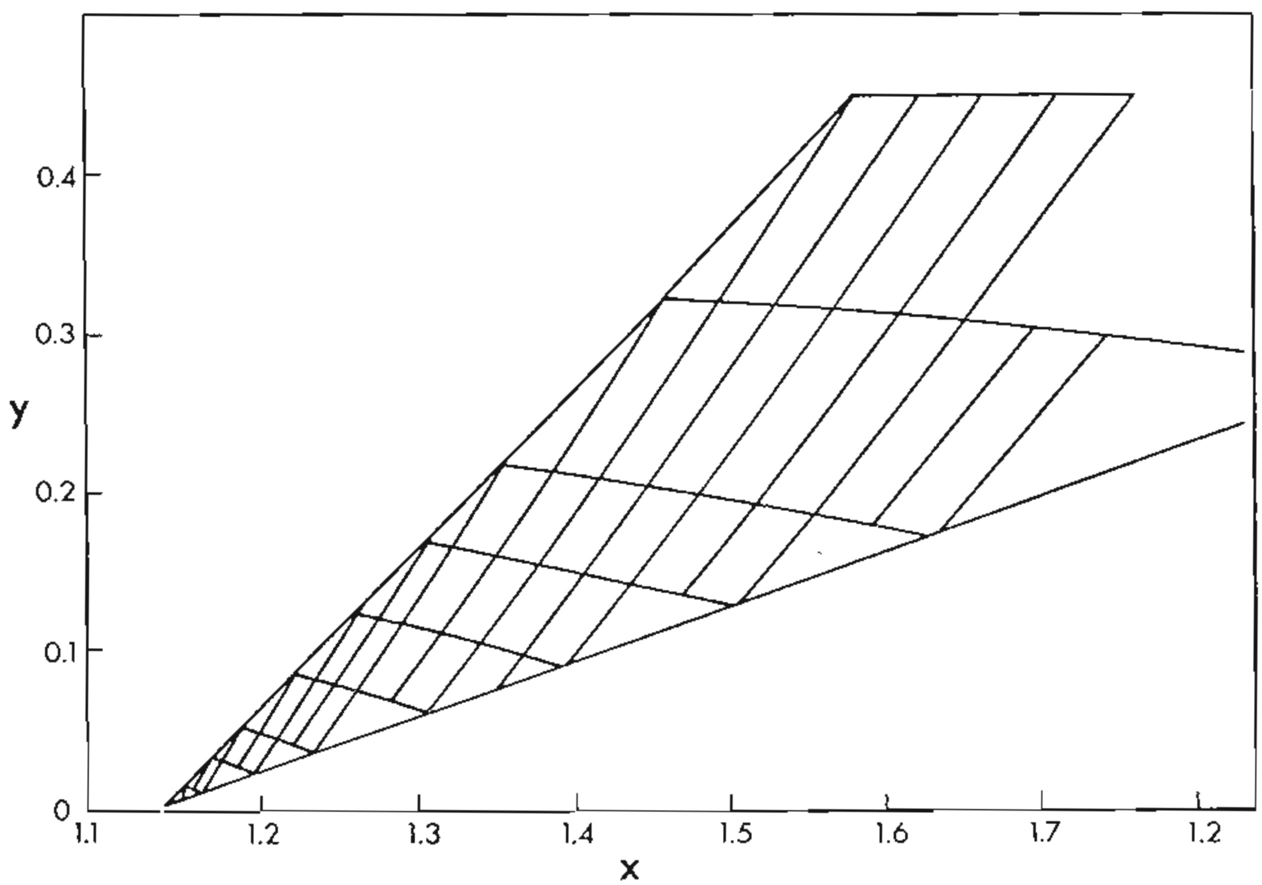

Fig. 5. Downstream Mach net between shock and wedge in diverging channel.

