DYNAMIC INTERACTIONS BETWEEN HYPERSONIC VEHICLE AERODYNAMICS AND PROPULSION SYSTEM PERFORMANCE

Final Report

to
Aircraft Guidance and Controls Branch,
Guidance and Control Division
NASA Langley Research Center
Contract Number NAG-1-1205

by

G. A. Flandro
Boling Chair Professor of Advanced Propulsion
University of Tennessee Space Institute
Tullahoma, TN 37388

and

R. L. Roach and H. Buschek
Georgia Institute of Technology
Atlanta, GA 30332

July 1992

ABSTRACT

This report describes the development of a flexible simulation model for scramjet hypersonic propulsion systems. Its primary goal is determination of sensitivity of the thrust vector and other system parameters to angle of attack changes of the vehicle. Such information is crucial in design and analysis of control system performance for hypersonic vehicles. The code is also intended to be a key element in carrying out dynamic interaction studies involving influence of vehicle vibrations on propulsion system/control system coupling and flight stability.

Simple models are employed to represent the various processes comprising the propulsion system. A method of characteristics (MOC) approach is used to solve the forebody and external nozzle flow fields. This results in a very fast computational algorithm capable of carrying out the vast number of simulation computations needed in guidance, stability, and control studies.

The three-dimensional fore- and aft body (nozzle) geometry is characterized by the centerline profiles as represented by a series of coordinate points and body cross-section curvature. The engine module geometry is represented by an adjustable vertical grid to accommodate variations of the field parameters throughout the inlet and combustor. The scramjet inlet is modeled as a two-dimensional supersonic flow containing adjustable sidewall wedges and multiple fuel injection struts. The inlet geometry including the sidewall wedge angles, the number of injection struts, their sweepback relative to the vehicle reference line, and strut cross-section are user selectable. Combustion is currently represented by a Rayleigh line calculation including corrections for variable gas properties; improved models are being developed for this important element of the propulsion flow field.

The program generates: (1) variation of thrust magnitude and direction with angle of attack, (2) pitching moment and line of action of the thrust vector, (3) pressure and temperature distributions throughout the system, and (4) performance parameters such as thrust coefficient, specific impulse, mass flow rates, and equivalence ratio.

Preliminary results are in good agreement with available performance data for systems resembling the NASP vehicle configuration. Two geometries were used to demonstrate the code. The first was scaled from three-view drawings available in the open literature. The nose shape utilized a convex contour that could be handled adequately by the present method of characteristics package. A two-dimensional profile (UX-30) provided by NASA was also studied in a preliminary way to demonstrate effects of a ramp forebody diffuser. The calculations demonstrated the great sensitivity of propulsion system performance to vehicle configuration.

An extensive set of development calculations have been carried out with the code. Thermodynamic parameters, thrust vector data, thrust line-of-action, and pitch moment plots are presented as functions of angle of attack and flight Mach number. Sample calculations were performed in the Mach number range $5 < M_\infty < 25$, and angle of attack range $-5 < \alpha < 15$ degrees. Plots of flow properties and velocity distributions through the system show that the code generates valid simulations of hypersonic propulsion processes and their dependence on vehicle attitude.

CONTENTS

PREFACE

This is the final report of a one-year research effort under NASA Contract NAG-1-1205 to the School of Aerospace Engineering, Georgia Institute of Technology. The work was funded by the Aircraft Guidance and Controls Branch/Guidance and Control Division, NASA Langley Research Center. The goal of the research was development of an efficient method for estimating the thrust vector and other performance parameters for hypersonic airbreathing propulsion/airframe systems. An algorithm of this type is required in carrying out the extensive computations needed in hypersonic vehicle stability and control studies. Simple one-dimensional models are inadequate because they do not provide information regarding the thrust vector. Thrust produced pitch moments, and line of action of the thrust vector as functions of vehicle attitude. More complete models usually require application of numerical methods that are often computationally expensive. The approach described in the report is based on application of numerical techniques that are fast and efficient. The results are physically realistic and easily applied to a wide variety of hypersonic vehicle configurations.

The program managers were J. D. McMinn and J. D. Shaughnessy of the Aircraft Guidance and Controls Branch/Guidance and Control Division, NASA Langley Research Center. The Principal Investigator was Dr. G. A. Flandro who was a Professor of Aerospace Engineering, Georgia Institute of Technology when the program began. He is now the Boling Chair Professor of Advanced Propulsion at the University of Tennessee Space Institute. He continued to managed the program after moving to UTSI. Dr. Flandro was also responsible for the combustion modeling and performance calculations, and wrote the stand-alone Macintosh version of the final code.

Co-Investigator was Dr. Robert L. Roach of Georgia Tech who developed key numerical algorithms such as the several method of characteristics (MOC) routines required in the computations. Harald Buschek of Georgia Tech was a graduate assistant who was assigned the difficult task of writing the complex shock-expansion routines for the scramjet inlet analysis.

The authors wish to express their sincere thanks to Dana McMinn and John Shaughnessy for their assistance in the efficient operation of this study and for their patience when progress appeared to be slow. We also wish to acknowledge the efforts of Fred Morrell and Jerry Elliot of the Guidance and Control Division for giving us the opportunity to carry out this research.

SUMMARY

This report describes the initial development of a flexible simulation model for scramjet hypersonic propulsion systems. Its primary goal is determination of sensitivity of the thrust vector and other system parameters to angle of attack changes of the vehicle. Such information is crucial in design and analysis of control system performance for hypersonic vehicles.

Since the algorithm is intended for vehicle dynamic simulation rather than propulsion system design and optimization, simple models are employed to represent the various processes comprising the propulsion system. A method of characteristics (MOC) approach is used to solve the forebody and external nozzle flow fields. However, at the present stage of development only forebody shapes with convex lower surfaces can be accommodated. Concave surfaces, more representative of current vehicle concepts, generate shock waves in the solution domain that defeat the present implementation of the method of characteristics. Thus a shock-expansion package is employed as a preliminary concave forebody solver, but it is restricted to a two-dimensional nose geometry.

The three-dimensional fore- and aft body (nozzle) geometry is characterized by the centerline profiles as represented by a series of coordinate points and body cross-section curvature. The engine module geometry is represented by an adjustable vertical grid to accommodate variations of the field parameters throughout the inlet and combustor. The scramjet inlet is modeled as a two-dimensional supersonic flow containing adjustable sidewall wedges and multiple fuel injection struts. The inlet geometry including the sidewall wedge angles, the number of injection struts, their sweepback relative to the vehicle reference line, and strut cross-section are user selectable. Combustion is currently represented by a Rayleigh line calculation including corrections for variable gas properties, but improved models are being developed for this important element of the flow field.

The program generates: (1) variation of thrust magnitude and direction with angle of attack, (2) pitching moment and line of action of the thrust vector, (3) pressure and temperature distributions throughout the system, and (4) performance parameters such as thrust coefficient, specific impulse, mass flow rates, and equivalence ratio.

Preliminary results are in good agreement with available performance data for systems resembling the NASP vehicle configuration. Two geometries were used to demonstrate the code. The first was scaled from three-view drawings available in the open literature. The nose shape utilized a convex contour that could be handled adequately by the present method of characteristics package. A two-dimensional profile (UX-30) provided by NASA was also studied in a preliminary way to demonstrate effects of forebody diffusion. Thermodynamic parameters, thrust vector data, thrust line-of-action, and pitch moment plots are presented as functions of angle of attack and flight Mach number. Sample calculations were performed in the Mach number and angle of attack ranges $5 < M_\infty < 25$, and $-5 < \alpha < 15$ degrees.

The computations demonstrate the great sensitivity of the thrust vector to small vehicle pitch attitude changes. The thrust vector for the baseline vehicle configuration exhibits a significant downward component (negative lift contribution) under most flight conditions. This is largely the result of (1) the poor design of the baseline configuration, (2) the lack of boundary layer corrections in the aerodynamic representations, (3) neglect of adverse pressure gradients and flow separation in on the nozzle surfaces, and (4) instabilities caused by compression zones in the nozzle MOC solutions that led to inaccuracy in the predicted nozzle performance.

Despite the several obvious inadequacies of the present version of the program, it demonstrates significant promise for use in its intended role. The preliminary calculations indicate the great sensitivity of the results to changes in the shape of the forebody and nozzle contour, and the position and configuration of the cowl. Thus, effects of boundary layer offsets are vitally important and must be incorporated in later versions of the computer program already under development.

As the model evolves, it will also be used to estimate effects of propulsion system interactions with flexible vehicles. Eventually, effects of vehicle bending and unsteady atmospheric parameters will be assessed by means of advanced versions of the algorithm. Other improvements to be developed will include representation of three-dimensional flow effects (including effects of complex nozzle and fore-body cross sections), sensitivity to sideslip angle, improved combustion thermodynamics, and boundary layer corrections. Of greatest importance will be incorporation of adequate aerodynamics modules capable of dealing with shock waves in the solution domain.

To enable persons interested in the development of the code to assess both its weaknesses and potentials, a users manual, sample input files, and a complete source code listing of the current method of characteristics version of the program are included in appendices. A stand-alone version of the code is available that is configured for performing control system data base calculations on desktop computers (Macintosh with MC68030/68882 processor configuration and at least 4 MB of RAM). This version computes a complete propulsion system analysis case in about 40 seconds for the full MOC algorithm and about 15 seconds for the shock/expansion forebody version. Data is automatically stored for later use in plotting the results for a selected set of Mach number/angle of attack combinations. The Macintosh versions were used in preparing most of the performance plots presented in this report.

INTRODUCTION

This report describes the initial development of a flexible simulation model for scramjet propulsion systems. Its primary goal is determination of sensitivity of the thrust vector and other system parameters to the angle of attack of the vehicle. Such information is crucial in design and analysis of control system performance for hypersonic vehicles.

Several current programs concentrate on development of detailed representations of separate elements of the type described. Some of these modeling efforts have produced highly accurate codes intended mainly for detailed design and analysis purposes. However, they are usually not in a form that makes them practical for rapid performance calculations. For example, much current research is focused on detailed CFD solutions of the vehicle aerodynamics and its interaction with the airbreathing motor external and internal flow fields. Results from such research are obviously of crucial importance, but are not in a form that is of immediate use in performance tradeoffs and trajectory studies. Even if modified to be so they are generally expensive and complicated to run making parametric studies impractical.

Since the algorithm described herein is intended for vehicle dynamic simulation rather than propulsion system design and optimization, simple models are employed to represent the various processes through the propulsion system. A method of characteristics module is used to solve the forebody and external nozzle flow fields. The inlet consists of a two-dimensional supersonic flow containing adjustable sidewall wedges and multiple fuel injection struts. The number of struts and their sweeback relative to the vehicle reference line are user selectable. The inlet geometry including the sidewall wedge angles is also user selectable. Combustion is currently represented by a simple Rayleigh line calculation, but improved models are being developed for this important element of the flow field. The engine module geometry is represented by an adjustable vertical grid to accommodate variations of the field parameters throughout the inlet and combustor. Thrust, pitching moment , and thrust line are computed by integrating the pressure distributions over the entire surface wetted by flow involved in the propulsion process. The effects of forebody drag and inlet interactions are included in the thrust calculations.

Attention has been paid to incorporation of a convenient user interface to allow efficient study of the effects of small changes in vehicle configuration, structural characteristics, or propulsion system design. The program generates: (1) variation of thrust magnitude and direction with angle of attack, (2) pressure and temperature distributions through the system, and (3) performance parameters such as thrust coefficient, specific impulse and equivalence ratio. As the model evolves, it will also be used to estimate effects of propulsion system interactions with flexible vehicles. Eventually, effects of vehicle bending and unsteady atmospheric parameters will be assessed.

The algorithm is evolving as areas of weakness are discovered. The method of characteristics tools used in the curent version do not accommodate compressions of the type that would be typically employed on a hypersonic vehicle forebody as part of the propulsion diffusion process. They are also very sensitive to any irregularities in the geometry that lead to local compressions. No boundary layer corrections have been incorporated, so the result do not yet yield a reliable assessment of a given vehicle geometry. Nevertheless, the basic approach shows great promise and work has begun to overcome the present limitations and inaccuracies.

APPROACH

The computational algorithm consists of a set of flow field zones that are patched together at the interfaces into an integrated computational code. An important feature of the computation is incorporation of a flexible representation of the vehicle geometry with variable angle of attack. Work is currently underway to include effects of sideslip on the scramjet thrust vector using extensions of the methods described in what follows. Proven classical methods are utilized to produce a simple but sufficiently accurate representation of the vehicle. However, some of these have proven inadequate for realistic representation of a hypersonic vehicle. Clearly it will be necessary to accommodate shockwaves in the solution domain in the forebody since these play a major role in the diffusion process for the supersonic combustor. Shock/expansion methods are used temporarily for the forebody analysis until our shock-embedded MOC tool is completed

Vehicle geometry is currently input by means of a data file consisting of contour points along the body, cowl, and nozzle profiles. A large number of points is not required, since the program includes routines to fit the data points by splines. To demonstrate the approach, it is applied to a typical NASP geometry in this paper. Fig. 1 shows the geometry assumed for use as a baseline model. It is based on artists conceptions available in recent open literature. It is an unrealistic model for current vehicle design concepts. Some calculations were also carried out for a two-dimensional aerospaceplane configuration (UX-30) provided by NASA.

The user selects a convenient coordinate system such as the wing chordline and determines coordinates of an adequate number of points along the forebody, cowl, and nozzle. Since an axisymmetric forebody is assumed, it is also necessary to determine the curvature at each data station. Other input information required includes location and geometry of inlet sidewall wedges, injector struts and the combustor configuration. Figure 1 shows a typical representation of the vehicle profiles needed in the computations. The curved bow shock, nozzle shock and slip lines are illustrated.

A view of the set of flow features comprising the propulsion system of a hypersonic vehicle are also shown in Fig. 1. The underside of the vehicle is divided into 4 main components: forebody, inlet, combustor, and aftbody. In each region an appropriately rapid but accurate computational method was chosen. The guiding principle for the selection of methods to be used was speed, flexibility, accuracy, and robustness. By flexibility is meant that future enhancements to the code must be compatible with the methods chosen. This impacts not only the type of method but choices about its implementation. The methods chosen for each of the regions is briefly outlined below with a detailed description provided in later sections.

To put the four main solving procedures together, several smaller regions have been devised which pass data from one section to another, provide supplementary computations as needed as boundary or starting conditions for the main regions, and compute forces and moments of the overall flow. These will be described in some detail below as well.

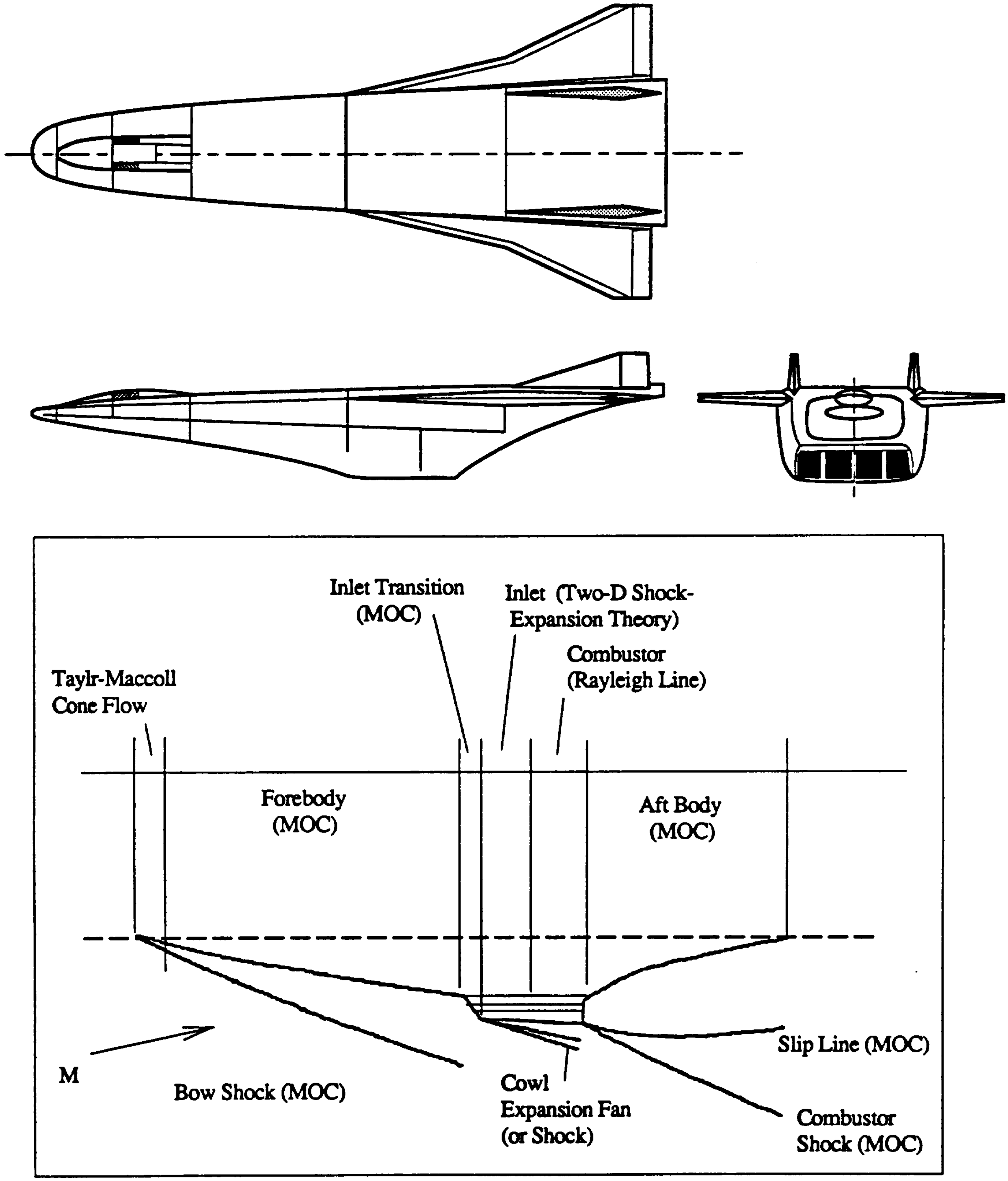


Figure 1. Assumed Hypersonic Vehicle Configuration

Methods used in the 4 main regions

1. Vehicle Forebody - Here a two dimensional method of characteristics is used to compute the flow between the forebody and the shock. A cone flow solution is used to provide the starting conditions. Work is in progress on a blunt body starting condition as well. The flow and the shock position at a number of streamwise stations are computed until the inlet is reached. Fig. 2 shows the near field of the forebody showing the individual stations used and their associated characteristics lines.

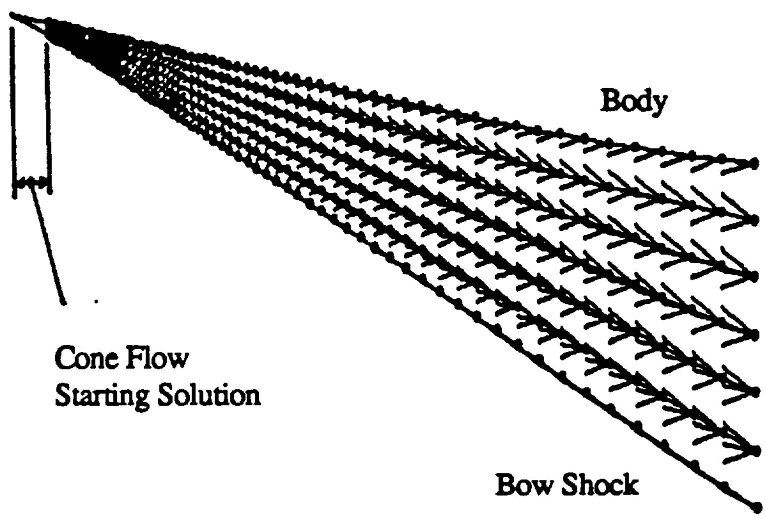


Figure 2. Characteristics net in the forebody region.

2. Inlet - The flow in the inlet is computed using shock-expansion theory. Since the inlet flow is, in general rotational, the inlet is divided into a number of streamtubes vertically. The flow in each streamtube consists of shock waves, expansions, slip lines, and solid walls and their intersections. The code logic permits an arbitrary placement of up to 5 struts in the inlet. The procedure stops when the combustor section is reached as specified by the user. Fig. 3 shows the top view of the cross section of a typical inlet streamtube showing fuel injection struts. This is quite out of line with current design directions, which utilize sidewall propellant injection and no side panel wedges.

3. Combustor - Here a simple Rayleigh line heat addition is used. The user may specify an equivalence ratio greater than unity. The algorithm allows combustion heat release only up to the point that a maximum combustor static temperature is reached. Any excess propellant is ûtilized to modify the mixture properties of the combustion products. When hydrogen is utilized as the fuel, the excess H_2 tends to lower the molecular weight of the gases exiting the combustor with an attendant performance advantage. Combustion chemistry including effects of dissociation are not presently accounted for in the code.

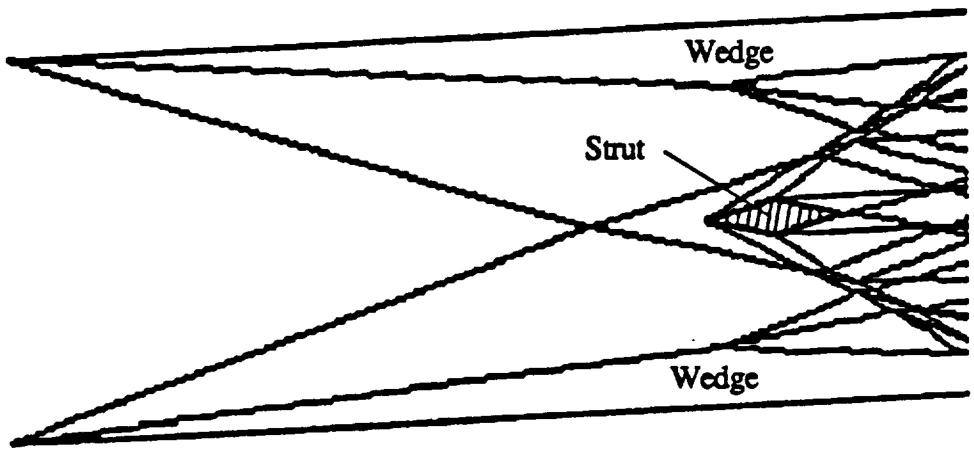


Figure 3. Top view of an inlet streamtube showing the shock/expansion structure.

4. Aftbody - Here the method of characteristics is again used, but the high pressure gas from the exhaust of the combustor creates a shock wave which extends away from the body and also creates a slip line. The computation of the forebody was rewritten for this case and stands alone as a separate module. Fig. 4 shows a typical flow pattern of the shock wave, the slip line, and the characteristics net.

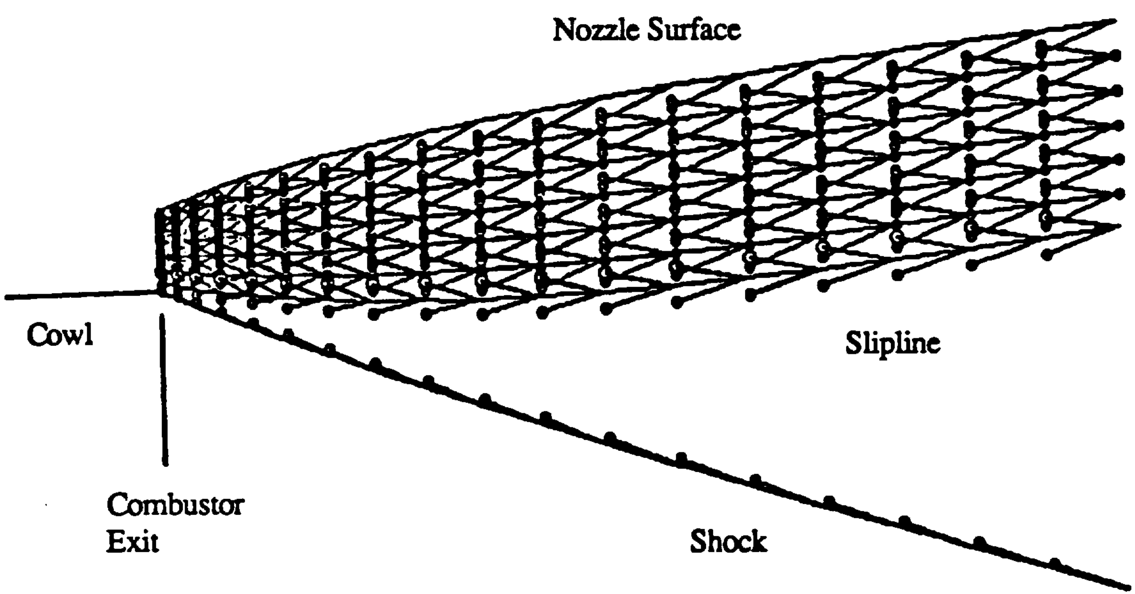


Figure 4. Characteristics net in the aftbody region.