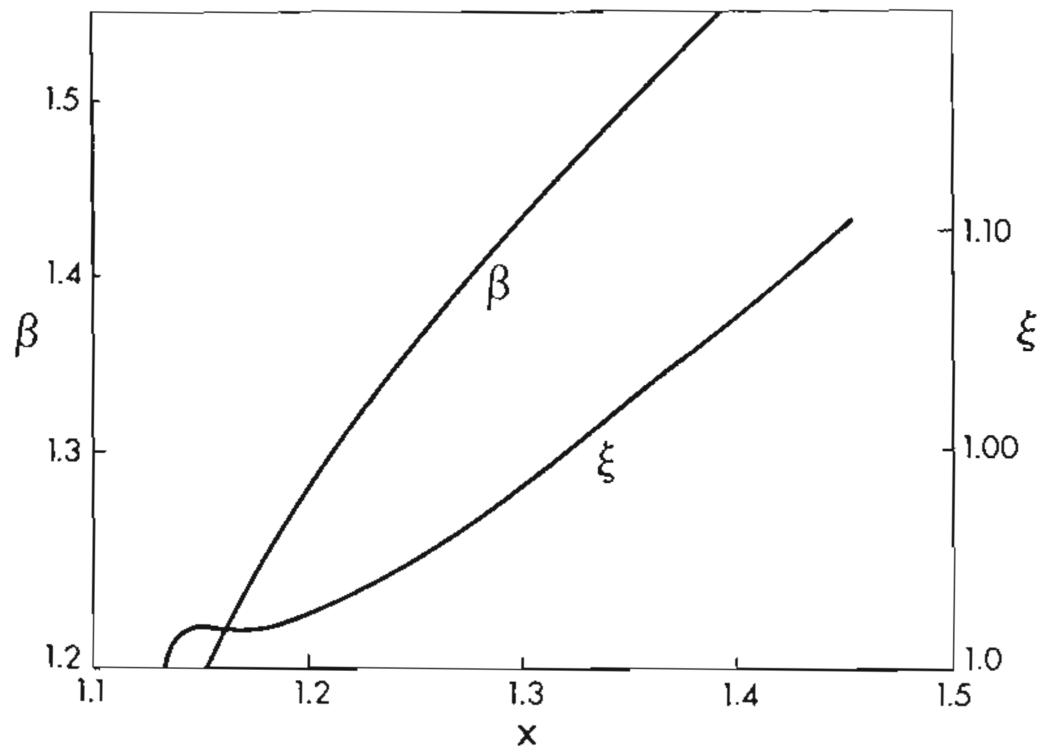

Fig. 6. The distribution of the cotangent β of the Mach angle along the wedge and the tangent ξ of the shock wave angle as functions of x, the horizontal coordinate.