ANALYSIS

A more detailed description of the solution methods, the governing equations and their incorporation in the complete algorithm are given in this section.

Rotational Method of Characteristics for the Vehicle Forebody

The partial differential equations for inviscid, supersonic flow over solid bodies are hyperbolic and may be solved by a streamwise marching technique known as the Method of Characteristics. The term characteristics refers to the existence of real directions from a point in the flow along which certain combinations of quantities are constant allowing the equations to simplify to ordinary differential equations. In solving these equations in the characteristic directions, the fixed quantities are thus automatically preserved. The method thus maintains great fidelity to the solution of the original partial differential equations.

For rotational flows the ordinary differential equations remain nonlinear. Hence an iterative method is required. Since there are a large number of approaches to the solution of nonlinear equations it is not surprising that there are correspondingly a large number of methods reported. An excellent summary is given by Chushkin [1]. We proceed in a manner most closely related to that of Rakich [2] who used a variant of the Hartree method (see, e.g.. Fox [3], Ch. 27). The methods reported by these authors are for a completely three dimensional flow about arbitrary bodies and are rather time consuming. However, the axisymmetric versions are significantly faster and are appropriate for incorporation in the scope of this analysis. The axisymmetric version was also chosen since, if interest warrants and computational capability does not prohibit, it would be relatively straightforward to modify the procedure to be able to compute a full 3D configuration. This is to be preferred over a CFD type shock capturing procedure since the forebody geometry is relatively simple and the MOC, even in 3D, requires only a single streamwise sweep.

Another advantage of the formulation is the relative ease in being able to include imbedded shocks in the analysis. This is so since, as it will be shown, the user has a line along which to distribute any number of points. If a shock crosses the line, then two points are assigned to the intersection, one for each side of the shock. It is then straightforward to apply the shock/characteristics conditions to each of these points in the same manner as the shock point described above.

The basic equations for rotational , compressible flows are given by:

$$\frac{\partial \rho}{\partial s} = \frac{1}{a^2}\frac{\partial p}{\partial s}$$

$$\frac{\beta}{\rho V^2}\frac{\partial p}{\partial c_1} + \frac{\theta}{\partial c_1} = -\frac{\sin\theta}{Mr}$$

$$\frac{\beta}{\rho V^2}\frac{\partial p}{\partial c_2} - \frac{\theta}{\partial c_2} = -\frac{\sin\theta}{Mr}$$

$$\frac{\gamma}{\gamma - 1}\frac{p}{\rho} + \frac{V^2}{2} = H$$

where c1 and c2 refer to the characteristic directions due to local Mach waves. The other characteristic direction required for rotational flows is a streamline. The first of these is an equation of state. The second and third equations are momentum conservation, and the last is an energy equation. Note that these are first order. In addition:

$$a = \sqrt{\frac{\gamma p}{\rho}} \qquad \beta = \sqrt{M^2 - 1}$$

The derivatives are to be replaced by finite differences. The differencing takes place between streamwise stations and along the indicated characteristic direction. That means that the directions must be locally computed. Since the flow angle is one of the unknowns, these directions are not known to start with, so an initial guess is made.

Generally, the solution procedure is to march downstream from some initial line of data. In marching downstream, the equations are solved at a series of points along a vertical line at the next downstream station. Wall information is used to start at the surface, and the computations proceed outward, point by point until the shock is reached. The solution at this station is then used to proceed to the next. This continues until the inlet is reached. Two columns of points illustrating the set-up of the grid is shown in Fig. 5. Here, the characteristic lines are drawn from one point back upstream to the previous station. Since the values of the dependent variables at this point are dependent only on upstream information, all the points on the column can be computed independently.

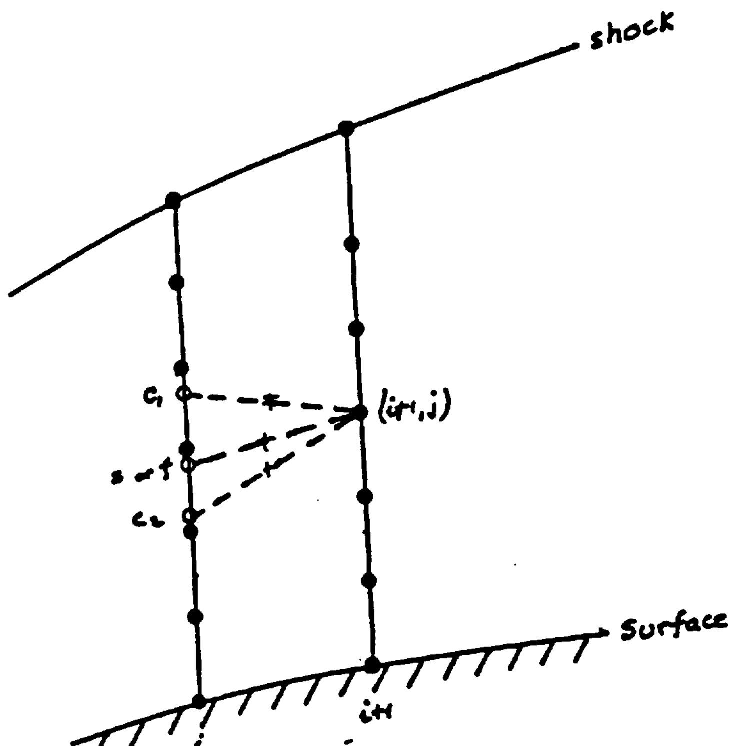


Figure 5. Finite difference grid points and characteristic directions used in the differencing.

The finite difference forms of the equations are as follows.

$$\frac{\rho_{i+1}-\rho_s}{\Delta s}-\frac{1}{a_s^2}\frac{p_{i+1}-p_s}{\Delta s}=0 \qquad (1)$$

$$\left(\frac{\beta}{\rho V^2}\right)_1\frac{p_{i+1}-p_1}{\Delta c_1}+\frac{\theta_{i+1}-\theta_1}{\Delta c_1}=-\frac{\sin\theta_1}{M_1 r_1} \qquad (2)$$

$$\left(\frac{\beta}{\rho V^2}\right)_2\frac{p_{i+1}-p_2}{\Delta c_2}-\frac{\theta_{i+1}-\theta_2}{\Delta c_2}=-\frac{\sin\theta_2}{M_2 r_2} \qquad (3)$$

$$\frac{\gamma}{\gamma-1}\frac{p_{i+1}}{\rho_{i+1}}+\frac{Vi+1^2}{2}=H \qquad (4)$$

Here the subscripts, 1, 2, and s refer to the characteristic line on which the equation is being solved. Hence the right hand side of the second equations is being solved between the current point and the point on the previous station in the direction of the c1 characteristic. The subscript "1" on the variables indicates that an average value is to be used. Since initially, no information is known at the unknown point, the first guess is the value at the previous station. When the first iteration is carried out, an approximate value of the variables at the new station will be known, and an average can then be used.

As written, the equations have been "linearized" with the derivatives containing the unknowns in equations (1), (2) and (3). Thus the nonlinear coefficients are lagged by evaluating them with the last known values. These terms are usually formed as averages between the two stations. Equations (2) and (3) can then be combined to eliminate the flow direction. This gives

$$p_{i+1}=\frac{\left(\frac{\beta}{\rho V^2}\right)_1 p_1+\left(\frac{\beta}{\rho V^2}\right)_2 p_2-\frac{\sin\theta_1}{M_1 r_1}-\frac{\sin\theta_2}{M_2 r_2}}{\left(\frac{\beta}{\rho V^2}\right)_1+\left(\frac{\beta}{\rho V^2}\right)_2} \qquad (5)$$

The values of the variables at the last station which get used in these finite difference expressions are those which come from the intersection of the characteristic lines with the station location. As can be seen in Fig. 5, these intersections do not, in general, coincide with the points at the last station. Thus, an interpolation must be performed. This is currently done using a cubic passed between the nearest four points. Near the wall or the shock, a three point, quadratic interpolation is used.

1. From the initial data line, a new shock location is guessed. Currently this is done by simply extending it from its current position in a straight line.
2. In the distance between the surface and the shock, a number of grid points is distributed equally.
3. Starting at the wall, where the flow angle is already known, equation (5) is used to solve for the pressure, equation (1) is used to solve for the density, and equation (4) is used to solve for the surface velocity.

4. Equations (5), (2), (1), and (4) are solved for pressure, flow angle, density, and total velocity respectively for each of the remaining interior points. The bulk of the work is done in this step as most of the points are interior points. The details of this step are given below.
5. At the shock point, using the freestream Mach number and the initial guess as to the shock angle, the pressure downstream of the shock can be computed along with the flow direction. Either of these may then be used to compute the other using equation (3). For example, if the pressure was used to compute the flow angle with equation (3), then this is compared with that given by that computed from the guessed shock angle. The mismatch between these guides the adjustment to the shock angle. This is currently done by computing a new shock angle using the freestream Mach number and the flow angle given by equation (3). This new shock angle is averaged with the old one and the average value is used as the new shock angle guess. once convergence of the shock angle has been reached, equations (1) and (4) are used to find the density and total velocity.
6. A new station downstream of the current one is located; the process repeats starting at step 2.

Fig. 2 and 4 showed how the characteristic net is built up down stream from the initial data. The lines joining the solid points on one line to the open points on the previous line are the characteristic lines. Those shown are those which are obtained after several iterations. In each case, convergence was relatively rapid (usually 3-5 iterations are required per station).

It is necessary to provide initial starting conditions for the MOC procedure. Since the equations governing supersonic, inviscid flow are hyperbolic, two sets of characteristics are necessary. Since the flow is also rotational, a third characteristic is also needed. For the leading edge region of the vehicle, this third characteristic is simply the freestream flow direction. Specification of the others requires knowledge of the leading edge geometry. This information will be obtained from the specification of the vehicle geometry and will be assumed to be known for purposes of this discussion. The remaining conditions required are the starting conditions. These are the values of the dependent variables at some station extending from the body surface to the bow shock. At present, the starting conditions for the MOC procedure is provided by computing placing an equivalent cone near the nose and applying cone-flow theory.

Conical Flow Analysis

Input to this section requires only the specification of the freestream Mach number, ratio of specific heats, freestream static or stagnation temperature, and nose cone angle. In the case of a non conical nose, but with small streamwise radius of curvature, an effective cone angle can be defined as the cone which best fits the average slope of a specified small percent of the forebody as shown in Fig. 6. Currently this is 5-10%. The cone is projected forward to compute a cone apex. From here, the cone flow analysis computes the shock angle and downstream properties.

The basic equations governing the supersonic flow over cones must be solved simultaneously for the two velocity components V_r and V_ω, the radial and tangential velocity components along a ray emanating from the cone apex. Since they are nonlinear, it is necessary to use an iterative technique for solution. The method we use here, starts by eliminating V_w to obtain a single differential equation for V_r.

$$-\left[\frac{\gamma+1}{2}\left(\frac{\partial V_r}{\partial \omega}\right)^2 - \frac{\gamma-1}{2}\left(V_{max}^2 - V_r^2\right)\right]\frac{\partial^2 V_r}{\partial \omega^2} + \left[\frac{\gamma-1}{2}\left(V_{max}^2 - V_r^2\right)\cot\omega\right]\frac{\partial V_r}{\partial \omega} +$$

$$\left[(\gamma-1)\left(V_{max}^2 - V_r^2\right) - \gamma\left(\frac{\partial V_r}{\partial \omega}\right)^2\right]V_r = \frac{\gamma-1}{2}\left(\frac{\partial V_r}{\partial \omega}\right)^3 \cot\omega$$

In addition:

$$V_\omega = \frac{\partial V_r}{\partial \omega}$$

where

$$V_{max} = \sqrt{2c_p T_0}$$

Finite difference relations are used to change the differential equation into an algebraic equation. This equation is solved by starting from an initial guess as to the shock angle and marching in along radial lines (whose origins are at the nose point) from the shock to the surface. We are currently using 21 stations. Upon reaching the surface, a check is made to see if the normal component of the velocity

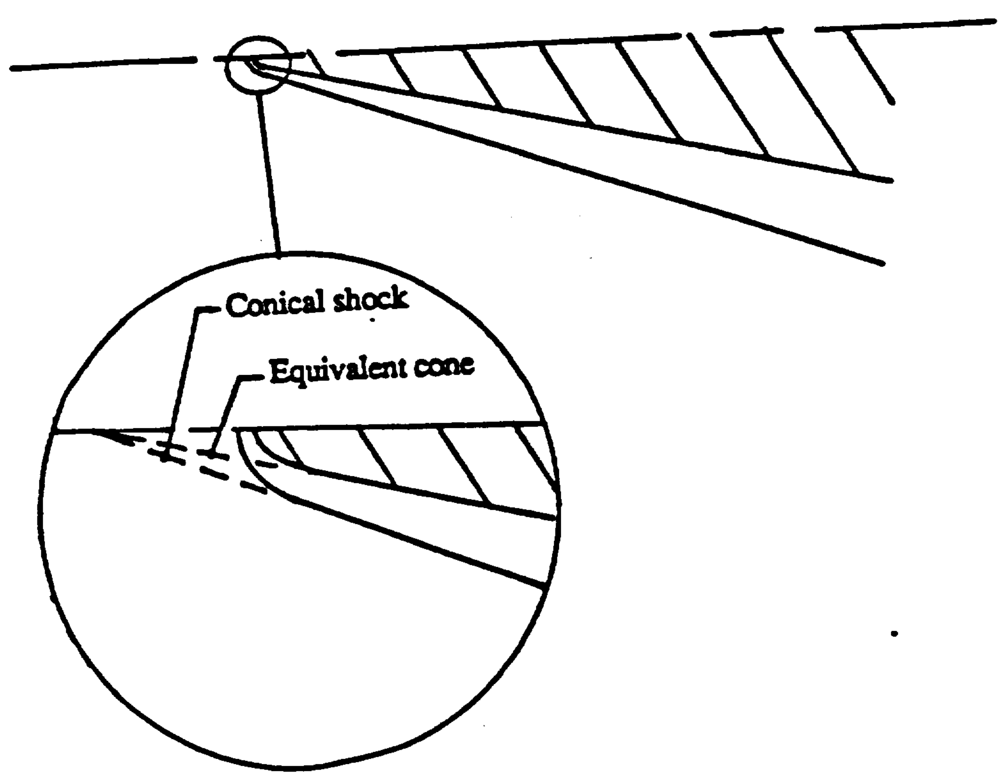


Figure 6. Equivalent cone for a slender forebody with small streamwise nose radius.

is zero. If not, the shock angle is adjusted and the procedure is repeated. Convergence is attained when V_w is less than a specified tolerance. We are currently using a rather severe tolerance and have found that about 4 to 6 iterations have been required.

While it is not vital to make a good first guess as to the shock position, doing so reduces the number of iterations required. There have been several approaches for providing an initial guess and these include:

1. using the Mach angle associated with the freestream Mach number
2. using the results from 2D oblique shocks
3. averaging these two
4. using an analytical curve fit

Use of either of the first two of these results in a doubling of the number of iterations in some cases since. The first badly underestimates the shock angle for fat cones and the second always overestimates the angle. The most popular is averaging these and we initially used that for our initial shock angle. However, we found that the rather excellent analytical curve fit of Doty and Rasmussen[4] to always be within 2 degrees of the final answer and subsequently obtained a much closer approximation with it. For cones at an angle of attack, they also provide a very close analytic modification to the exact results for the modified shock cone. This is fortunate since the computation of the shock cone in this circumstance would be otherwise a rather lengthy computation.

Since we found this aspect of the task to be useful as a stand-alone tool, a program was written around this procedure in BASIC and an example of the output screen in shown in Fig. 7 The user specifies the freestream Mach number and cone angle. Currently, the program assumes a ratio of

Mx = 2.00 δ = 20.0000 (deg)

Mach Angle 30.0000 deg

Max Wedge Angle 22.9735 deg

θ (Wave Angle) = 53.4229 deg (4 Iterations)

Mxn	My	Py/Px	Rhoy/Rhox	Ty/Tx	P0y/P0x
1.6061	1.2102	2.8429	2.0420	1.3922	0.8929

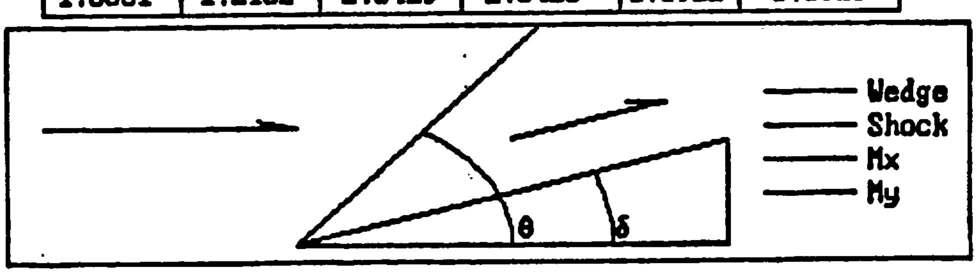


Figure 7. Example of the SHANG stand-alone program output.

specific heats of 1.4 which is typical for air and also assumes a freestream temperature of 400 deg R so that the velocities can be computed from the Mach numbers and sound speeds. The figure shows the number of iterations required, the shock wave angle, the surface pressure coefficient, and the surface Mach number. Properties along the rays is also available but not shown. These will be used as the initial conditions by the MOC procedure. The figure also shows a sketch of the resulting flow so the user can easily visualize the flow. The cone is drawn with the resulting shock and several streamlines are drawn to show how the flow turns behind the conical shock. A low Mach number was chosen only so that the features of the output could be seen. The result of putting the cone shown at a 15 degree angle of attack is shown in Fig. 8. The output screen shows the resulting shock angle for various azimuthal angles around the cone.

	M	ω	μ	P2/P1	Fan Angle
1	2.0000	26.3798	30.0000	0.2752	29.3118
2	2.8306	46.3798	20.6882		

ω = Prandtl-Meyer Function (deg)
μ = Mach wave angle (deg)

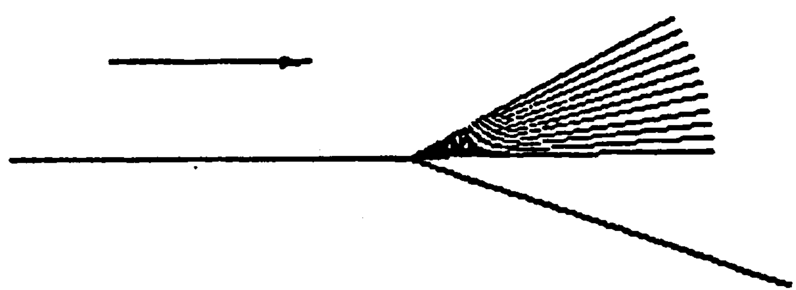


Figure 8. Example of the PM stand-alone program output.

Inlet face

Depending on the relative change of the stagnation pressure in the vertical direction, the flow entering the inlet will be divided into a number of streamtubes. The vertical stations of the forebody in general will not line up in any convenient way with these streamtubes. Further, the inlet face is not vertical either. Thus, the conditions at the inlet face must be computed in a separate fashion from the conditions in the forebody region. The rotational method of characteristics are still used, here but a separate routine was written specifically for the transition to the inlet streamtubes.

Inlet Analysis

The flow downstream of the shock is likely to be rotational and as such stagnation quantities will thus be different on each stream line. It thus seemed prudent to divide the propulsion section into a series of small stream tubes with roughly rectangular cross section, in which the stagnation quantities are roughly constant. The analysis in each of these stream tubes can then be accomplished individually and relatively simply using 2D oblique shock relations. In Fig. 9 the slicing of the propulsion unit into the streamtubes is illustrated. The internal geometry, a cross section view of one of the streamtubes, was seen in Fig. 3. The analysis then consists of the dividing process and the flow computation in each streamtube. The division process is currently envisioned to include an analysis of the severity of the stagnation quantities across the streamlines to assess the number of streamtubes that ought to be created. This process would also include the generation of the conditions across the face of each streamtube from the forebody MOC data.

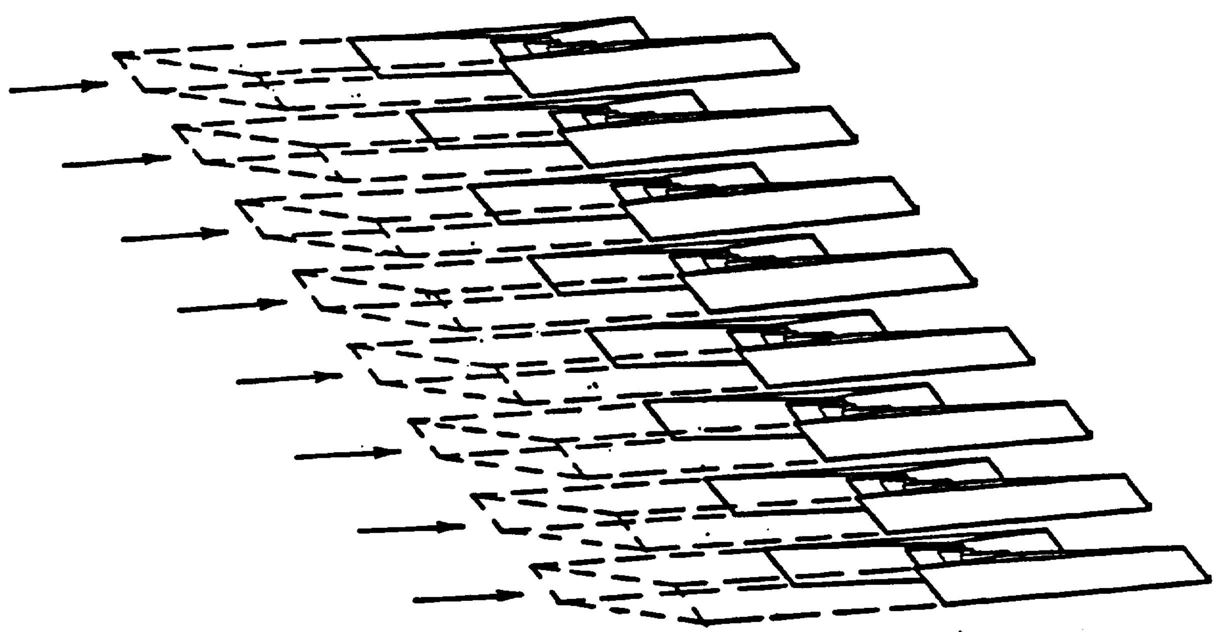


Figure 9. Division of the combustion chamber into separate vertical zones for computation. (Each zone contains a single stream volume in which the shock-expansion inlet calculations, followed by Rayleigh line heat addition is performed)

The main effort, however, is to compute the supersonic flow inside the scramjet using shock/expansion theory. The internal geometry consists of two side walls with ramps which compress the flow and a series of struts in between the walls near the minimum area. The flow through this region will encounter the two oblique shocks from the side walls which may or may not intersect prior to encountering the struts. The struts will create more shocks and expansions depending on their shape. These waves will likely intersect the sidewall shocks and all waves will eventually strike a some wall. One other flow feature will be the creation of slip lines at any of the wave intersections and, of course, there will be intersections of waves and slip lines (which we shall generically also call a wave). Thus the flow inside the scramjet can become rather complex, but we have outlined a procedure below, have identified the subtasks required, and have begun to create coding for these subtasks.