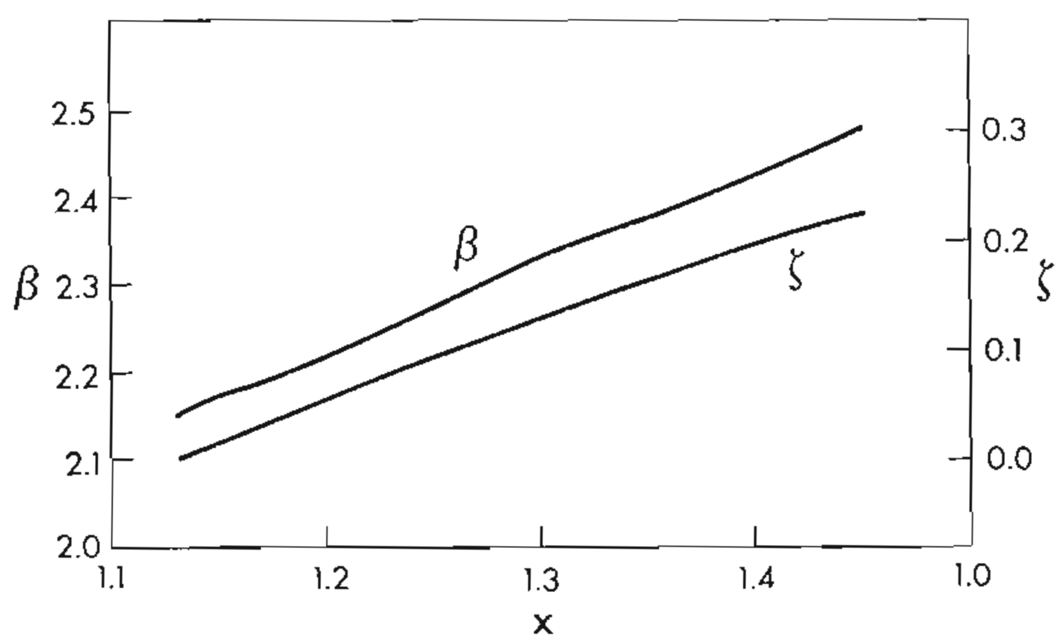

Fig. 7. The distribution of the cotangent β of the Mach angle and the tangent ζ of the flow angle ahead of the shock wave as functions of x, the horizontal coordinate for the flow over the wedge.

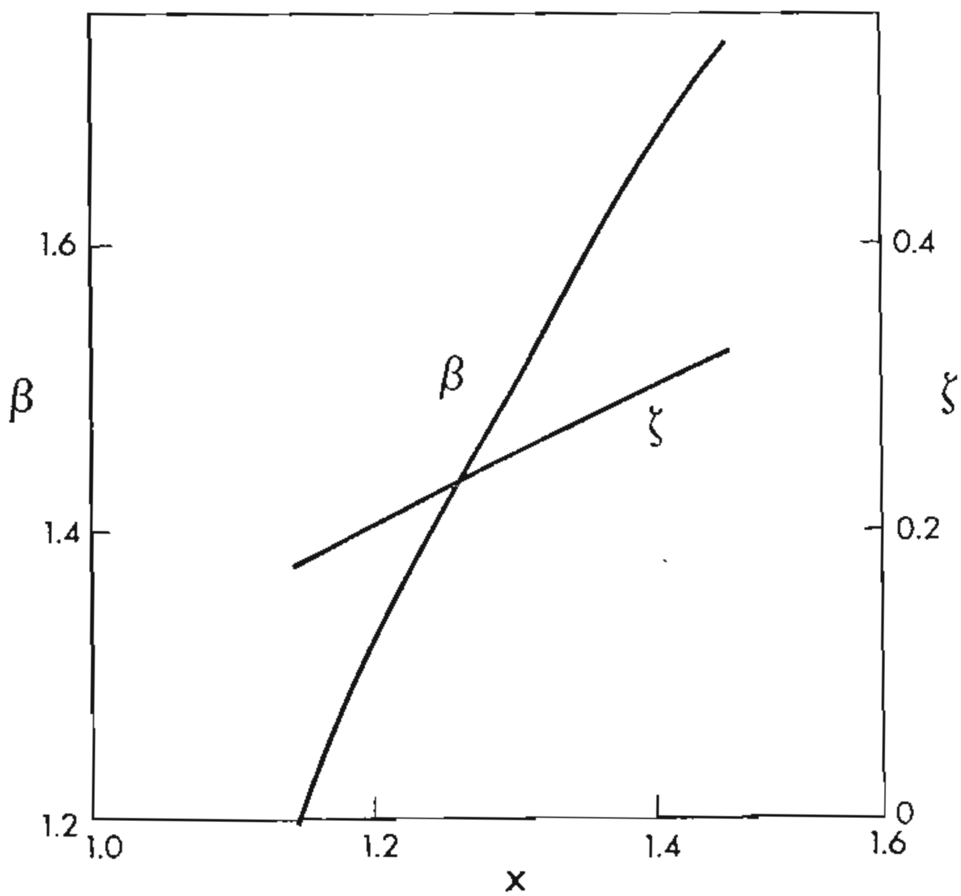

Fig. 8. The distribution of the cotangent β of the Mach angle and the tangent ζ of the flow angle behind the shock wave as functions of x, the horizontal coordinate for the flow over the wedge.

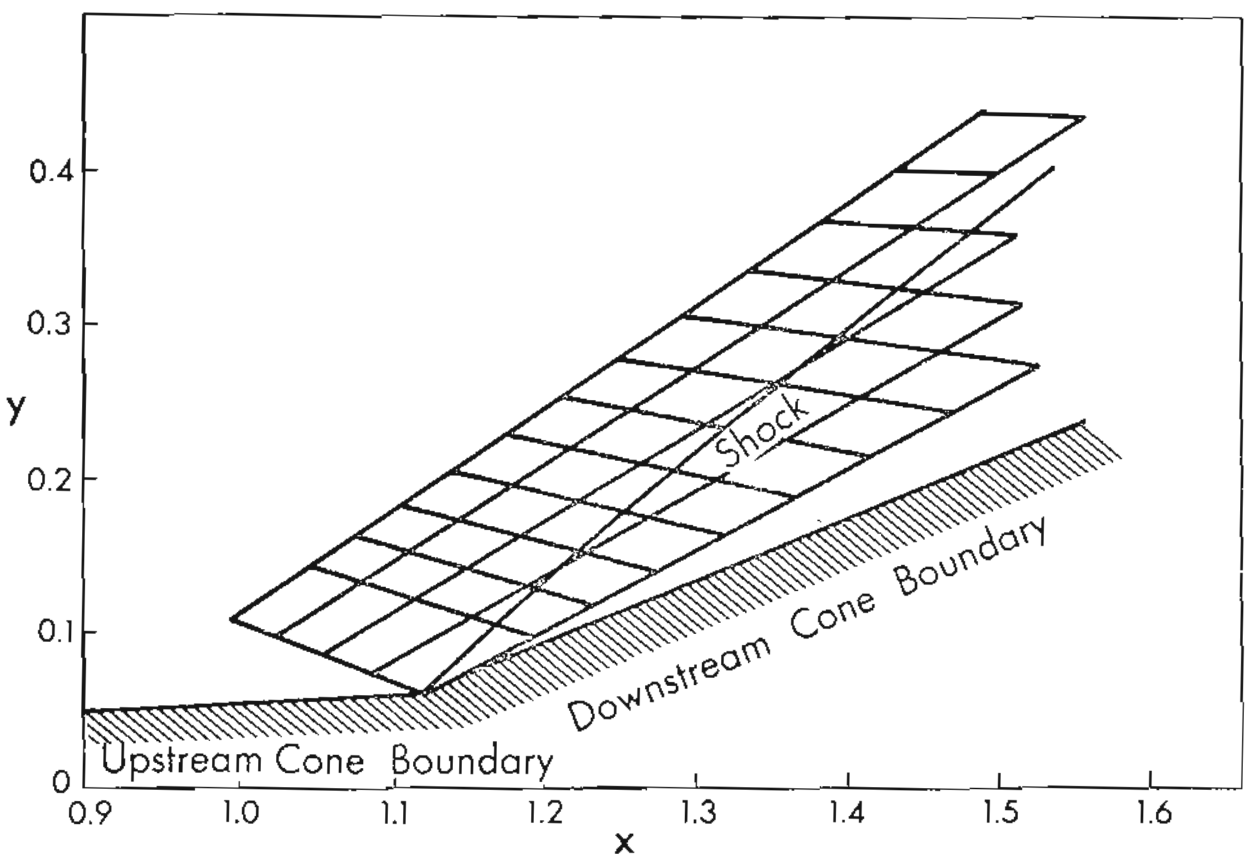

Fig. 9. Upstream Mach net, cone body, and shock wave in source flow.