It is first useful to identify all possible types of waves and events which can occur in the flow in some organized fashion.

The five possible wave types are:

- shock waves of family I (right running)
- shock waves of family II (left running)
- expansion waves of family I
- expansion waves of family II
- slip lines (created at each wave intersection)

The ten possible intersections are:

- shocks of the same family
- shocks of different families
- shock intersecting a wall
- shock intersecting a slip line
- expansions of same family
- expansions of different families
- expansion intersecting a wall
- expansion intersecting a slip line
- shock/expansion of the same family
- shock/expansion of different families

In addition, shocks or expansions will be created at any geometry change of slope.

Each of the intersection types has its own unique downstream flow and thus a separate module for each one is required. When one of the above intersections is detected, the proper routine is called to create the downstream information. In addition, waves will be created by a change in the geometry of the side walls or the struts.

While it seems that we are getting into a rather complex set of logic, it must be remembered that we are taking advantage of the fact the properties between waves are constant and hence large portions of the volume need not be computed. We are now ready to describe the marching procedure and then will describe the coding we have created for this task to date.

The procedure starts by computing the oblique shocks from each of the side walls. This is done by a routine called SHANG (SHock ANGle - named in honor of Joe Shang at Wright-Patterson Air Force Base) which provides the shock angles and downstream flow information. Next the location of the intersection of these two waves is computed from the geometry to determine if this occurs before encountering any struts. If so, the routine INSECT (written by a our graduate student - presumably in honor of Bob Roach who wrote SHANG, PM, and the cone flow programs) is called to determine the downstream quantities and slip line. This intersection is only the second station necessary since properties in between the inlet and this station are fixed as either freestream (inlet conditions) or those behind one of the shocks. If a strut occurs prior to the intersection, then the second station is the location of the leading edge of the strut. At this point, two more oblique shocks are created. Again, SHANG determines their direction of propagation and downstream properties.

The next streamwise station is where the next event occurs. This will either be an intersection of some kind or a change in side wall or strut geometry. Hence, it can now be seen that the general procedure is to march downstream determining the next station by the location of the next event. The next intersection is determined by knowing the locations and directions of travel of each of the waves and the walls and struts. In between these locations, properties are constant. The next streamwise station is found by simply determining the next intersection of any of the types listed above. This proceeds until the combustion chamber is reached.

Analysis of the Flowfield Events

Once an event is detected, the proper routine is called to compute the changes which take place at the intersection. Several specialty routines were written specifically for these events and are described below.

1. Shock/Shock Intersection, Different Families (Subroutine SSDF)

When shocks of two different families intersect, they essentially travel through one another with an interaction that depends on the relative strengths of the two waves. Downstream, two new regions are formed with the determining condition that a slip line exists between them as shown in Fig. 10.

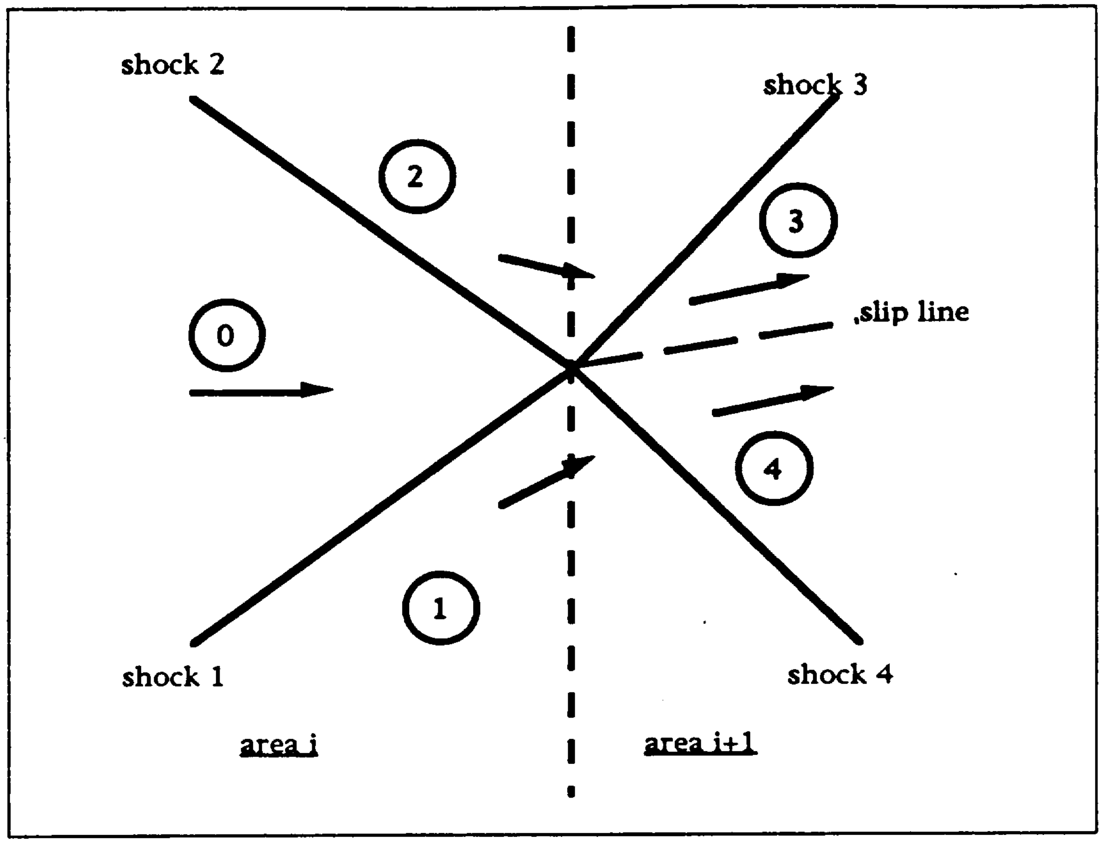


Figure 10. Intersection of two shocks of different families.

The flowfield in fields 0, 1, and 2 are already known with the flow in regions 3 and 4 to be determined. The slip line between the two regions requires that the velocities in the two directions must be parallel and that the pressure across the slip line be continuous. The remaining task is iterating the two shock wave angles q_3 and q_4. Using the oblique shock relations and marching from 1 to 4 and 2 to 3, respectively, the condition of pressure equality yields

$$\sin^2\theta_3 - \frac{p_1}{p_2}\left(\frac{M_1}{M_2}\right)^2 \sin^2\theta_4 - \left(1-\frac{p_1}{p_2}\right)\frac{\gamma-1}{2\gamma M_2^2} = 0$$

and the condition of equal flow angles gives:

$$\tan^{-1}\left(\frac{2\cot\theta_3\left(M_2^2\sin^2\theta_3-1\right)}{2+M_2^2\left(\gamma+1-2\sin^2\theta_3\right)}\right)+\delta_2+\tan^{-1}\left(\frac{2\cot\theta_4\left(M_1^2\sin^2\theta_4-1\right)}{2+M_1^2\left(\gamma+1-2\sin^2\theta_4\right)}\right)-\delta_1=0$$

These are two transcendental equations for the unknown shock wave angles q_3 (shock 3) and q_4 (shock 4) which are solved iteratively by a two-variable Newton iteration method. With the shock wave angles calculated, the flowfields in 3 and 4 can be determined out of the regular oblique shock relations.

2. Shock / Shock Intersection, Same Family (Subroutine SSSF)

If two shocks of the same family cross then downstream there are two more shocks. If the angle between them is relatively narrow, it is possible that the intersection may result in the formation of a single shock wave which is illustrated in Fig. 11.

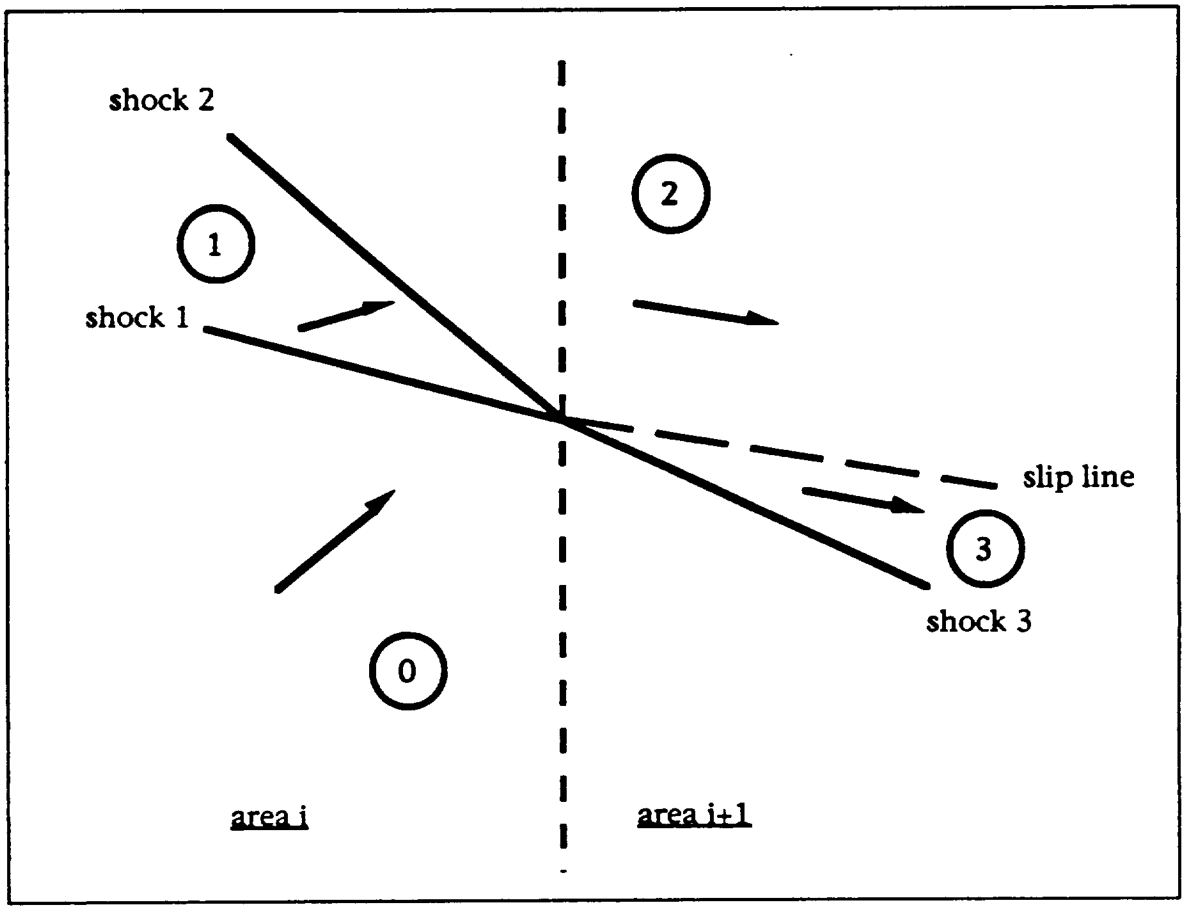


Figure 11. Intersection of Shock waves of same family

For this event the flow conditions in **0**, **1** and **2** are already known, such that only field **3** has to be determined. Given the flow angle in **2** and using the condition of equal flow direction in **2** and **3**, the flow can be directly calculated out of region **0** by using the subroutine SHANG. Let the deflection angle from **0** to **3** be $d = | a_2 - a_0 |$, then SHANG iterates the shock wave angle q_3 (shock 3) and the flow conditions in **3**. This procedure can be used for both family I and family II waves.

3. Shock / Expansion Wave Intersection, Different Families

There are two possible ways in which shocks and expansions of different families may cross depending on their initial orientation. Fig. 12 shows the case where the expansion is a family I wave while the other is a family II.

Case A: expansion wave of family I / shock of family II (subroutine SEDFA)

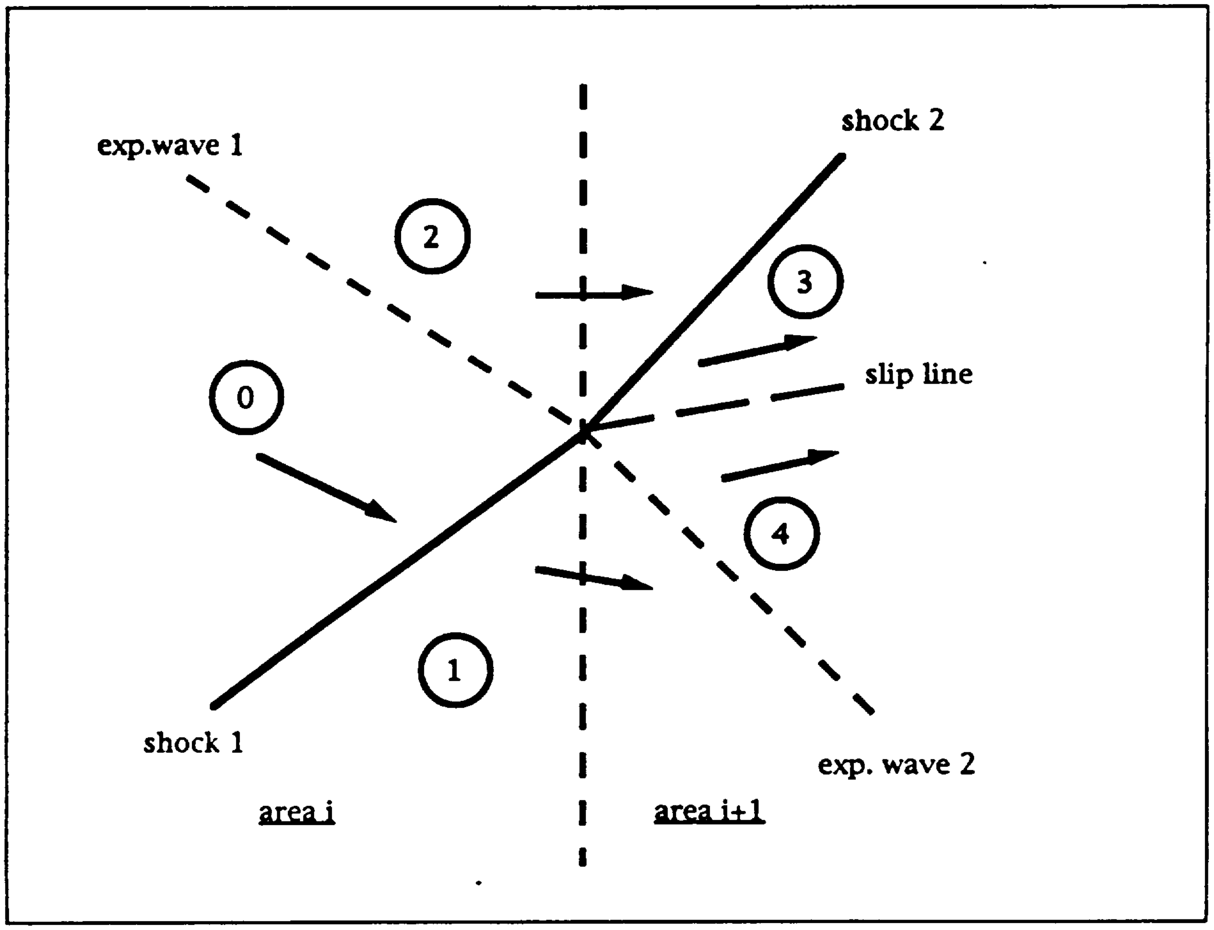


Figure 12. Intersection of Shock wave and expansion of different families.

Again, the flow conditions in 0, 1 and 2 are known whereas fields 3 and 4 have to be determined. As before this is done in an iterative fashion. The relevant equations are given below.

2 to 3: shock with deflection angle d = $a_3 - a_2 > 0$
pressure ratio:

$$\frac{p_3}{p_2} = 1 + \frac{2\gamma}{\gamma + 1}\left(M_2^2 \sin^2 \theta_3 - 1\right)$$

deflection angle:

$$\tan(\alpha_3 - \alpha_2) = \frac{2\cot\theta_3\left(M_2^2 \sin^2\theta_3 - 1\right)}{2 + M_2^2(\gamma + 1 - 2\sin^2\theta_3)}$$

These are functions of the Mach number M_2 (known) and the shock wave angle q_3 (unknown).

<u>1 to 4:</u> isentropic expansion with deflection angle d = a_4- a_1 > 0 and pressure ratio:

$$\frac{p_4}{p_1} = \left(\frac{1 + \frac{\gamma - 1}{2}M_4^2}{1 + \frac{\gamma - 1}{2}M_1^2}\right)^{-\frac{\gamma}{\gamma - 1}}$$

deflection angle:

$$\alpha_4 - \alpha_1 = \nu(M_4) - \nu(M_1)$$

where the Prandtl-Meyer function is defined as

$$\nu(M) = \sqrt{\frac{\gamma + 1}{\gamma - 1}}\tan^{-1}\sqrt{\frac{\gamma - 1}{\gamma + 1}(M^2 - 1)} - \tan^{-1}\sqrt{(M^2 - 1)}$$

These are functions of the Mach numbers M_1 (known) and M_4 (unknown). The condition of pressure equality in 3 and 4 (p_3=p_4) gives:

$$\frac{2\gamma M_2^2 \sin^2\theta_3 - (\gamma - 1)}{\gamma + 1} - \frac{p_1}{p_2}\left(\frac{1 + \frac{\gamma - 1}{2}M_4^2}{1 + \frac{\gamma - 1}{2}M_1^2}\right)^{-\frac{\gamma}{\gamma - 1}} = 0$$

The condition of the same flow direction (a_3=a_4) yields:

$$\tan^{-1}\left(\frac{2\cot\theta_3\left(M_2^2 \sin^2\theta_3 - 1\right)}{2 + M_2^2(\gamma + 1 - 2\sin^2\theta_3)}\right) + \alpha_2 - \nu(M_4) + \nu(M_1) - \alpha_1 = 0$$

Again, these two equations can be solved iteratively with the Newton- iteration to determine the two unknowns q_3 and M_4. Then using the given equations above, the conditions in 3 and 4 can be calculated.

Case B: shock wave of family I / expansion wave of family II (subroutine SEDFB)

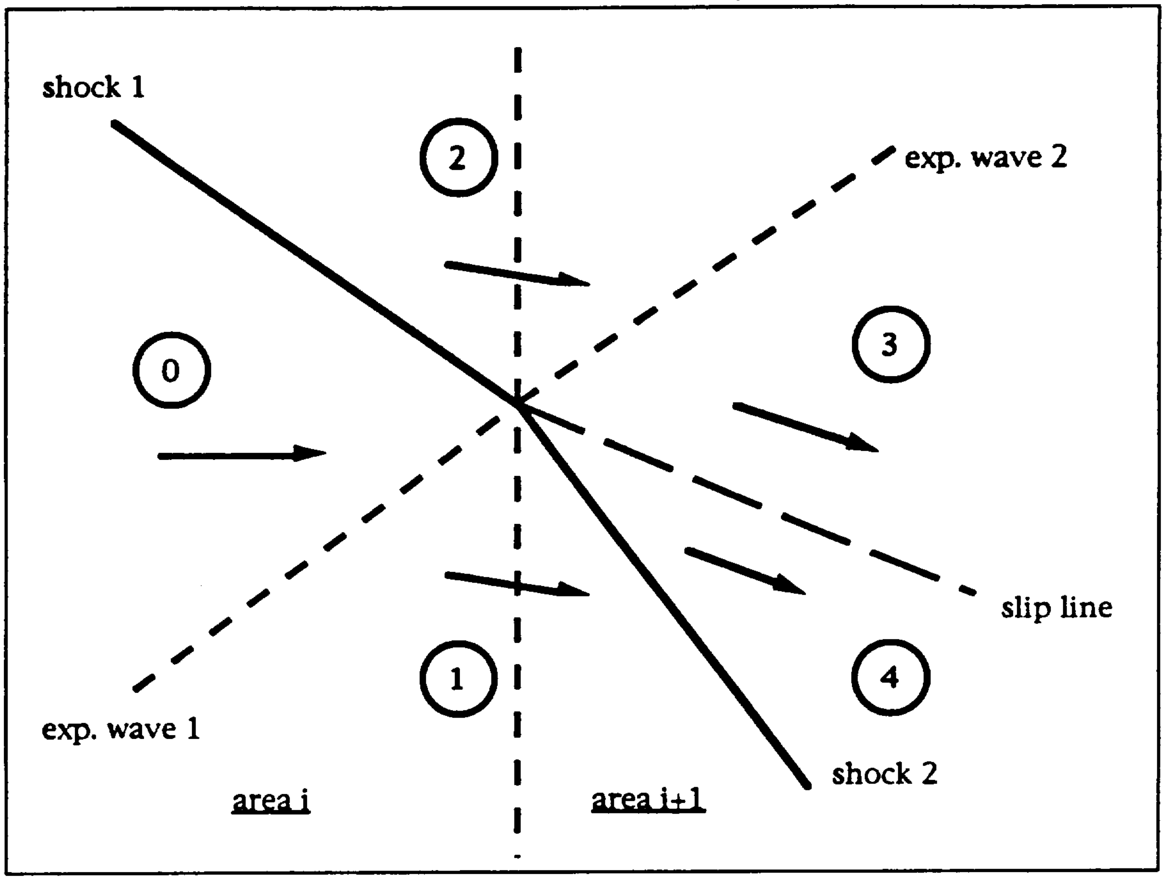


Figure 13. Intersection of a family I shock wave with a family II expansion.

2 to 3: expansion wave with deflection angle $d = a_3 - a_2 < 0$
1 to 4: shock wave with deflection angle $d = a_4 - a_1 < 0$

Since both times the deflection angle would be negative but the formulas given in Case A are only valid for positive deflection angles, it is better to set $d = a_2 - a_3$ and $d = a_1 - a_4$, respectively.
Otherwise the formulas of case A stay the same so that the condition of pressure equality in 3 and 4 gives:

$$\frac{2\gamma M_1^2 \sin^2\theta_4 - (\gamma - 1)}{\gamma + 1} - \frac{p_2}{p_1}\left(\frac{1 + \frac{\gamma - 1}{2}M_3^2}{1 + \frac{\gamma - 1}{2}M_2^2}\right)^{-\frac{\gamma}{\gamma - 1}} = 0$$

Similarly, the same flow direction yields:

$$-\tan^{-1}\left(\frac{2\cot\theta_4\left(M_1^2\sin^2\theta_4 - 1\right)}{2 + M_1^2(\gamma + 1 - 2\sin^2\theta_4)}\right) + \alpha_1 + \nu(M_3) - \nu(M_2) - \alpha_2 = 0$$

Again, these two equations can be solved iteratively for the unknown shock wave angle q_4 and Mach number M_3 and the conditions in 3 and 4 can be calculated.

4. Shock Wave / Expansion Wave Intersection, Same Family

There are again two cases, as shown in Fig. 14 and 15. In the first case, both waves are of family I. In the second they are of family II type.

Case A: both waves of family I (subroutine SESFA)

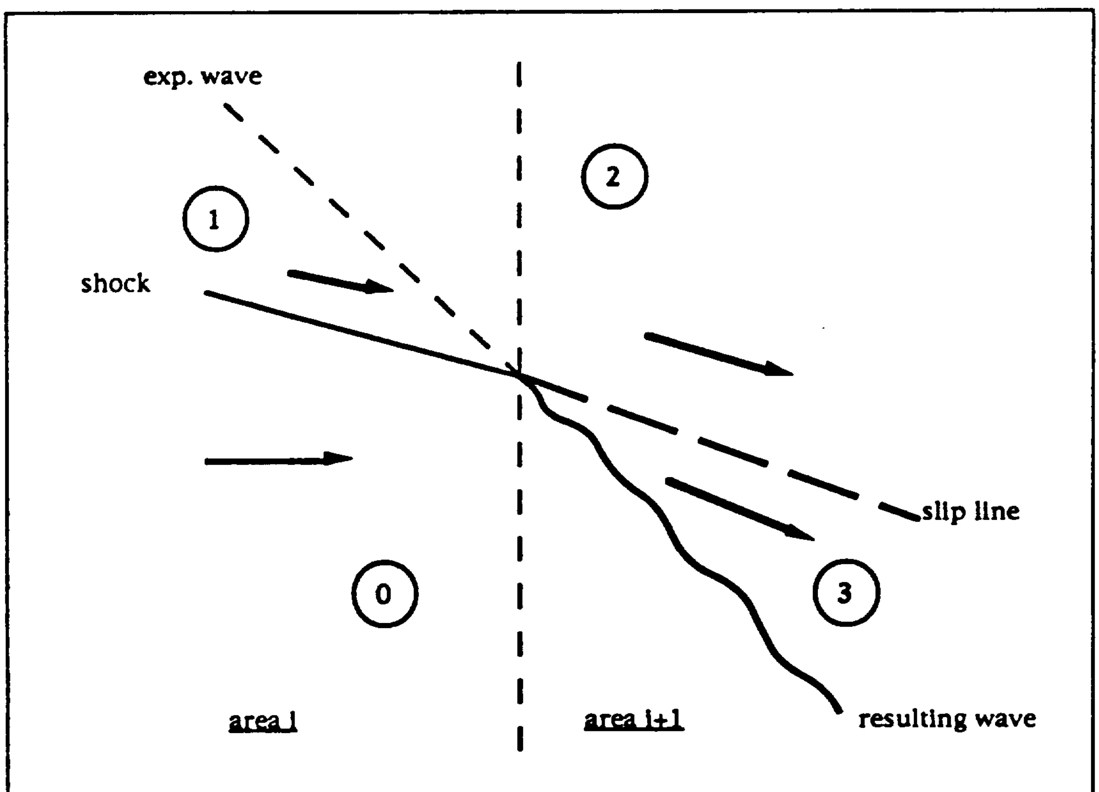


Figure 14. Intersection of shock and expansion of same family.

Since there has to be the same flow direction in 2 and 3:

a) if $(a_2-a_0) \le 0$: the resulting wave is a shock wave

By using the deflection angle $d = |a_2 - a_0|$ and the subroutine SHANG field 3 can be computed out of 2.

b) if $(a_2-a_0) \ge 0$: the resulting wave is an expansion wave

By using the deflection angle $d = a_2 - a_0$ and the subroutine PM field 3 can be computed out of 2. (Remark: The incoming shock and expansion waves can be interchanged since the conditions in 2 which represent the state behind both incoming waves and are given already determine the flowfield in 3.)

Case B: both waves of family II (subroutine SESFB)

This represents simply the mirror image of case A, such that:

a) if $(a_2-a_0) \le 0$: the resulting wave is an expansion wave (use PM)

b) if $(a_2-a_0) \ge 0$: the resulting wave is a shock wave (use SHANG)

5. Expansion Wave / Expansion Wave Intersection, Different Families, (Subroutine EEDF)

Here two expansion fans of different families intersect (Fig. 15).

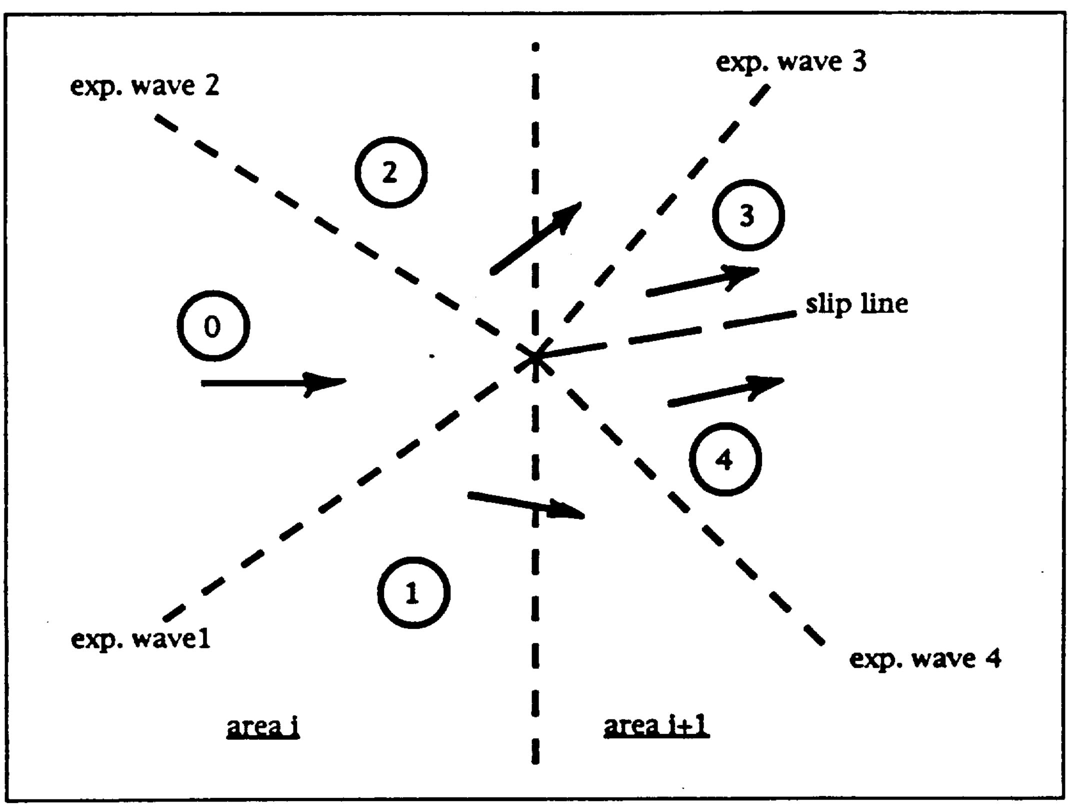


Figure 15. Intersection of expansion waves of different families.

Here there are two different isentropic expansions from 2 to 3 and 1 to 4, respectively. Using the already mentioned formulas, pressure equality in 3 and 4 gives

$$\left(\frac{1+\frac{\gamma-1}{2}M_3^2}{1+\frac{\gamma-1}{2}M_2^2}\right)^{-\frac{\gamma}{\gamma-1}} - \frac{p_1}{p_2}\left(\frac{1+\frac{\gamma-1}{2}M_4^2}{1+\frac{\gamma-1}{2}M_1^2}\right)^{-\frac{\gamma}{\gamma-1}} = 0$$

The same flow direction in both fields yields the equation:

$$\alpha_2 - v(M_3) + v(M_2) - \alpha_1 - v(M_4) + v(M_1) = 0$$

As before, iterative solving gives the unknown Mach numbers M_3 and M_4. Therefore, the conditions in 3 and 4 can be determined.

6. Expansion Wave / Expansion Wave Intersection, Same Family (Subroutine EESF)

As shown in Fig. 16, in this case thw two expansions are of the same family.

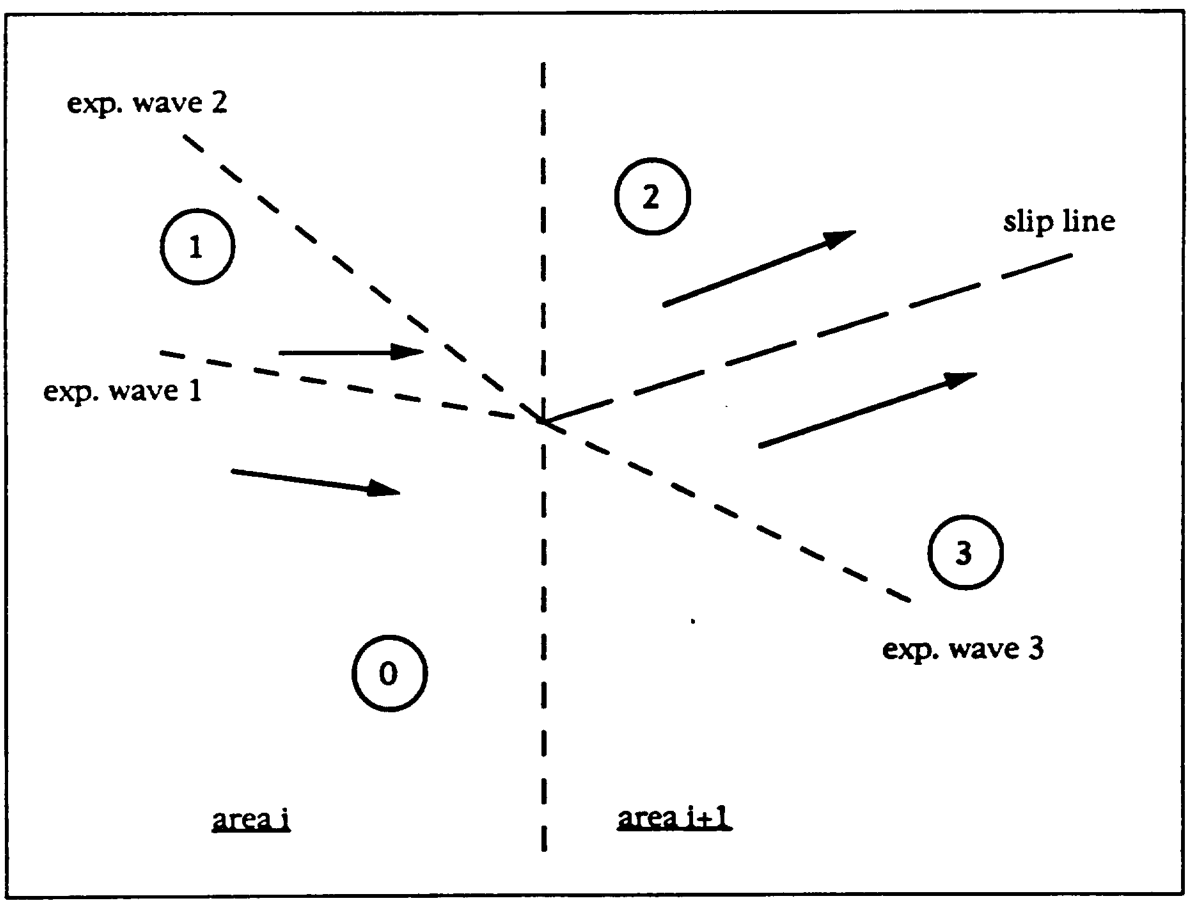


Figure 16. Intersection of two expansions of the same family.

Since the conditions in 2 are already known, the flowfield in 3 can be calculated by a simple Prandtl-Meyer expansion from 0 to 3 with deflection angle d = $|a_2 - a_0|$ using the subroutine PM.

7. Shock Reflection at Slip Line

Case A: shock of family I (subroutine SRSPA)

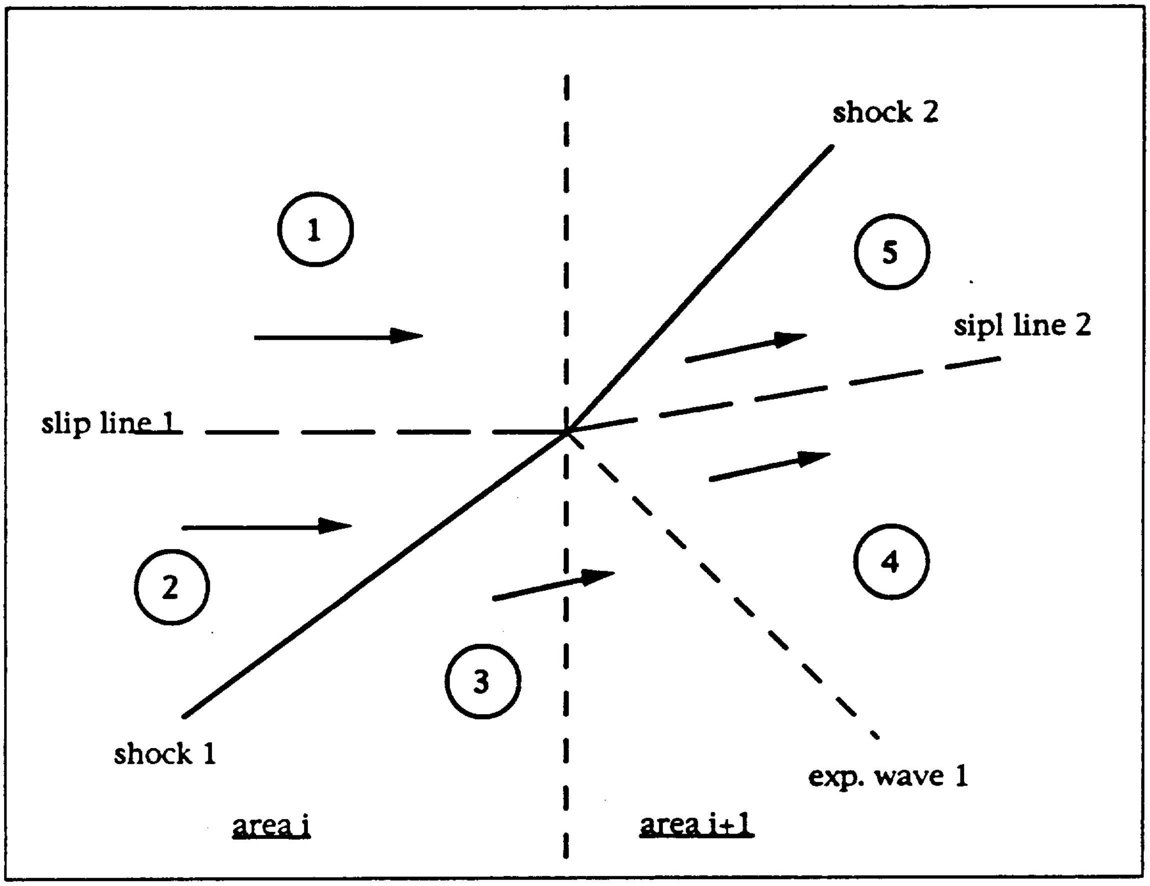


Figure 17. Reflection of shock from slip line in an unlike sense.

Since the air in field 1 is not at rest, the reflection of shock 1 and the slip line affects also this region and therefore not only shock 1 changes to expansion wave 1 but also a second shock (shock 2) is created to deflect the upper airflow. So the resulting flowfield in area i+1 looks like the one of the shock/ expansion wave intersection (event 3, case A) and will be treated in similar fashion.

Case B: shock of family II (subroutine SRSPB)

This is the mirror image of case A and for the same reason as above it will be treated like event 3, case B.

8. Shock Reflection at a Wall (Subroutine SRW)

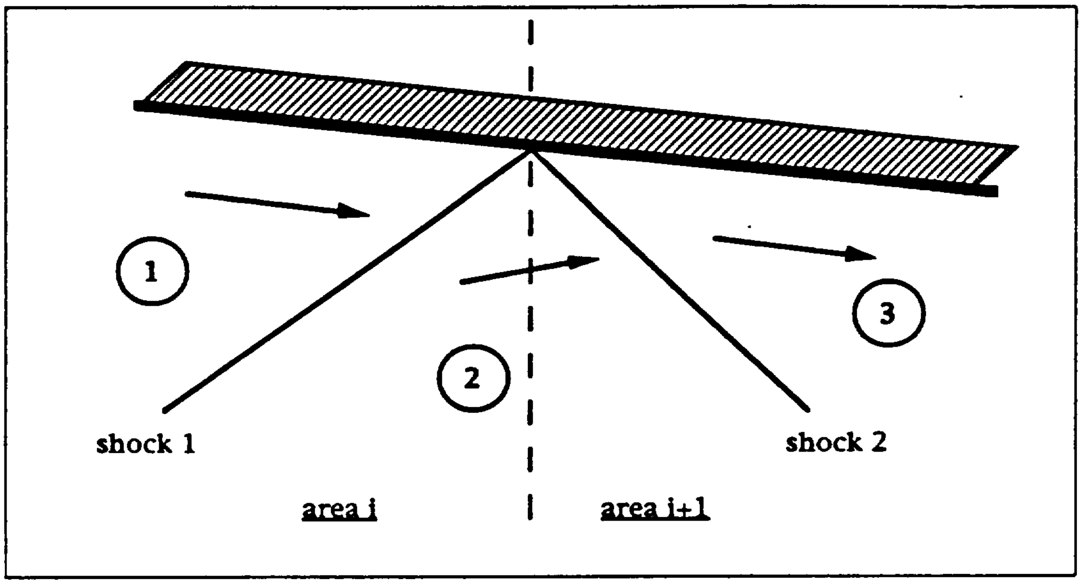


Figure 18. Reflection of shock from slip line in an unlike sense.

The condition for determining the flowfield in **3** is the flow angle along the wall, $a_3 = a_1$. Therefore, the deflection angle for the shock from **2** to **3** is $d = a_1 - a_2$ and SHANG can be used to calculate the conditions in **3** out of **2**.

9. Expansion Wave Reflection at slip line

Case A: expansion wave of family I (subroutine **ERSPA**)

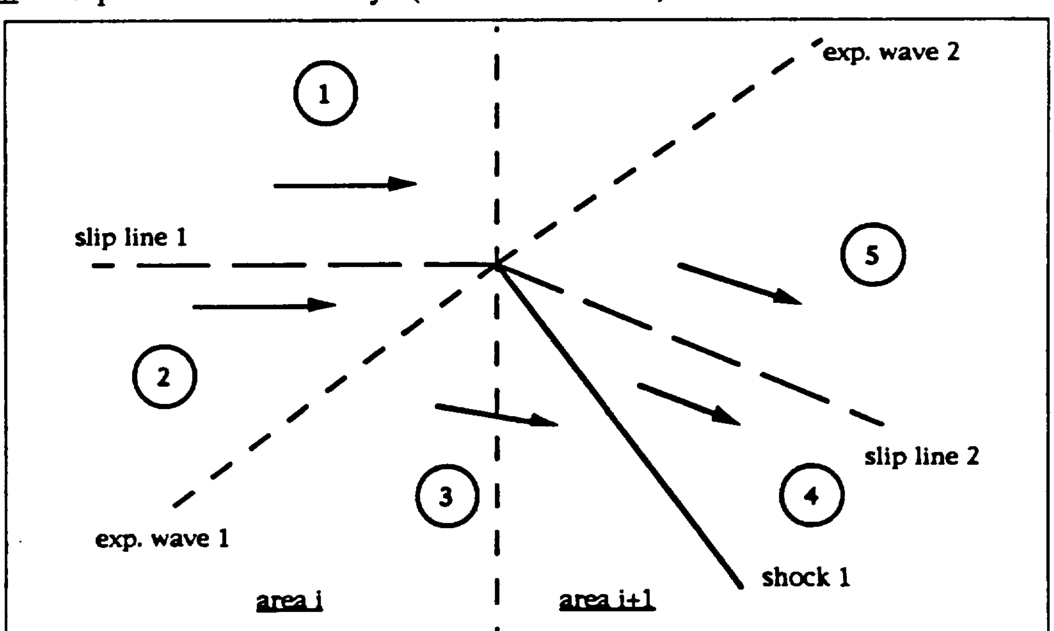


Figure 19. Expansion intersecting a slip line.

This case is similar to event 7 of a shock reflection at a slip line where field 1 above the slip line is also affected by the reflection and in addition to shock 1 a second expansion wave (exp. wave 2) is created. Here, the flowfield in area i+1 is identical with the one of the shock/expansion wave intersection of different families (event 3, case B).

Case B: expansion wave of family II (subroutine **ERSPB**)
For the same reason, this intersection can be treated like event 3, case A.

10. Expansion Wave Reflection at Solid Wall (Subroutine ERW)

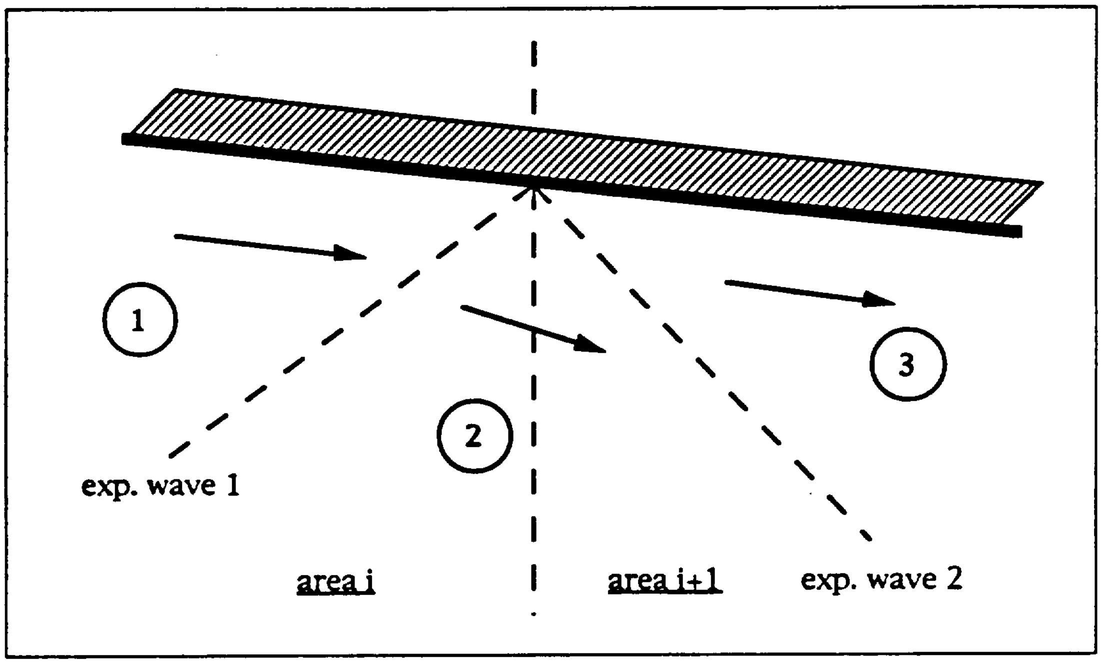


Figure 20. Reflection of expansion wave at a wall.

The condition for determining the flowfield in 3 is the flow angle along the wall, $a_3 = a$. Therefore, the deflection angle for the shock from 2 to 3 is $d = a_1 - a_2$ and PM can be used to calculate the conditions in 3 out of 2.

Some Auxiliary Routines used by Inlet

Subroutine SHANG

As mentioned, for each of the intersections, a different segment of code is called to determine the downstream quantities. There are, however, many similarities between each of the routines. All, for example make calls to the oblique shock routine, SHANG, and several make calls to the corresponding expansion routine PM (for Prandtl-Meyer). Listings of the SHANG an PM subroutines in FORTRAN are provided in the Appendices.

SHANG works by computing the shock wave angle from the oblique shock expression:

$$\tan\delta = \frac{2\left(M_1^2 \sin^2\theta - 1\right)\cot\theta}{2 + M_1^2\left(\gamma + 1 - 2\sin^2\theta\right)}$$

where δ is the wedge angle and θ is the shock angle with respect to the upstream flow direction.

The equation is transcendental in the shock angle and an iterative procedure must be used. Great care must be taken in the iterative procedure since the function is multivalued and badly behaved near the desired value. We use a Newton iteration procedure that is extremely fast. However, without taking a good initial guess Newton's method can easily diverge. We have prevented this by taking a rather good first guess by fitting a parabolic fit to the exact solution which is best in the region of greatest difficulty. The three pieces of information necessary to specify this parabola are known from the given information. These are the Mach wave angle (where the function has a vertical asymptote), the shock wave angle corresponding to the maximum wedge angle for an attached shock (function of Mach number only), and the fact that the oblique shock formula has a zero gradient with respect to the shock angle at this point.

With this initial condition we have yet to find a case where the procedure diverges. Convergence occurs to 5 decimal places in 4 to 5 iterations. Near the ill-behaved region, convergence is typically in 3 iterations.