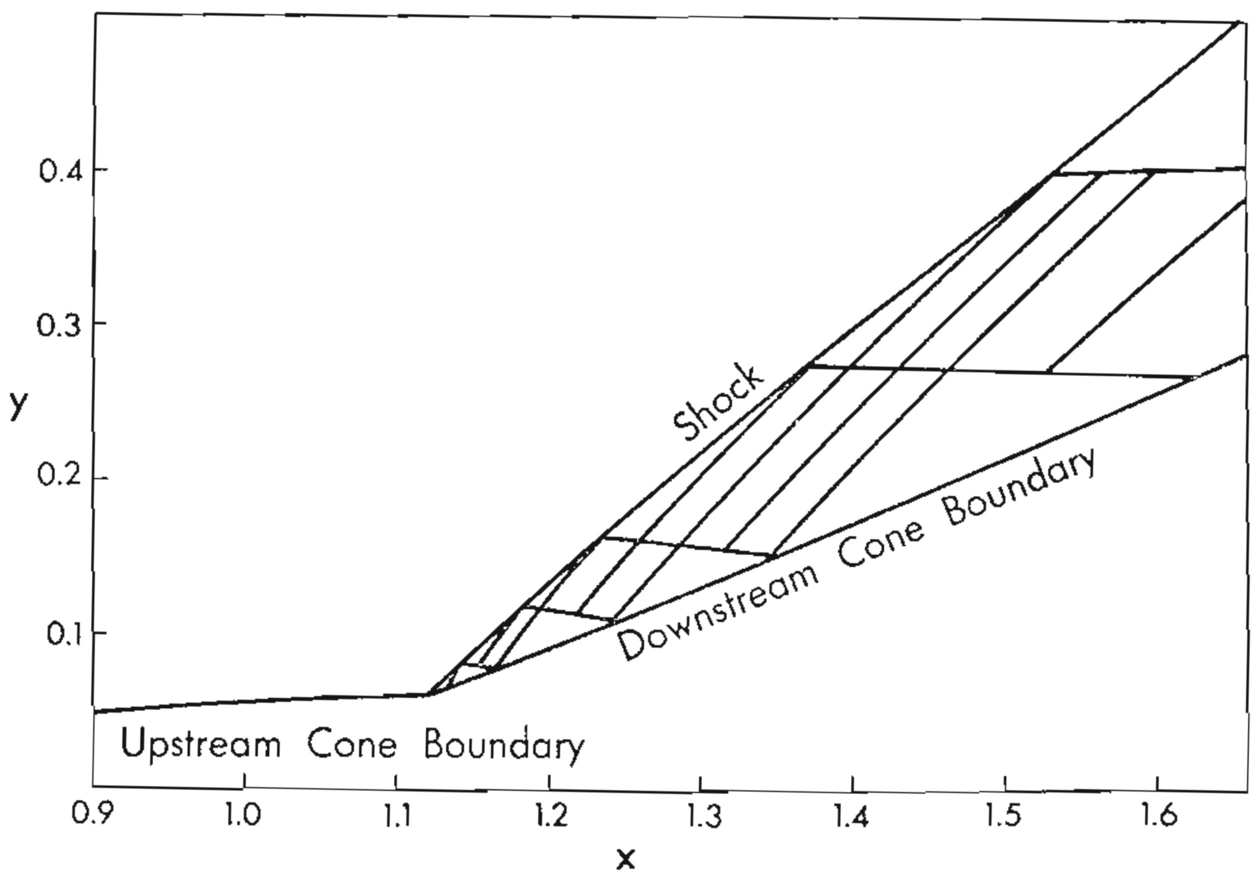

Fig. 10. Downstream Mach net between shock wave and cone body in source flow.