Subroutine PM

Similarly, the PM routine provides a computation of the Prandtl-Meyer function which describes the isentropic expansion of a supersonic flow as it turns a corner away from its flow direction. In this case, the relationship is between the Mach number and the Prandtl-Meyer angle. The PM angle changes by the same amount that the flow turns and each PM angle has associated with it a unique Mach number. Thus PM works by evaluating the PM angle for the given input Mach number and adds the turning angle to the PM angle. The new Mach number is then determined from the new PM angle. Since the PM equation is transcendental in the Mach number, an iterative procedure is again required. We again use Newton iteration, but have no need to be careful of the initial guess since the PM function is well-behaved.

For the various intersection problems, most are a straightforward application of either SHANG or PM. The intersection of any wave with a wall or a slip line, for example is a direct computation using oblique shock or Prandtl-Meyer relations. Intersections of waves of the same family is another example. Only the intersection of waves of opposite families is not a direct computation because the equations involved are nonlinear. In this case an iterative solution is required. A popular method is to make an initial guess for the downstream flow direction. This defines the remaining variables in the two regions downstream of the intersection. The flow directions are forced to match by choosing the direction. The other requirement is that the pressures must also match. Comparing the computed pressures shows which way the next flow direction ought to be used.

Interface between Inlet & Combustor

The oblique shock/expansion method used in each of the streamtubes results in a number of distinct regions between shocks, slip lines and expansions. These regions must be combined to give an overall flow for the combustion zone. This is done by using the conservation laws applied to a one dimensional mixing of several flows. It turns out that there are two solutions, one for subsonic flow and one for supersonic flow. In the case of the scramjet assumed here we always take the supersonic solution. Later, when ramjets are used, the subsonic solution can be selected.

Combustion Zone Analysis

The flow field through the inlet up to the end of the struts computed by the above analysis provides the initial conditions for the combustor. The flow in each of the stream tubes enters the aft portion of the scramjet to mix and burn with the fuel. One now requires an estimate of the overall heat release, fuel required to effect the heat release, and the resulting change in gas properties at the end of the combustor. Although mixing and combustion is an extremely complex process and a vitally important part of the system performance calculations, a simple model based on Rayleigh line thermodynamics provides a useful approximation. An improved model is under development that incorporates appropriate estimates of mixing effects, chemistry, and distributed heat release.

The Rayleigh line equations expressing the changes in gas properties are

$$\begin{cases} P_4 = P_3\left(\dfrac{1+\gamma M_3^2}{1+\gamma M_4^2}\right) \\ T_4 = T_3\left(\dfrac{M_4}{M_3}\right)^2\left(\dfrac{1+\gamma M_3^2}{1+\gamma M_4^2}\right)^2 \\ P_{o4} = P_{o3}\left(\dfrac{1+\gamma M_3^2}{1+\gamma M_4^2}\right)\left(\dfrac{1+\dfrac{\gamma-1}{2}M_4^2}{1+\dfrac{\gamma-1}{2}M_3^2}\right)^{\frac{\gamma}{\gamma-1}} \\ T_{o4} = T_{o3}\left(\dfrac{M_4}{M_3}\right)^2\left(\dfrac{1+\gamma M_3^2}{1+\gamma M_4^2}\right)^2\left(\dfrac{1+\dfrac{\gamma-1}{2}M_4^2}{1+\dfrac{\gamma-1}{2}M_3^2}\right) \end{cases} .$$

The heat transferred in the combustion process must be determined by inserting appropriate combustor end conditions and then using the energy equation to determine the amount of propellant flow required to produce the desired end conditions. The latter is best expressed as the fuel mass fraction (ratio of fuel mass flow rate to air mass flow rate)

$$f = \frac{T_{o4} - T_{o3}}{\eta_b \frac{Q_R}{c_p} - T_{o4}}$$

where

$$Q_R = \text{Fuel Heating Value}$$

and

$$\eta_b = \text{Combustion Efficiency}$$

The corresponding equivalence ratio is

$$\phi = \text{Equivalence Ratio} = \frac{f}{f_{stoichiometric}}$$

where $f_{stoichiometric}$ is the stoichiometric air/fuel ratio.

Certain constraints must be addressed in the computation. In particular, one must not introduce so much heat flux that the channel is thermally choked. To avoid this, the exit Mach number is constrained to be above a limit value taken to be $M_4 = 1.15$ in the present version of the code (all such parameters are easily modified as necessary).

In running the program, the user may specify the equivalence ratio. This number may be larger than unity, expressing a fuel-rich environment with the unburned fuel assumed to be mixed with the combustion products at the combustor exit. Such mixing may be beneficial in some flight environments mainly because of the lower molecular weight of the mixture and the cooling effect of the unburned hydrogen. Combustion heat release is allowed up to the point that one of three conditions exists:

1. Maximum allowable channel static temperature is exceeded.
2. Soichiometric fuel/air ratio is achieved
3. Thermal choking is imminent ($M_4 < 1.15$)

If the selected value of ϕ does not result in any of these conditions, then the program determines the heat release and corresponding combustion streamtube exit conditions.

If equivalence ratio is not specified, the program computes a value representing the maximum that satisfies the thermal choking the three constraints. The algorithm first assumes $M_4 = 1.15$ and then checks to be sure that condition 1 is not invoked. A computed ϕ up to a value of unity is determined. If the user requires the fuel-rich case, then the program must be rerun with the desired value of ϕ. The program assumes that maximum allowable combustion chamber temperature is 5500 ^oR. This limit may be changed to better match the material property limitations and surface cooling details of the combustor.

If a fuel-rich condition exists at the end of the combustor, the program iteratively adjusts the exit gas properties to reflect the resultant change in molecular weight, gas constant, and specific heat ratio.

Aftbody Method of Characteristics

The flow in the aftbody or nozzle region, like the forebody, uses a rotational method of characteristics. The starting conditions for the aftbody MOC (Subroutine MOCA) are provided by the flow quantities at the exit plane of the combustor. In addition, the corners of the exhaust plane are locations of expansions caused by the body shape on the inner corner and the great difference in pressure at the outer corner. The expansion values are computed using isentropic and shock relations similar to those used in the inlet program. These are applied at the corners only. Propagation of the expansion is left up to the MOC.

An additional feature in the aftbody region is the presence of a slip line between the gas flow from the propulsion system and the external flow field. Thus provision must be made in the MOC procedure. The manner in which this is done is as follows. Two points are placed at a slip line. They can be coincident for convenience. One point is considered to be just on the body side of the slip line while the other is considered to be just on the shock side. The point on the body side is computed using the single characteristic line which is inclined toward the body surface. The point on the other side, uses the characteristic line inclined toward the shock. The additional information required comes from the requirements at a slip line. That is, the flow must be parallel and the pressure must be continuous across the slip line. Again, the system is transcendental, but a fast solution procedure was set up.

Other Routines used in the Computation

In addition to the main computational modules just described, it is necessary to devote special attention to the interfaces and to deal with special regions of the flow outside the main flow through the propulsion duct.

Cowl Lip: Flow external to the propulsion module is required since the exhaust region mixes with this flow downstream of the propulsion system. The flow in this external region is generally different from the freestream since it has already traversed the bow shock. In addition, there will usually be an expansion at the cowl lip which is computed here by simple expansion theory.

Interface at combustor exhaust with external flow: The flow through the propulsion system has had much energy added. As a consequence, its pressure has been greatly raised. But while it has been partially expanded in the aft portion of the combustor, much of its final expansion will take place in the aft body region. Thus the gas exiting the combustor is at a higher pressure than the external flow. Since the gas will thus undergo an expansion, the external flow will see a compression. This results in a combination of a shock wave, expansion waves, and a slip line at the outer corner of the combustor exhaust. These conditions are computed using oblique shock/expansion theory in an iterative fashion.

Combustor exit plane: For the rest of the combustor exit plane, the quantities at the end of each streamtube must be interpolated to align with the beginning of the characteristics net of the aft body region. This is done using parabolic curve fits.

Program Operation

The user interacts with the code (in its standard UNIX environment) through an input data file which contains all of the freestream quantities, fuel characteristics and flow rate, and vehicle/propulsion system geometric parameters. The user is thus free to specify completely different freestream quantities such as Mach number and angle of attack as well as completely different vehicle geometries through the data file. Example input files are shown in Appendix 2. A standalone Apple Macintosh version of the program was also written enabling users to run the program on small desktop machines. Operation of the program is similar to the standard version except that results are read into data files for later graphing operations and extensive output is presented on the screen. This approach will be developed further with a complete graphical interface to simplify modification of parameters and to enable interactive study of variations in those parameters in an animated presentation mode.

The user never needs to make any changes to the main program itself, though there are a few places in the code itself where some parameters can be adjusted for accuracy or for simplicity. These include such things as the location on the forebody where the coneflow solution is to be applied to start the forebody method of characteristics and the number of vertical stations in the fore and aftbody regions. These may never have to be adjusted as reasonable values are currently used, but may be if desired.

It is important to point out however that there are severe limitations to the validity of the present code with the MOC forebody analysis when applied to geometries different than the one shown in Figure 1. Compression waves on the underside of the forebody are not accommodated in the computations. A second version of the program (Macintosh format) was developed to treat such problems using shock/expansion techniques until a shock-embedding scheme is implemented in the forebody code. It was utilized in the preliminary UX-30 computations to be described in the next section.

Program Output

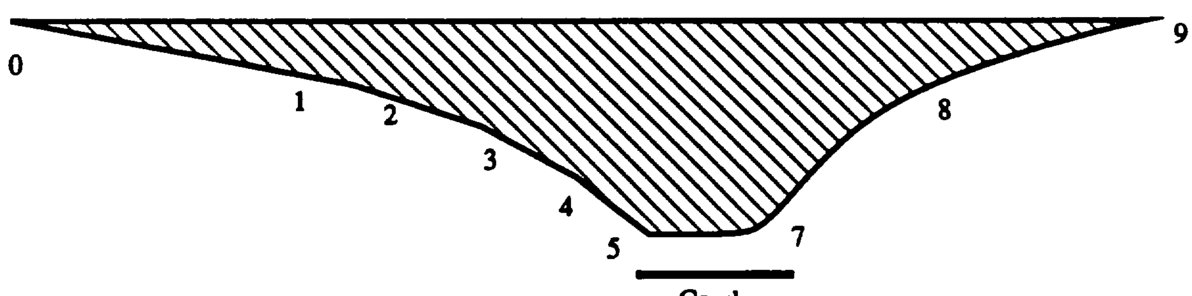
The complete solution of the flow on the underside of the vehicle is available as an output option. Contour plots of sundry flow variables can then be attained to study the character of the hypersonic flow in and around the geometry. The flow data may be put into files which can be used by such data reduction programs as PLOT3D. In addition, the net forces and moments are computed from a knowledge of the net changes in the mass and momentum fluxes and pressures on a control volume surrounding the underside flow region.

RESULTS

In what follows are presented some initial computations utilizing the algorithms just described. This section is broken up into two parts describing results for the two test vehicle geometries. The inpurt parameters are shown in the standard input file format in Appendix 2. A variety of flight conditions and vehicle attitudes were studied to identify any program bugs and to enable initial assessments of the validity of the various computational modules.

Results for NASA UX-30 Configuration

A generic configuration shown in Fig. 21 was used to represent the (Two-D) vehicle geometry. The propulsion duct was assumed to span the full width of the vehicle (33.33 ft). The table shows the coordinate points identified in the figure. In addition to the points, nozzle entrance and exit angles were specified as 25 and 5 degrees respectively. The nozzle contour was then represented by a fourth-order polynomial from which a set of seven points were computed to use in the UX-30 data file (see Appendix 2).



Body Coordinates

	x (inches)	y (inches)
1	0	0
2	640.4	56
3	844.7	83.8
4	954.9	104.9
5	1114	145.4
6	1114	154.5
7	1202.4	145.4
8	1501.2	44.6
9	1800	0

Notes
x is measured aft from nose station; y is measured down from nose station
Cowl lip is assumed to be directly below point 5 in this calculation
No rearward cowl overhang is represented in these preliminary results

Figure 21. Definition of UX-30 Configuration

The computer program was run with conditions representing flight at 95,636 ft. Mach number and angle of attack were varied. Table 1. shows typical input sequence and output from the computer program (Macintosh version: **UX-30**) for an angle of attack of $\alpha = -3$ degrees and flight Mach number $M = 10$.

Theresults show that performance is consistent with publishedresults showing a fuel specific impulse of about 1500 sec. The thrust vector points downward relative to the flight direction. This is the mainly theresult of neglect of boundary layer offsets in the nozzle flow. A large pitchup moment accompanies the downward pointing thrust vector. The line of action of the thrust vector crosses the vehicle reference (x-) axis at a point approximately one-third of the body length from the nose. Pitch moment about the mass center is readily determined from this information.\

Thrust and moment coefficients are also computed. These are defined as

$$C_T = \frac{T}{q_\infty A_{inlet}}, \qquad C_M = \frac{M}{q_\infty A_{inlet} L}$$

where T is the thrust vector magnitude, M is the pitching moment about the origin of the vehicle coordinates, and L is vehicle length. The scramjet inlet area is used as the reference area.

A quick check of the operation of the complete program is provided by printing of the average values of Mach number, velocity,and both static and total pressures and temperature at each main station along the vehicle underbody. For the selected equivalence ratio ($\phi = 0.671$), there was insufficient combustion heat release to approach the limiting exit Mach number and maximum temperature constraints. Thus, a larger fuel flow rate (higher throttle setting) is available for the flight conditions and vehicle attitude represented.

The program operation was checked further by running sweeps over a fairly comprehensive Mach number and angle of attack space. The vehicle did not operate well in scramjet mode at Mach numbers below about 6.5 as evidenced by the failure of the program to achieve converged nozzle flow. Operation appeared normal over a range of Mach numbers up to about 19. Again, difficulties with nozzle MOC convergence prevented useful output above that Mach number. As higher Mach numbers were reached, the available angle of attack range grows narrower. At Mach 19, the system would not operate outside an angle of attack range ($0 < \alpha < 1.5$ degrees). Reasons for these results may be either actual physical limitations of the chosen configuration or may be due to strictly numerical difficulties. The former seems to be the correct explanation, since the version of the program written to address the generic NASP geometry showed similar behavior over a different range of parameters.

Figures 23-29 show plots of basic performance output as functions of vehicle angle of attack. Curves are plotted for constant Mach number in the stated range. The relatively smooth variation with attitude and Mach number suggests that the code is working correctly at least from the point of view of internal consistency. The plots display systematic changes from one flight condition to the next that are consistent with physical limitations. The significant downward thrust angle emphasizes the need to incorporate boundary layer effects in both the nozzle and theforebody compression ramps. Also, the large adverse pressure gradient at the end of the nozzle would

TABLE 1

Typical Program Input/Output Sequence

Input Sequence:

Enter Desired Equivalence Ratio (ERmin):	0.671
Enter Mach Number Range (Mstart,Mfinal,DeltaM):	10,10,1
Enter Angle of Attack Range (Astart,Afinal,DeltaA):	-3,-3,1
Enter Plot File Name [type NONE if none reqd.]:	NONE
Enter Input data file name:	UX-30.dat

SCRAMJET THRUST VECTOR ANALYSIS

Flight Conditions:

Minf =10.00 Alpha =-3.00°
Tinf = 406.44 °R Toinf = 8535. °R Pinf = 27.830 psf
Dynamic Pressure = q = 1948.100 psf

Vehicle Geometry:

Total Length = 149.79 ft
Module Inlet Area = 25.23 ft^2
Ai/Acapture =0.224

System Performance:

Ct =4.439 Cm =1.397
ISPfuel = 1576.86 sec
Effective Equivalence Ratio = 0.671
Air Mass Flow Rate = 49.28 slug/sec
Fuel Mass Flow Rate = 0.96 slug/sec
Fx = 48908.7 lbf Fy =-212628.1 lbf
Thrust = 218180.6 lbf
Thrust Angle = -77.05°
Moment about Origin = 10284961 ft-lbf
Thrust line-of-action = 0.327

Average Flow Properties in Duct:

Station	M	V [ft/sec]	T [°R]	To [°R]	p [psf]	Po [psf]
Free Stream :	10.00	10942.	406.4	8535.	27.8	1181084.
Duct Inlet Plane:	6.94	9881.	801.9	8535.	277.3	1091118.
Combustor Inlet :	6.95	9638.	801.5	8535.	276.8	1090910.
Combustor Exit :	3.37	8759.	4494.8	11141.	293.4	23866.
Nozzle Entrance :	3.37	10512.	4496.5	11141.	293.9	23866.
Nozzle Exit :	6.15	11387.	1583.3	9378.	8.0	44183.

quite likely lead to a separated flow and major changes to the flow field in the nozzle. These would be expected to greatly modify the thrust vector angle.

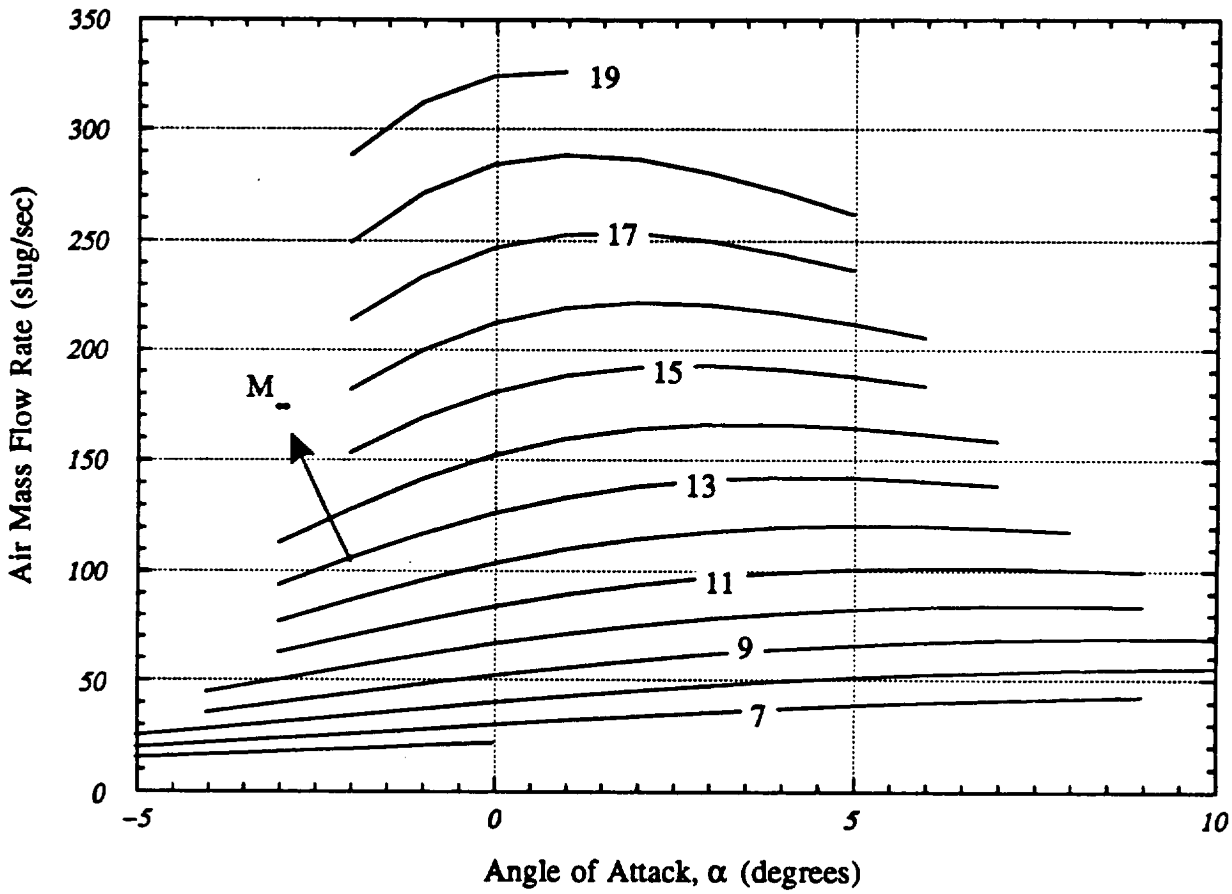


Figure 22. Air Mass Flow Rate vs Angle of Attack for UX-30 Vehicle

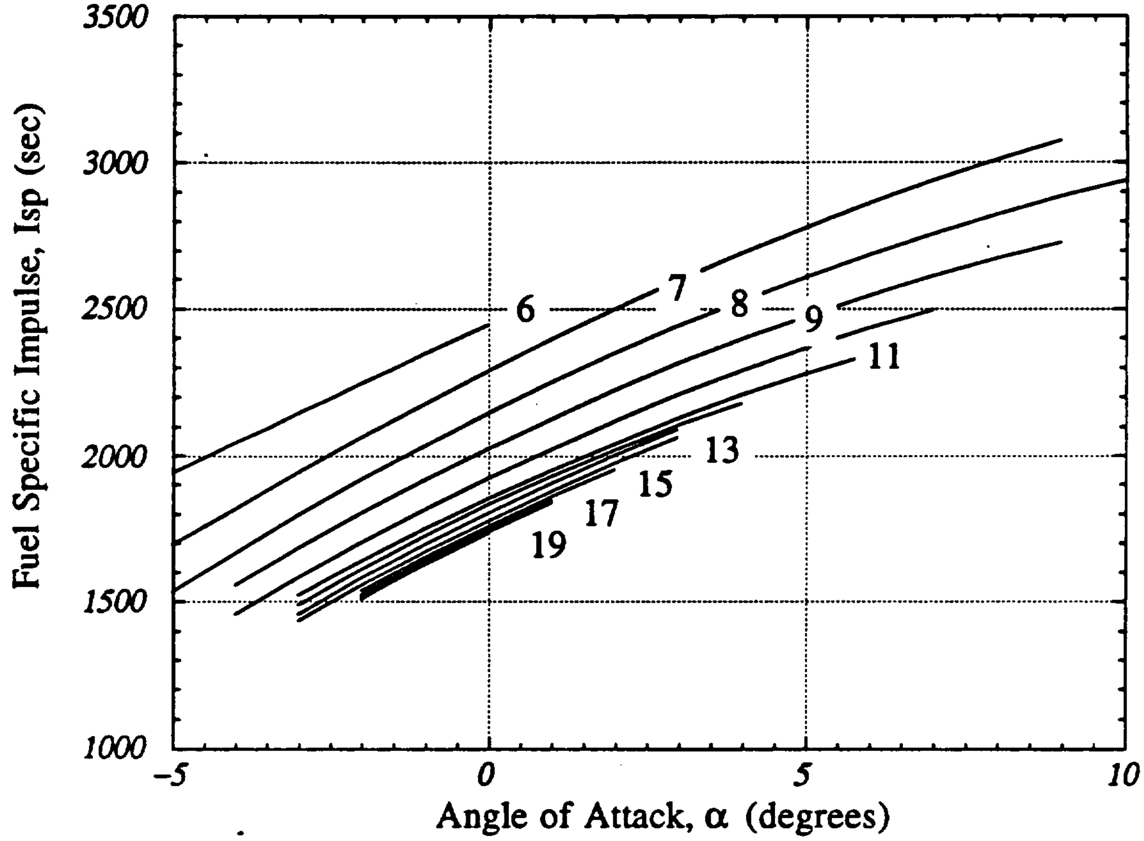


Figure 23. Fuel Specific Impulse vs Angle of Attack for UX-30 Vehicle

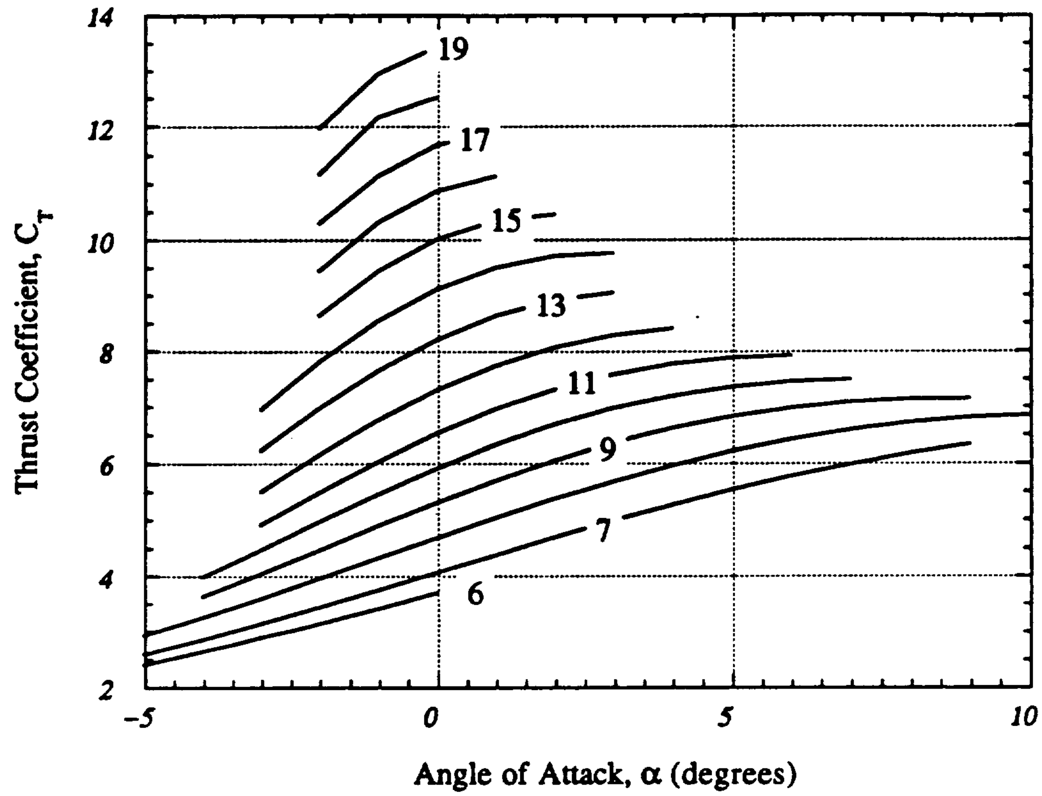


Figure 24. Thrust Coefficient vs Angle of Attack for UX-30 Vehicle

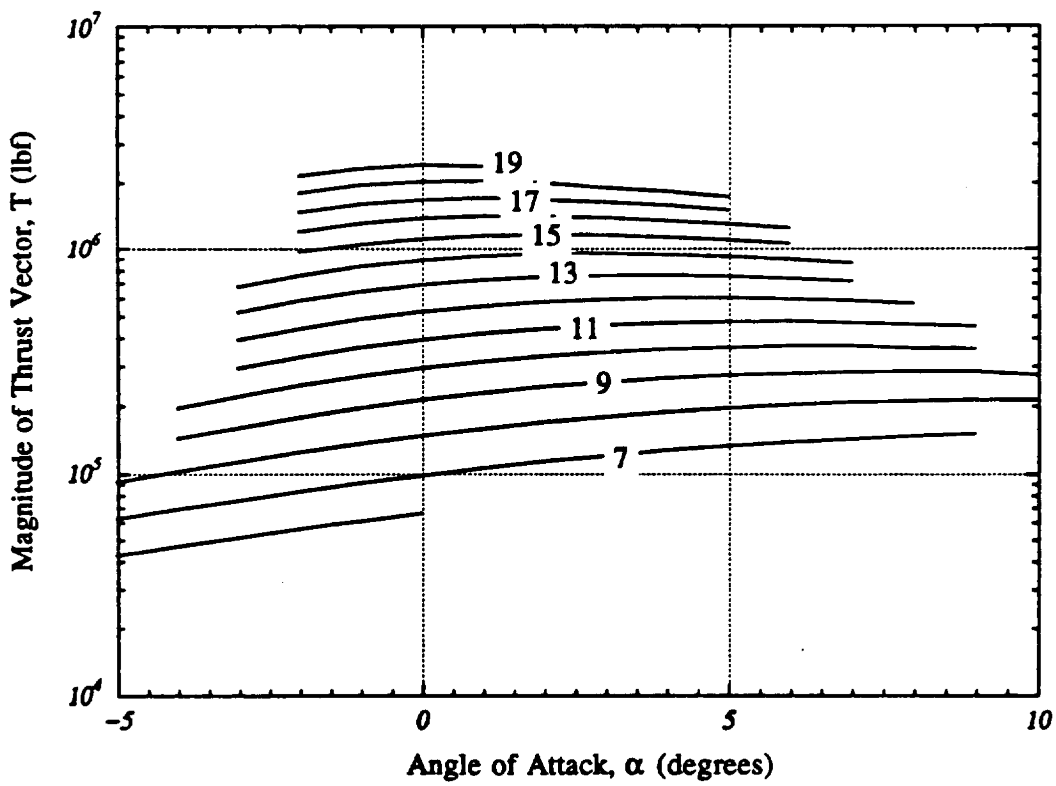


Figure 25. Thrust Vector Magnitude vs Angle of Attack for UX-30 Vehicle

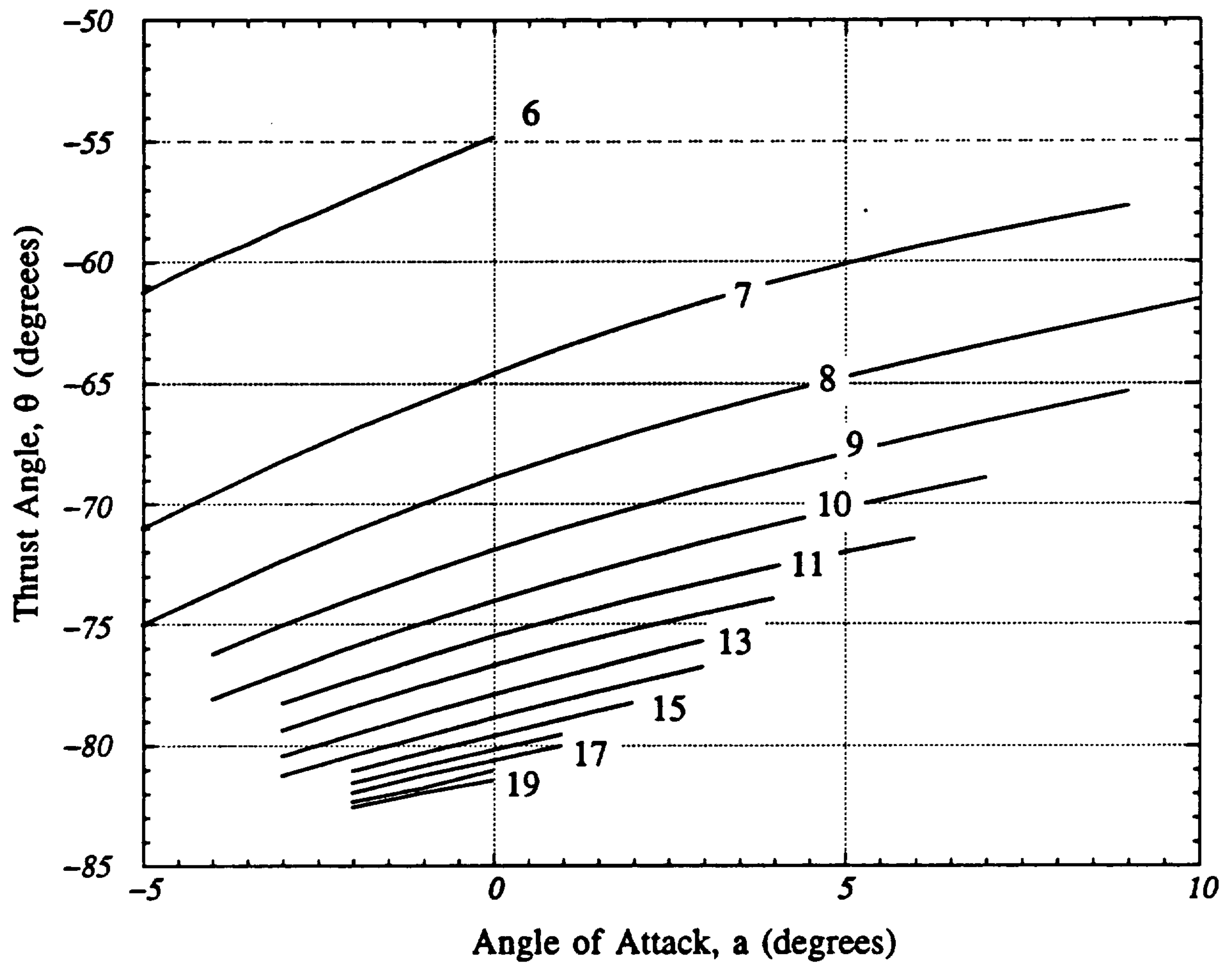


Figure 26. Thrust Angle vs Angle of Attack for UX-30 Vehicle

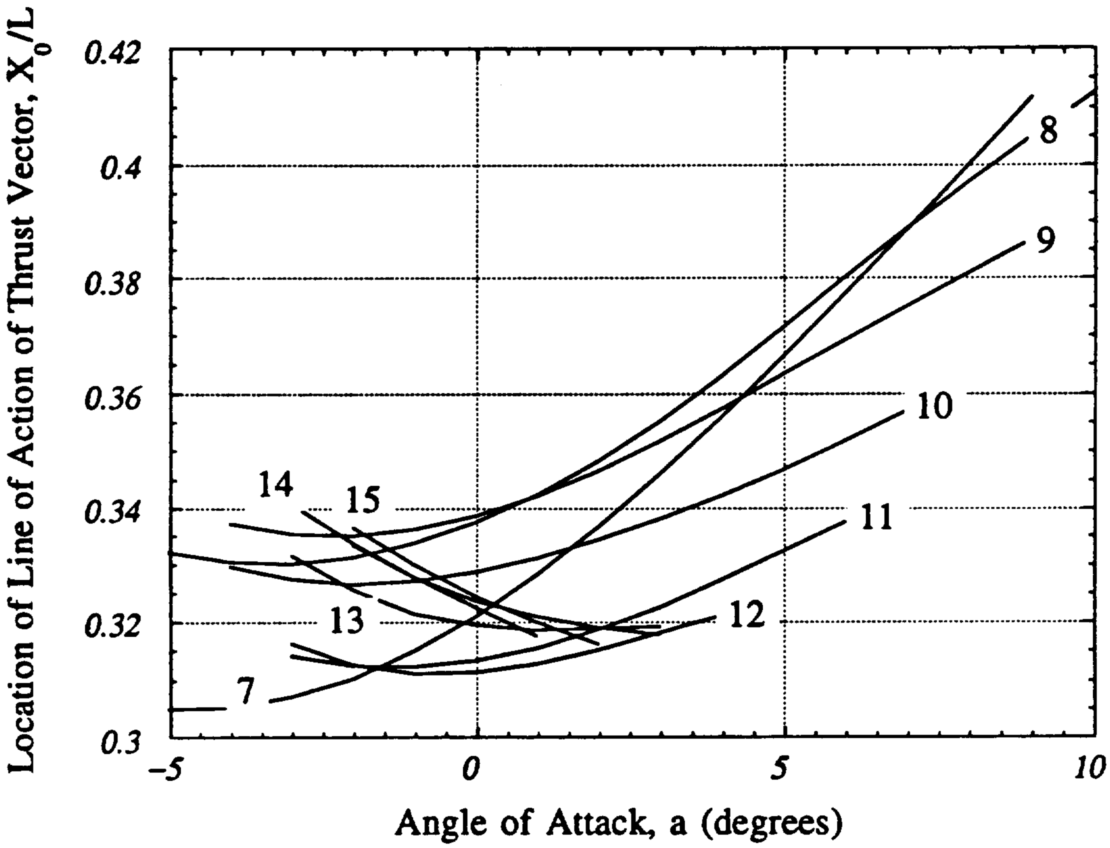


Figure 27. Location of Line of Action of Thrust Vector vs Angle of Attack for UX-30 Vehicle

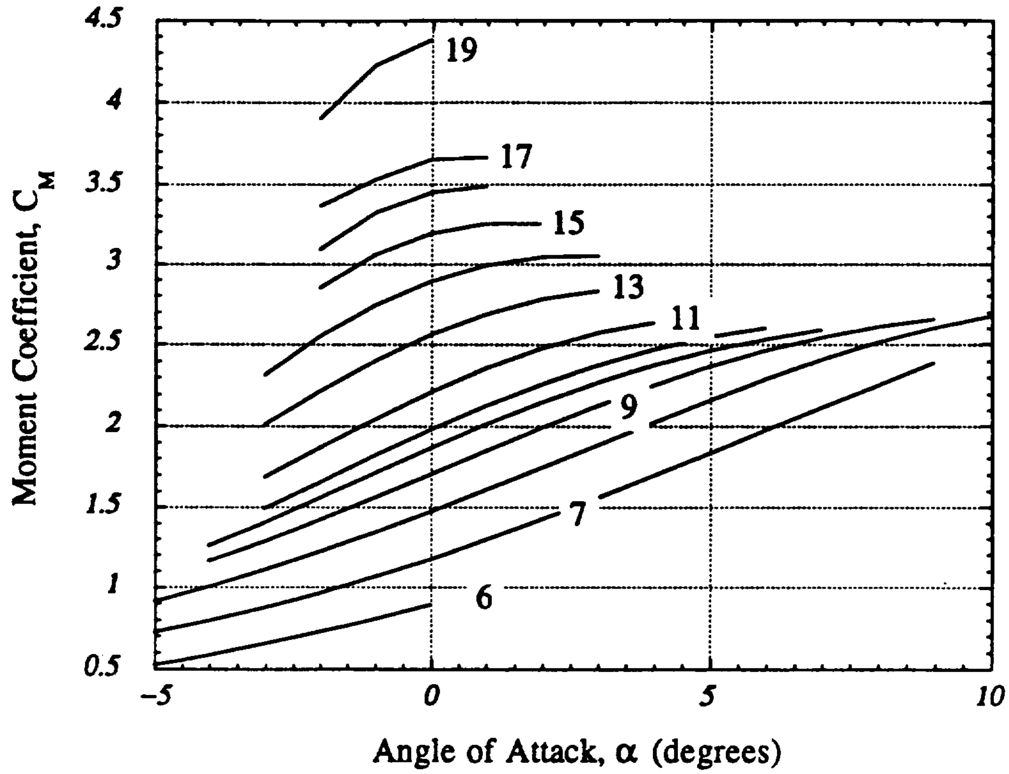


Figure 28. Thrust Pitching Moment Coefficient vs Angle of Attack for UX-30 Vehicle

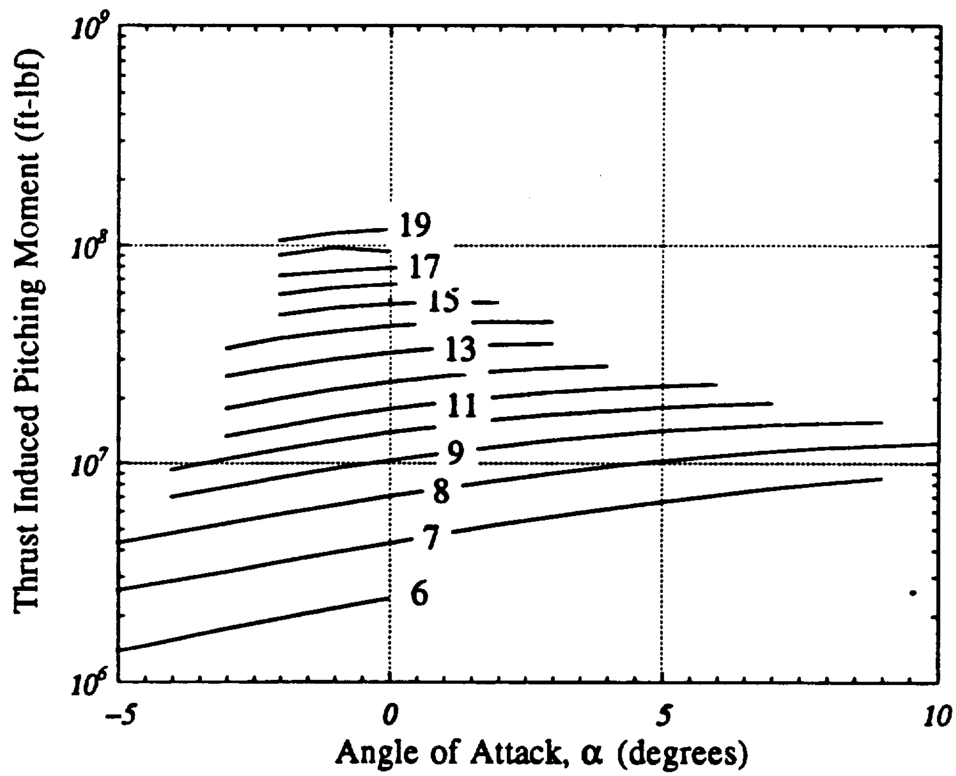


Figure 29. Pitching Moment vs Angle of Attack for UX-30 Vehicle

Results for Generic NASP Configuration

A geometry obtained by scaling dimensions from the NASP-like vehicle described in Aviation Week (AWST) was used in the following computations. The information in Appendix 2 is essentially the data file used to run the program. Figures 30-36 are constant Mach number contours of air mass flow rate, specific impulse, equivalence ratio, thrust coefficient, thrust, thrust angle, thrust line of action, moment coefficient, and pitching moment with angle of attack as the main parameter. The trends are similar to those for the UX-30 vehicle. However there are very significant performance differences showing the great sensitivity to seemingly small geometry variations. Flight conditions illustrated correspond to an altitude of 100,000 ft. An equivalence ratio of 1.1 was used for the calculations. The solid curves are for Mach numbers up to 7. At M = 7, the best performance was achieved as measured by fuel specific impulse. A vehicle length of 200 ft was assumed. The calculations are for a single engine module that has a channel width approximately one-fourth the vehicle width. Dashed curves are for an apparently different family of behavior for Mach numbers greater than 7.