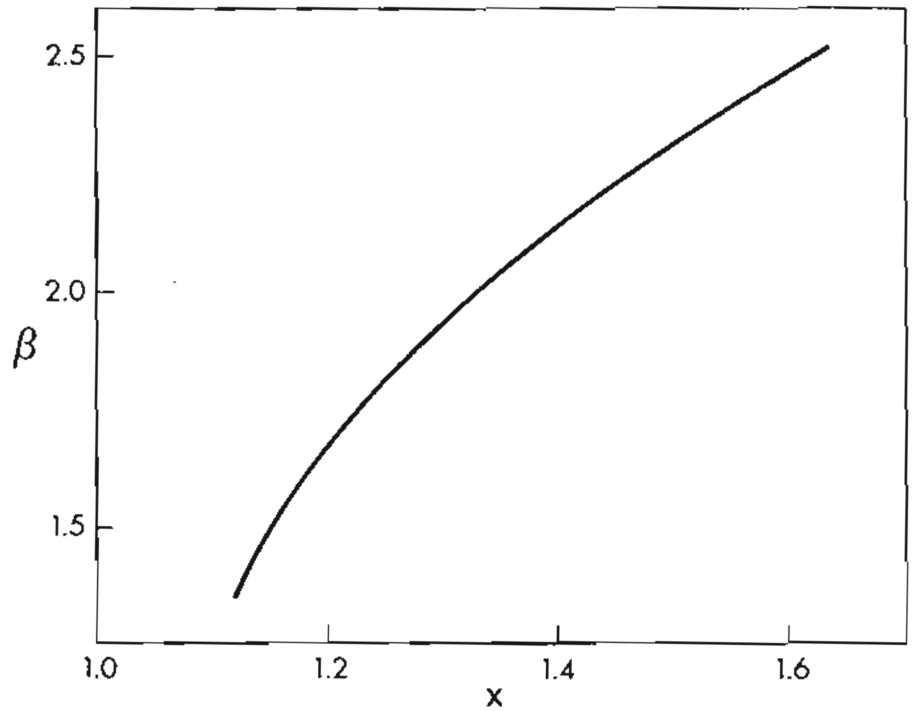

Fig. 11. The distribution of the cotangent β of the Mach angle on cone body as a function of x, the axial variable.

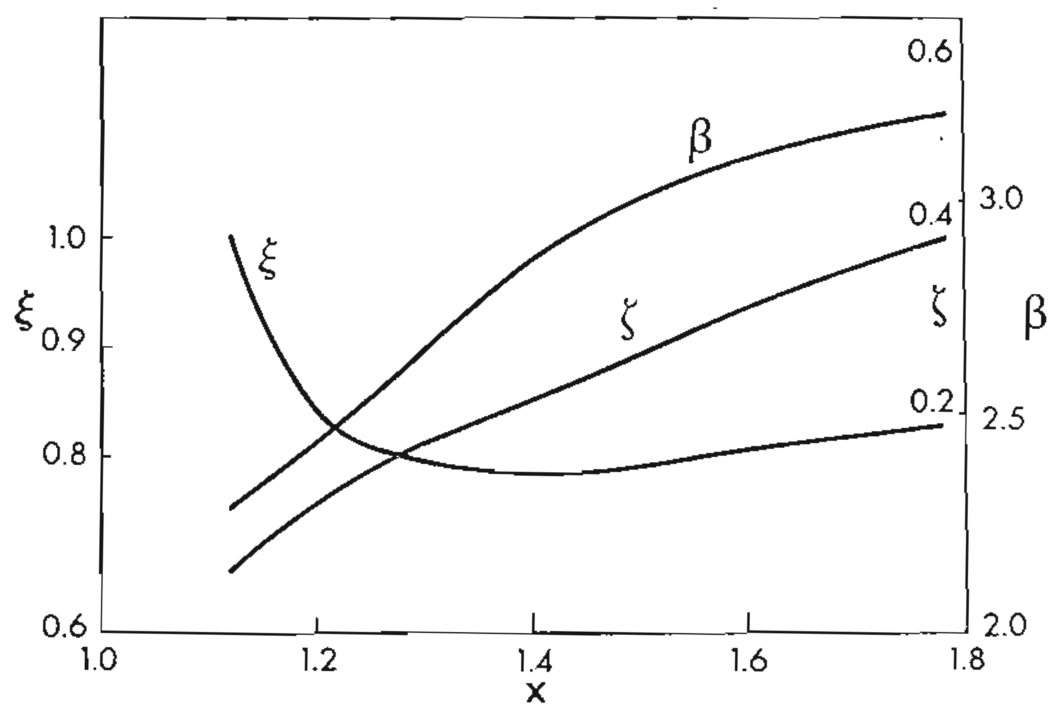

Fig. 12. The distribution of the cotangent β of the Mach angle, tangent ζ of the flow angle ahead of the shock wave, and tangent ξ of the shock wave angle as functions of x, the axial variable for the flow over the cone.