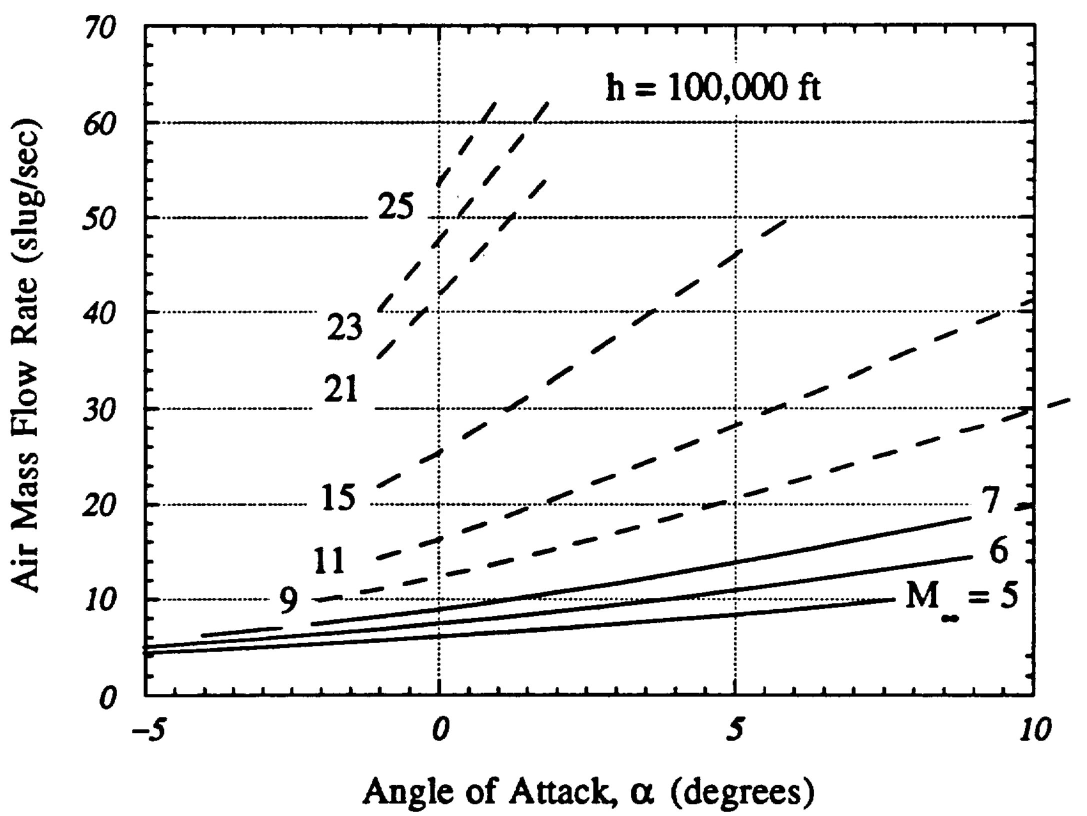


Figure 30. Mass Flow Rate vs Angle of Attack for AWST Vehicle

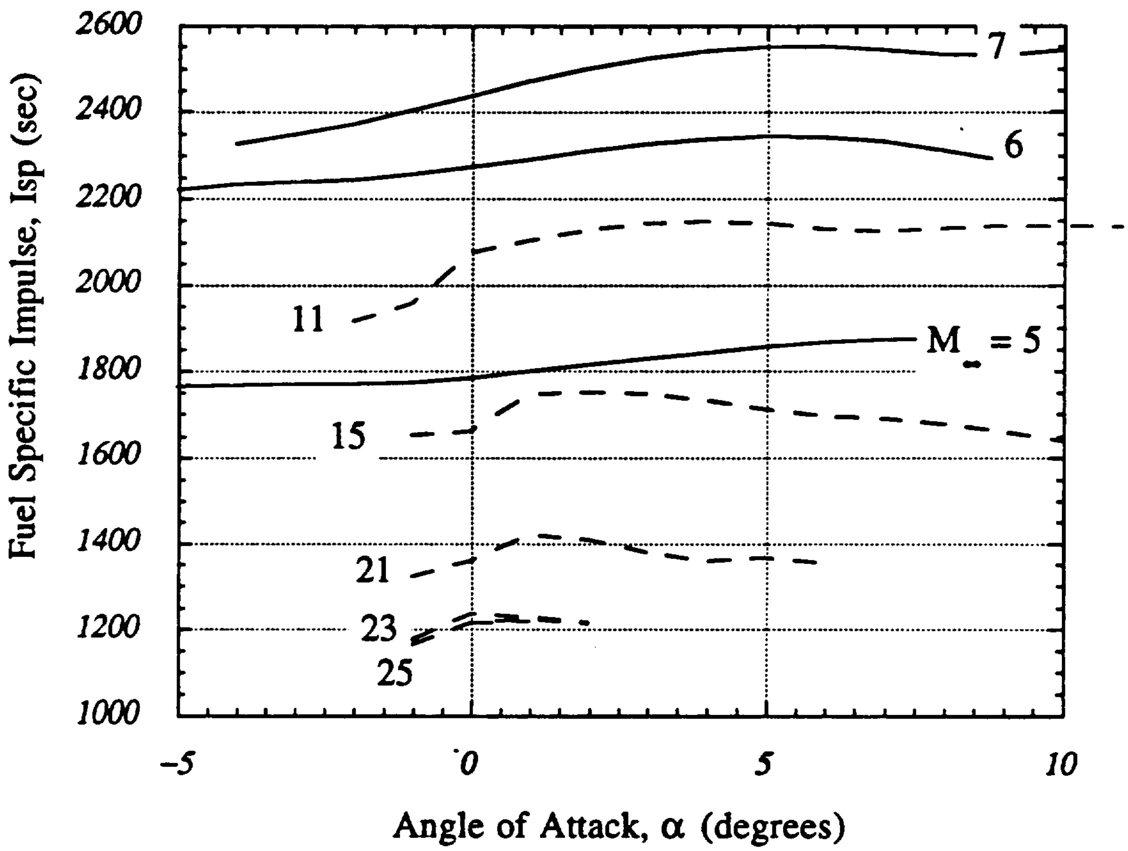


Figure 31. Specific Impulse vs Angle of Attack for AWST Vehicle

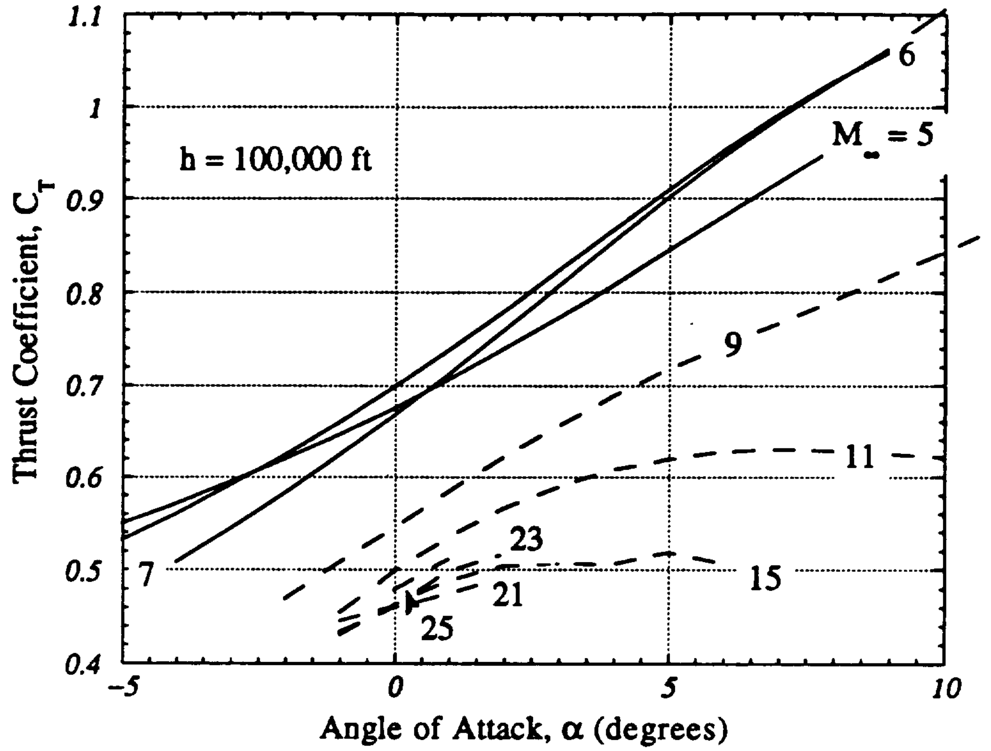


Figure 32. Thrust Coefficient vs Angle of Attack for AWST Vehicle

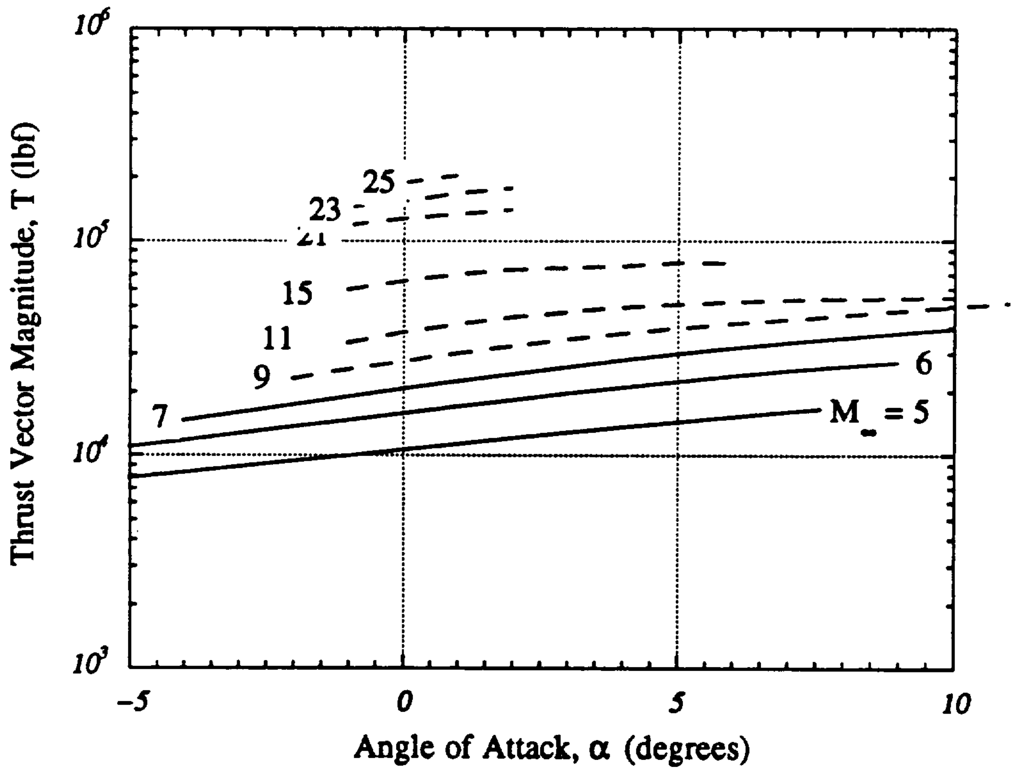


Figure 33. Thrust Vector Magnitude vs Angle of Attack for AWST Vehicle

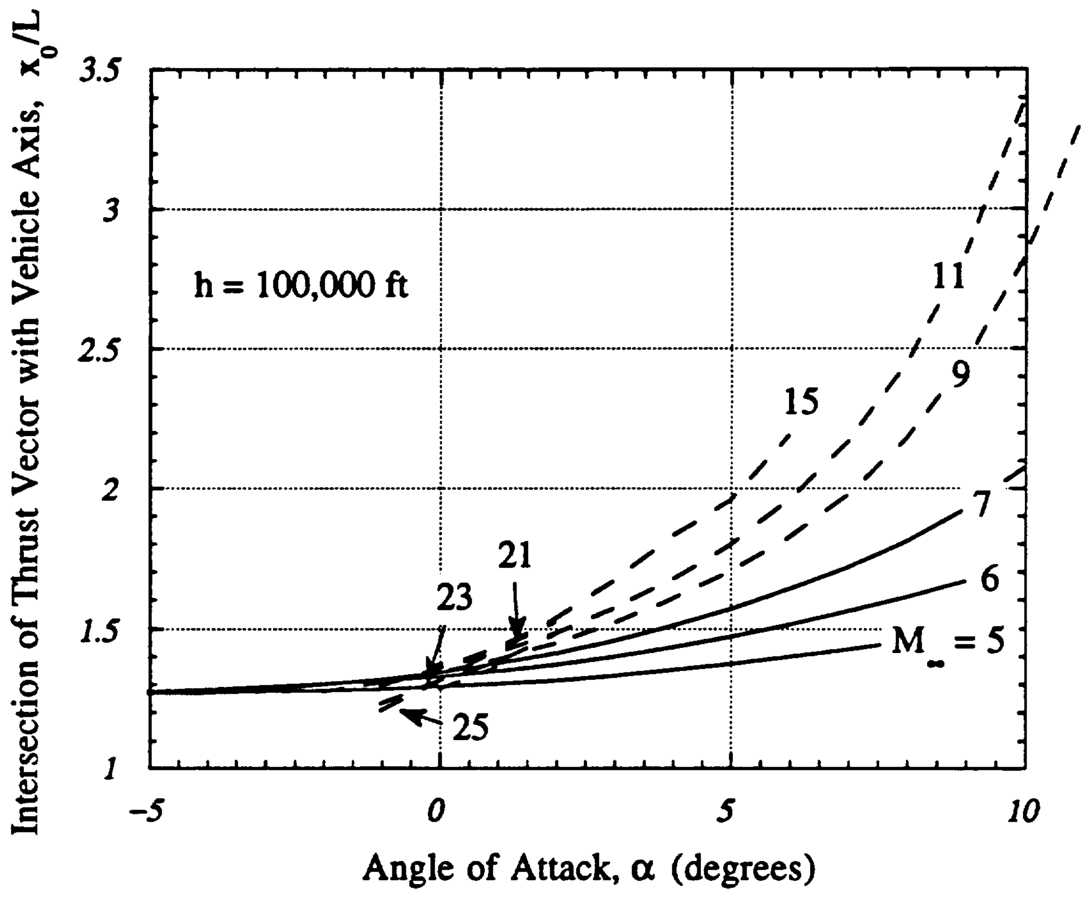


Figure 34. Thrust Line of Action vs Angle of Attack for AWST Vehicle

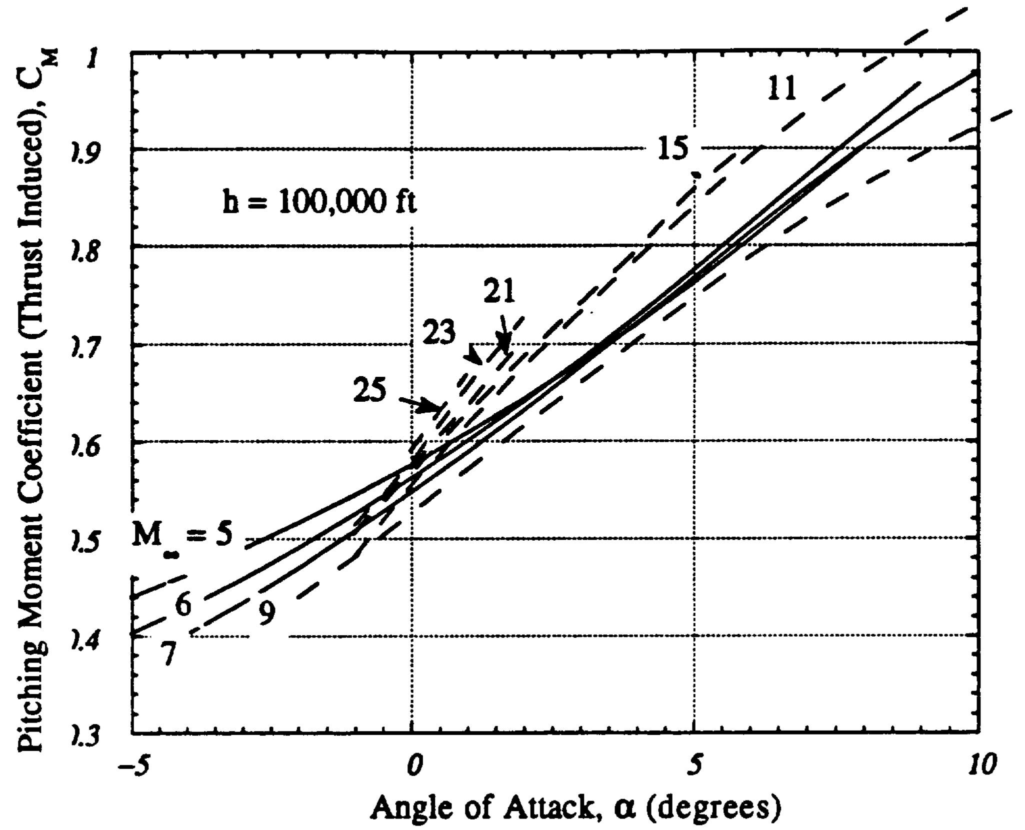


Figure 35. Moment Coefficient vs Angle of Attack for AWST Vehicle

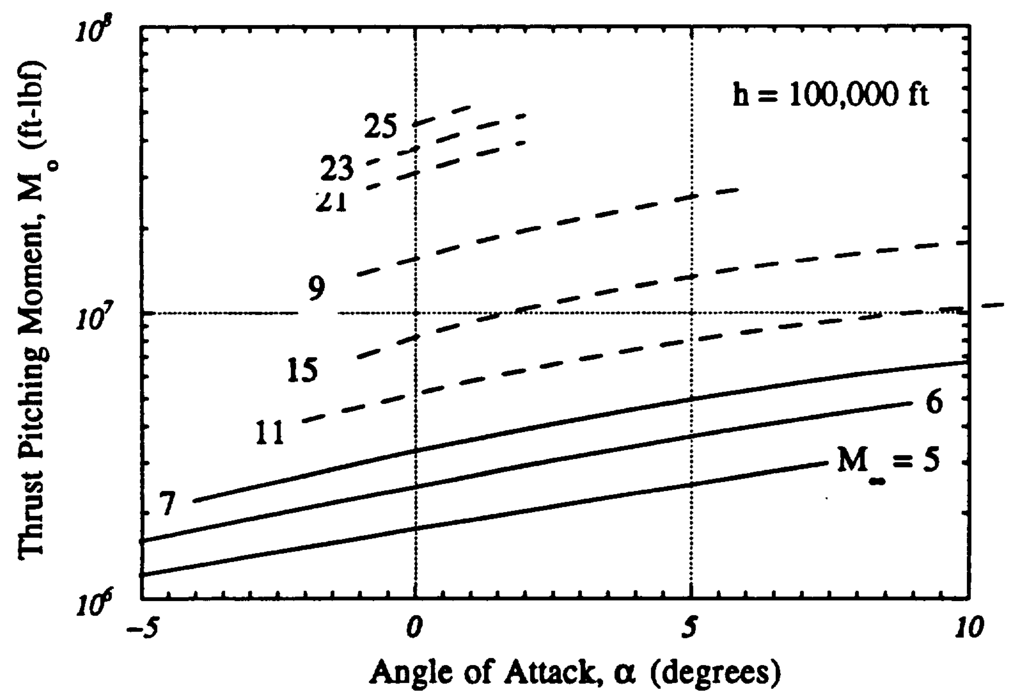


Figure 36. Moment vs Angle of Attack for AWST Vehicle

Some additional results are shown in the Fig. 37-56. These are detailed plots of the flow properties determined by the program as functions of position in the flow field. They are divided into 6 groups. In groups I-III,plots of density, pressure, temperature and Mach number are presented. In groups IV-VI, only pressure and Mach number variations are plotted.

I. Surface variation of some of the variables with Mach number (Figs. 37-40)
II. Surface variation of some of the variables with Angle of Attack (Figs. 41-44)
III. Contours in the Fore and Aft Body Regions for varying Mach numbers (Figs. 45-48)
IV. Contours in the Forebody Region for varying Mach numbers (Figs. 49-50)
V. Contours in the Aftbody Region for varying Mach numbers (Figs. 51-52)
VI. Contours in the Forebody Region for varying Angles of Attack (Figs. 53-54)
VII. Contours in the Aftbody Region for varying Angles of Attack (Fig. 55)

In Fig. 37-40, the results are shown for 5 Mach numbers (5, 7, 9, 11, and 15). Hence the effect of Mach number on the flow is readily seen. Many of the flow features to be mentioned in the text point to the fact that the body shape given does not appear to be an optimum shape. This is not surprising, given the fact that its dimensions were derived by scaling an artist's concept drawing of the National Aerospace Plane from Aviation Week and Space Technology.

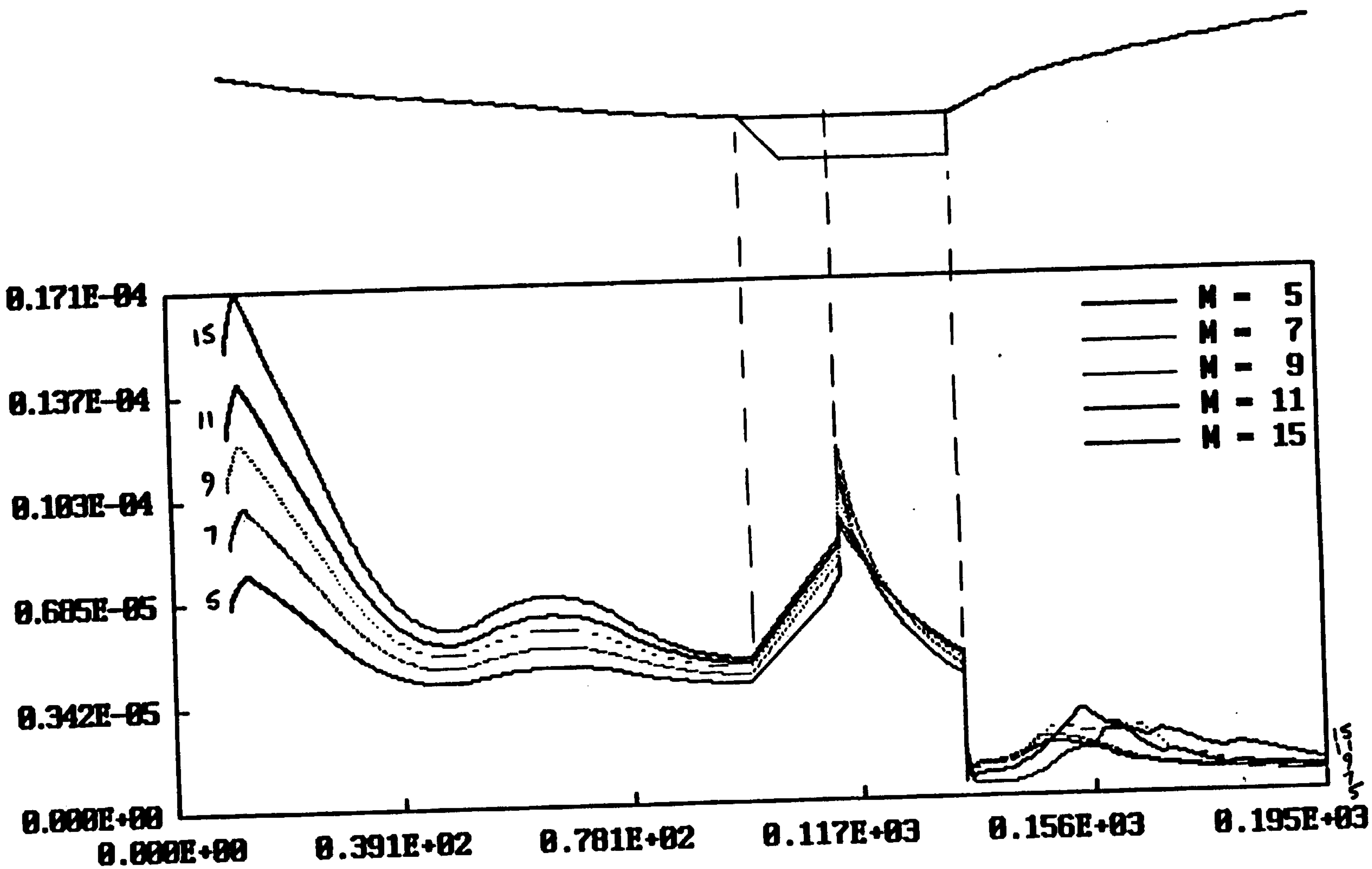
Group I. Surface Variation with Mach Number

In these figures, the values of density, pressure, temperature, and Mach number were recorded on the body surface from nose to tail. They are individually displayed in this series of four plots. In each figure a separate curve is used for each Mach number. The Mach numbers for each of the curves is recorded at the start and end of each of the curves for clarity. In addition, the lower body surface of the vehicle is drawn above the graph so that the relative locations can be more easily assessed.

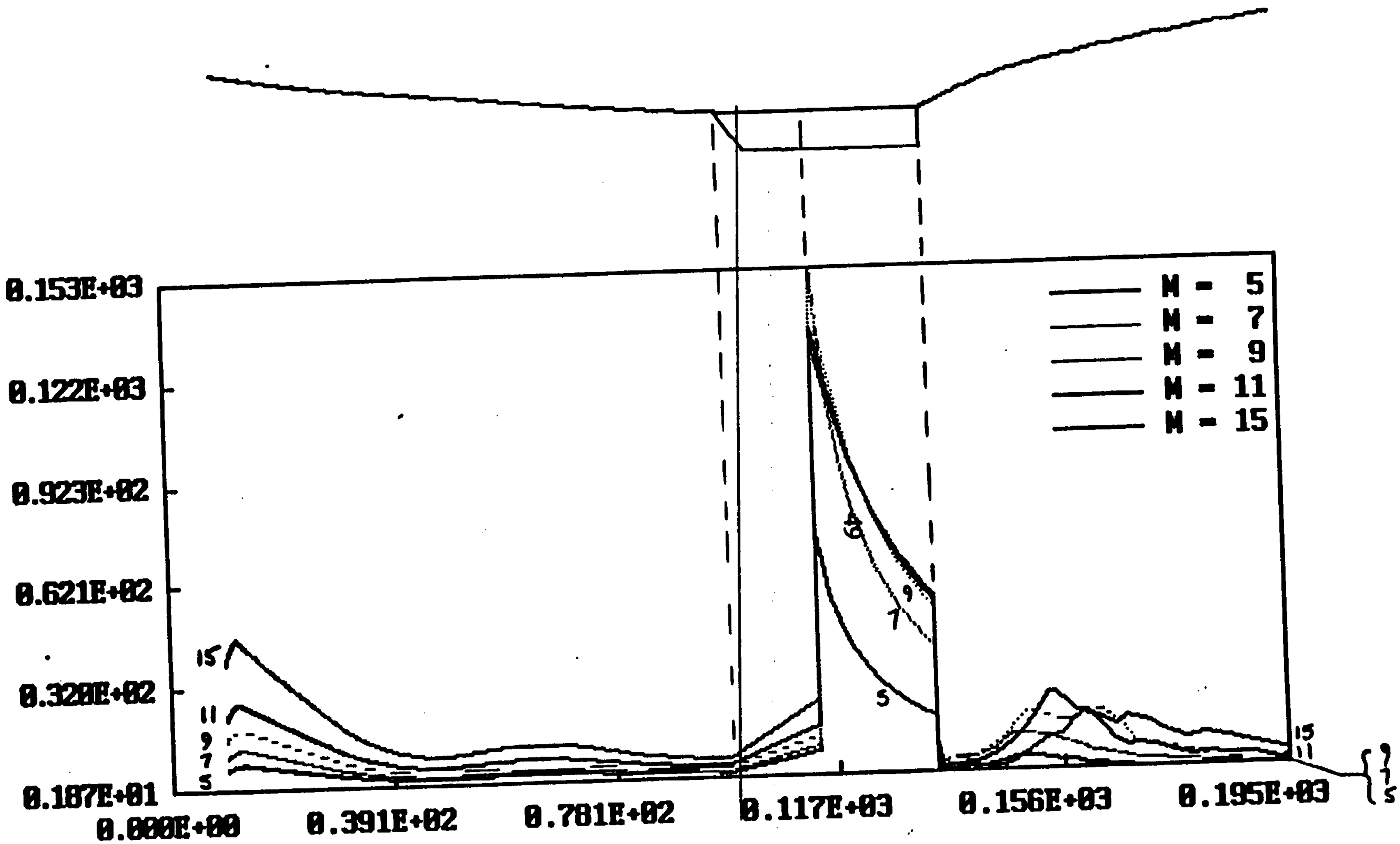
The curves on each of the figures show many of the expected flow features. On the forebody, the rapid expansion near the nose is seen as a rapid drop in the thermodynamic variables and rise in Mach number. A small indentation near the center of the forebody is seen to result in a mild compression which accounts for the forebody expansion not being monotonic. The compression in the inlet is seen by a rapid rise in the magnitudes of the density, pressure, and temperature and drop in Mach number.

At the combustion chamber, there is an abrupt change in the variables since no attempt is made to compute the length of the mixing/combustion zone. So for graphical display, heat is shown to be added at a single station. This is followed by an isentropic expansion in the aft portion of the combustion chamber. Thus a smooth acceleration is seen as the end of the combustion chamber is approached.