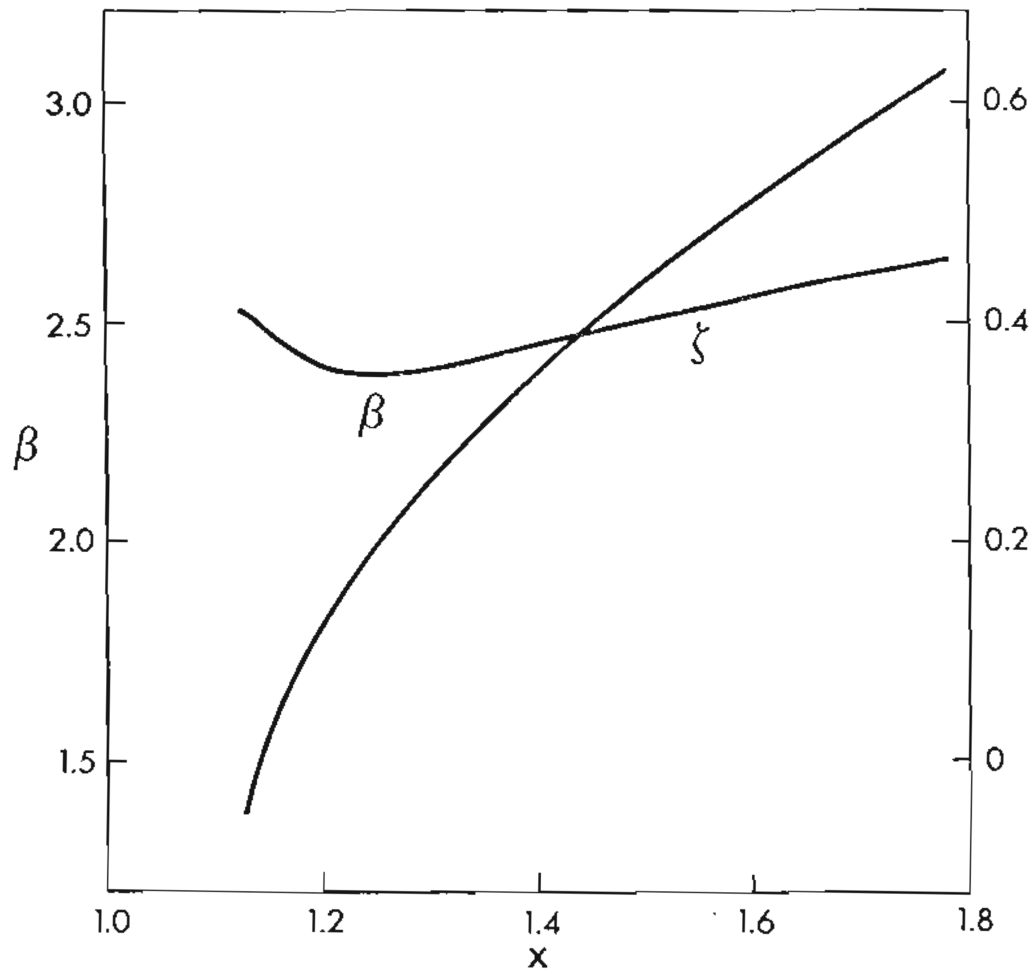

Fig. 13. The distribution of the cotangent β of the Mach angle and the tangent ζ of the flow angle behind the shock as a function of x, the axial variable for the flow over the cone.

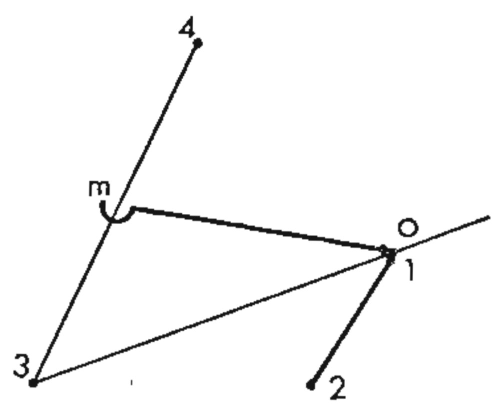

Fig. 14. Schematic diagram for illustrating procedure for computing the next point along a free streamline dividing two regions of supersonic flow.

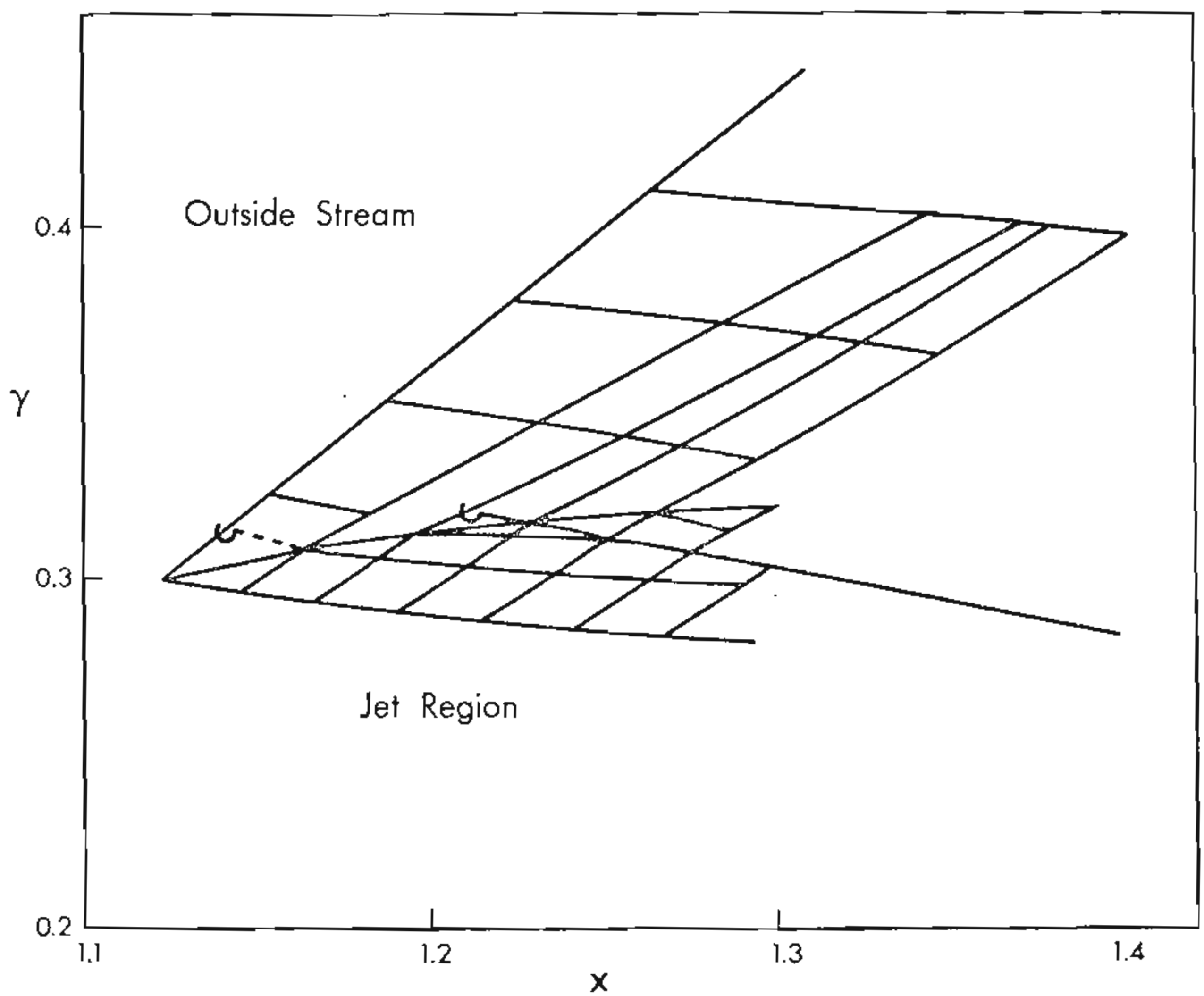

Fig. 15. An example of an axially symmetric supersonic jet issuing into a non-uniform exterior supersonic flow.