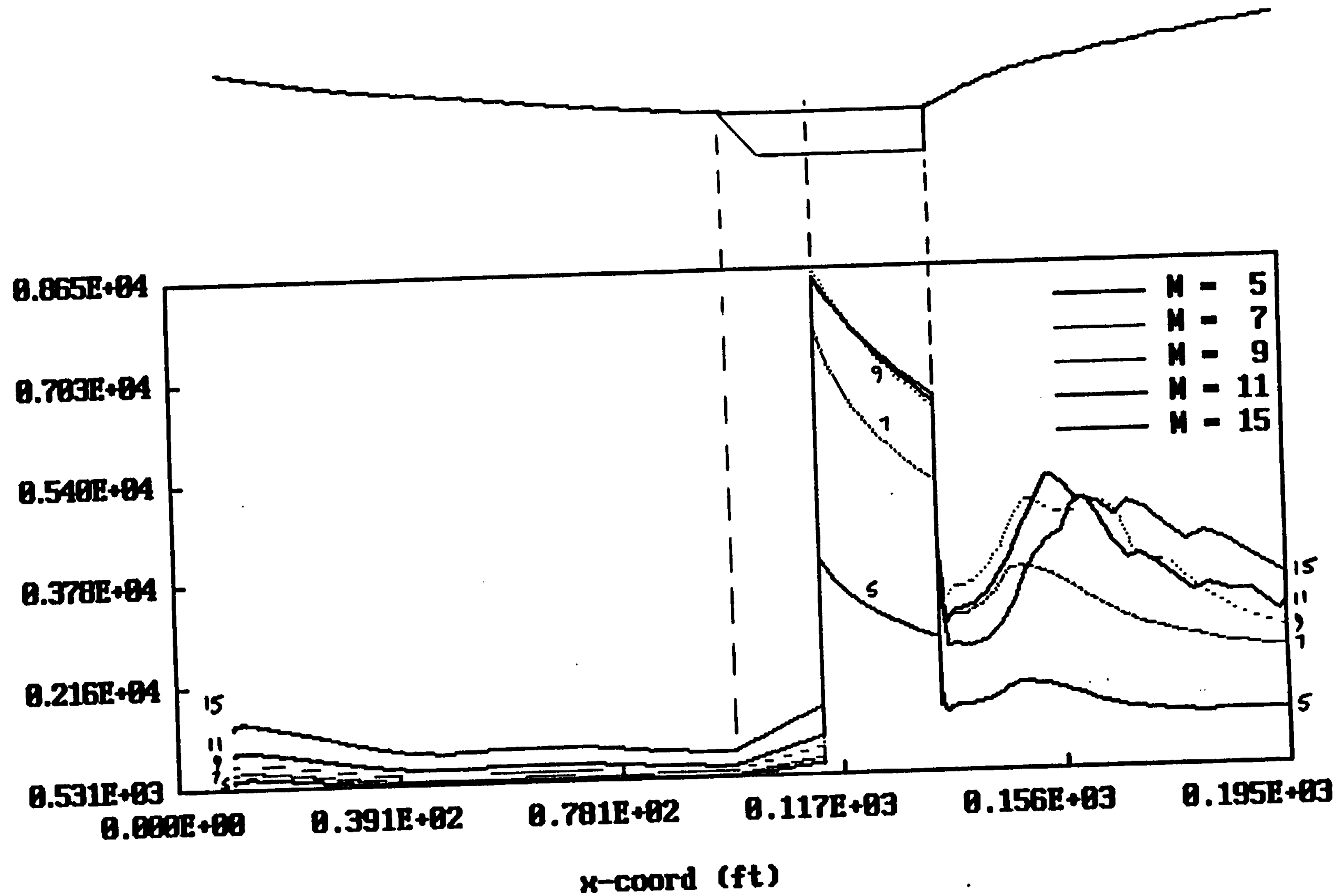
Just outside of the combustor exhaust is a 27 degree expansion which accounts for the sudden drop of the thermodynamic variables and the corresponding rise in Mach number. This is



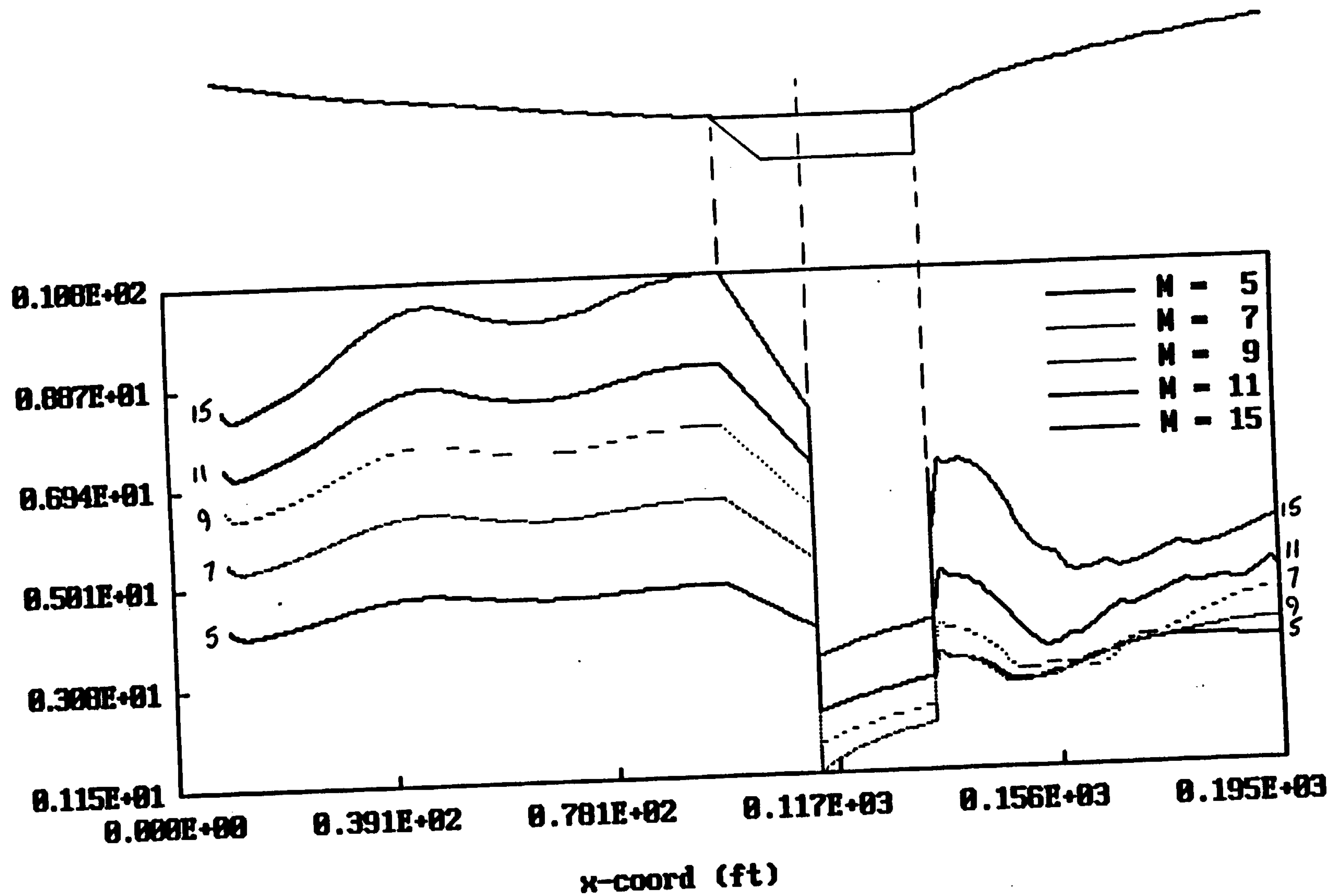
Density (slug/ft^3) Along Vehicle Surface



Pressure (psf) Along Vehicle Surface



Temperature (°R) Along Vehicle Surface



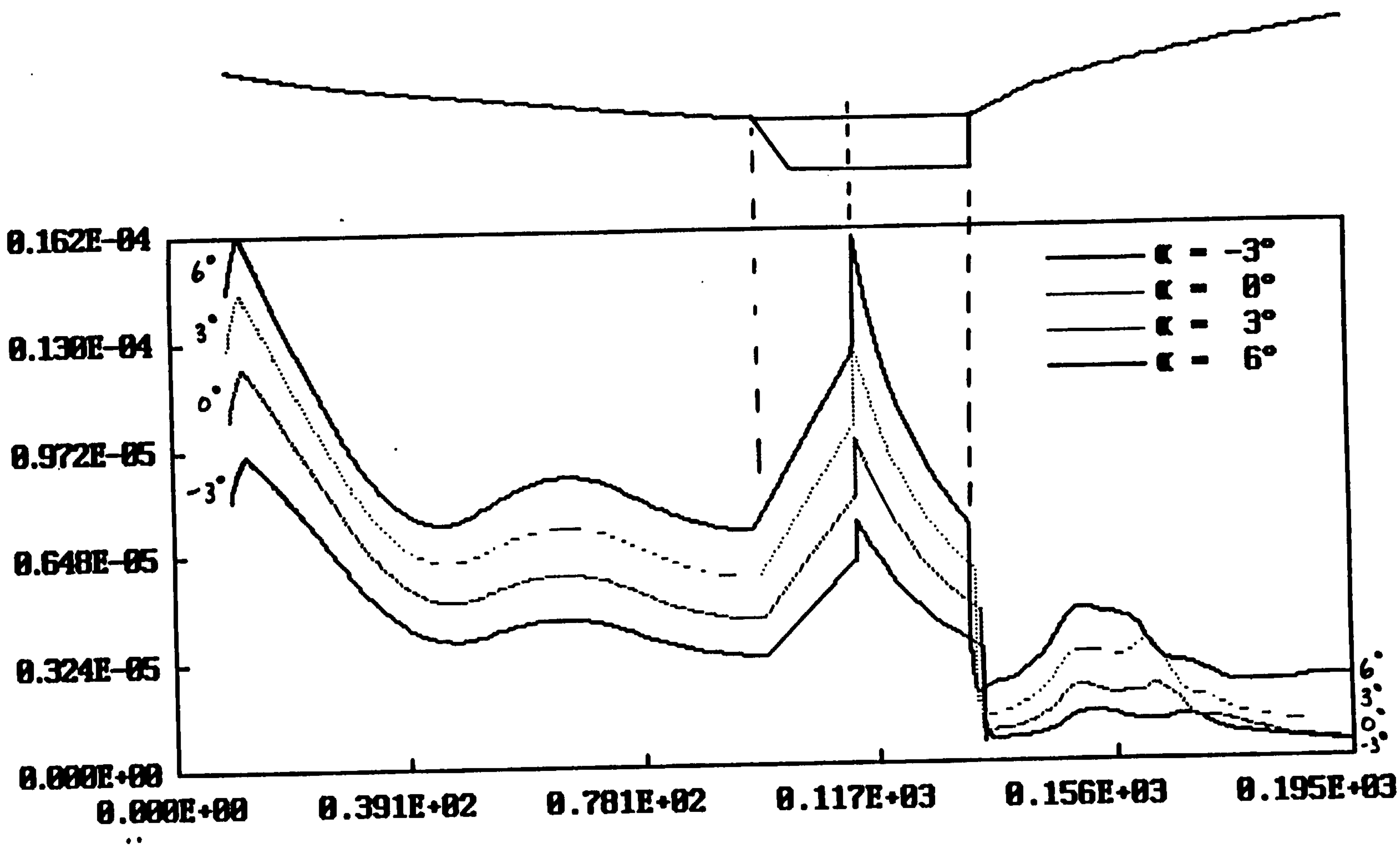
Mach Number Along Vehicle Surface

followed by a gradual compression downstream as the flow progresses along the concave aftbody. The values of the variables on the aft body are affected by this compression as well as the impingement of expansion waves from the outer corner of the combustor exhaust and compression waves reflected from the slip line. These conspire to give a rather confused type of behavior of the variables on the aft body surface. This really only means that the body used does not appear to be an optimum surface.

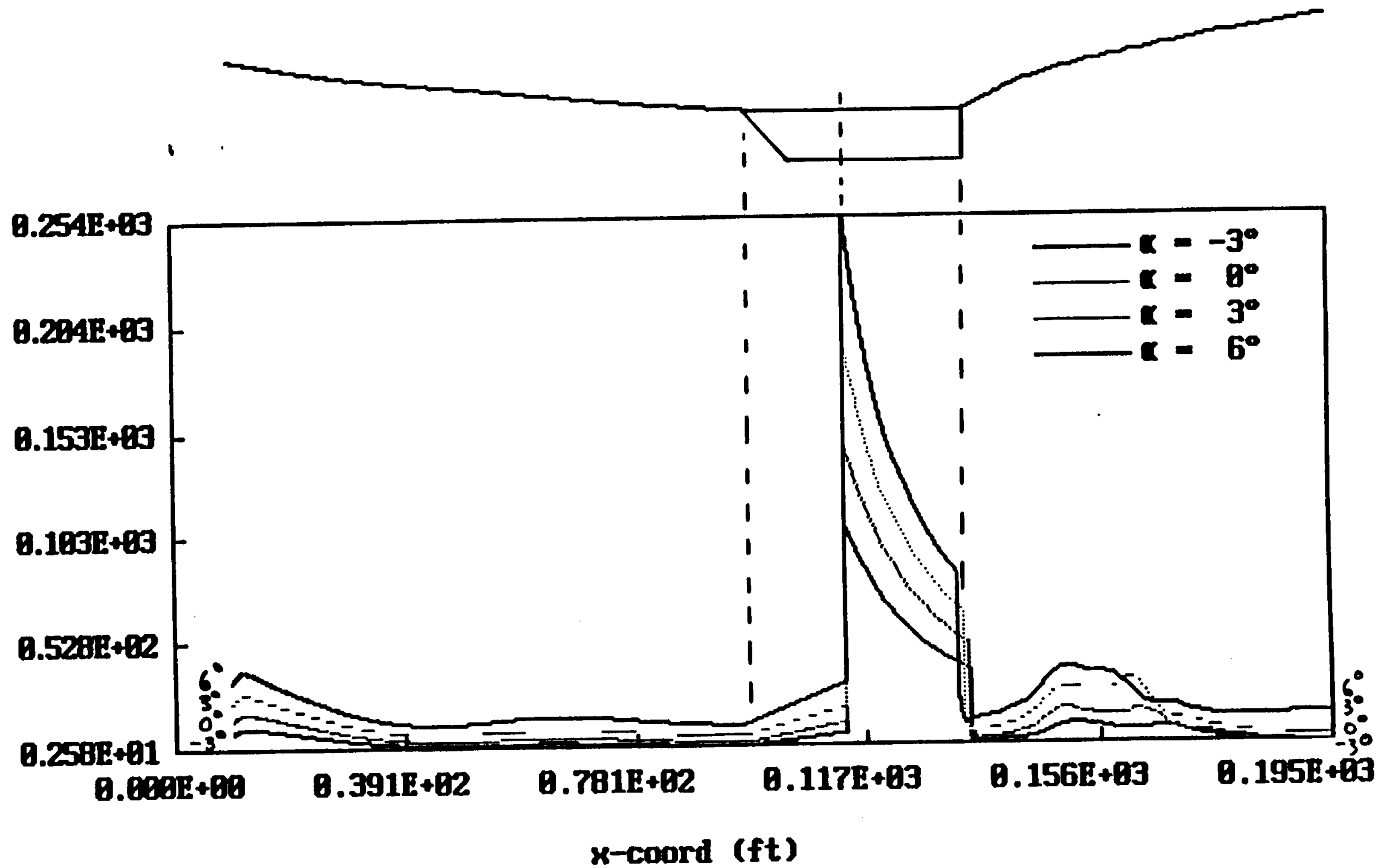
Group II. Surface Variation with Angle of Attack

In this series the angle of attack is varies at a freestream Mach number of 9. The angles used are from a negative 3 to 6 degrees. At an angle of 8 degrees the bow shock enters the inlet for this configuration and the solution procedure is stopped. The main effect of angle of attack is make changes in the variables on the forebody. The forebody Mach number in particular is lowered with increasing angle of attack as a larger disturbance is exposed to the flow. Correspondingly the thermodynamic variables are all increased in response to the greater compression.

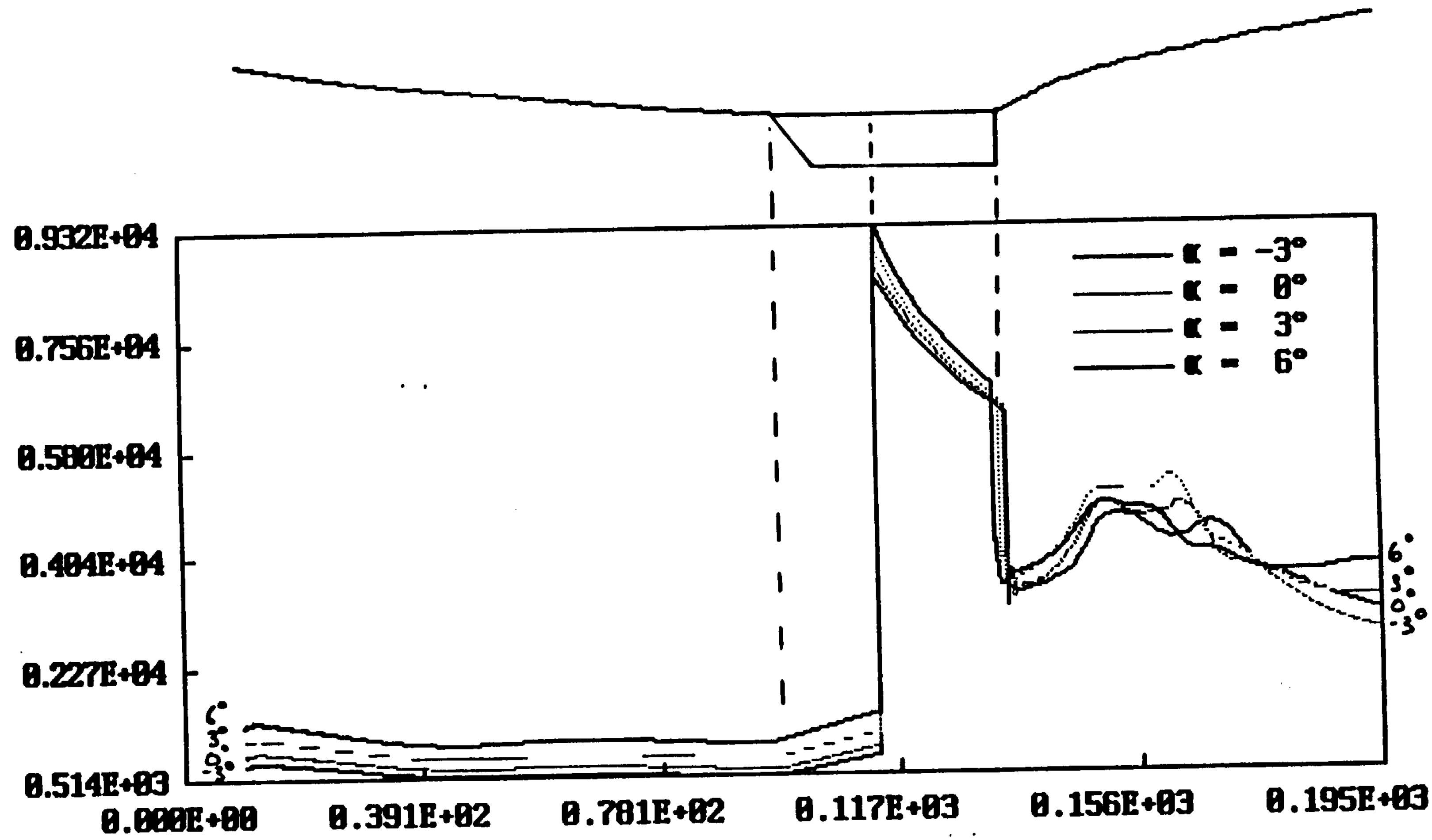
Downstream, the flow is less affected by changes in the angle of attack. This is particularly true of the Mach number which appears to be largely independent of the angle of attack once it enters the inlet. The thermodynamic variables still increase in the downstream portions of the domain in response to increases in angles of attack, but at a somewhat lessor amount than on the forebody.



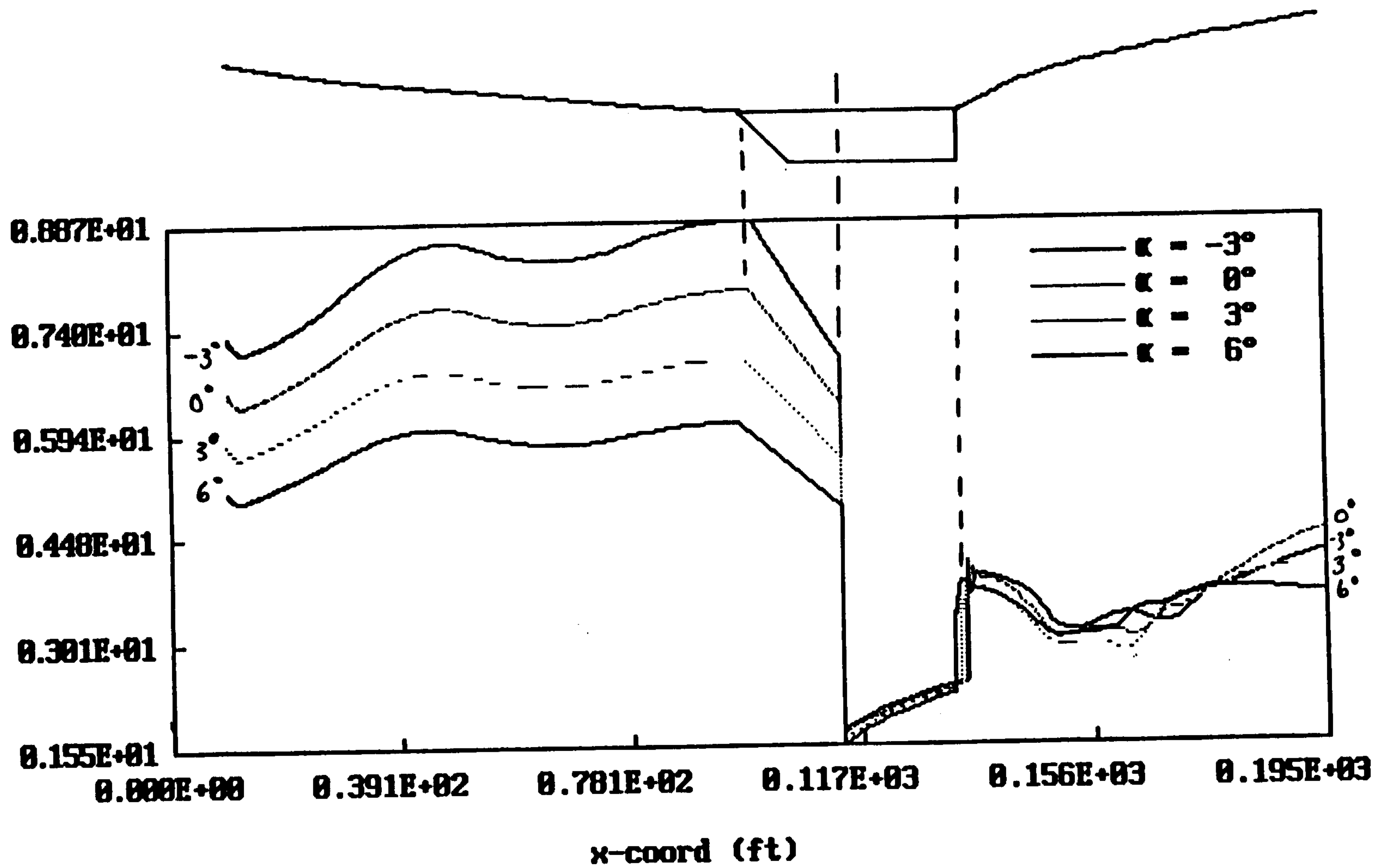
Density (slug/ft^3) Along Vehicle Surface for Various Angles of Attack



Pressure (psf) Along Vehicle Surface for Various Angles of Attack



Temperature (°R) Along Vehicle Surface for Various Angles of Attack



Mach Number Along Vehicle Surface for Various Angles of Attack

Group III. <u>Contours in the Fore and Aft Body Regions for varying Mach Numbers</u>

In this series of figures, contours are given of the density, pressure, temperature, and Mach number in the flows over the fore and aft surfaces. These figures show the main features of the flow in these regions. In each figure, the forebody region is given in the left side and the aft body region given on the right. The two regions were place a distance apart corresponding to the distance occupied by the scramjet module. In each figure, the upper boundary is the body surface and the lower boundary is a shock wave.

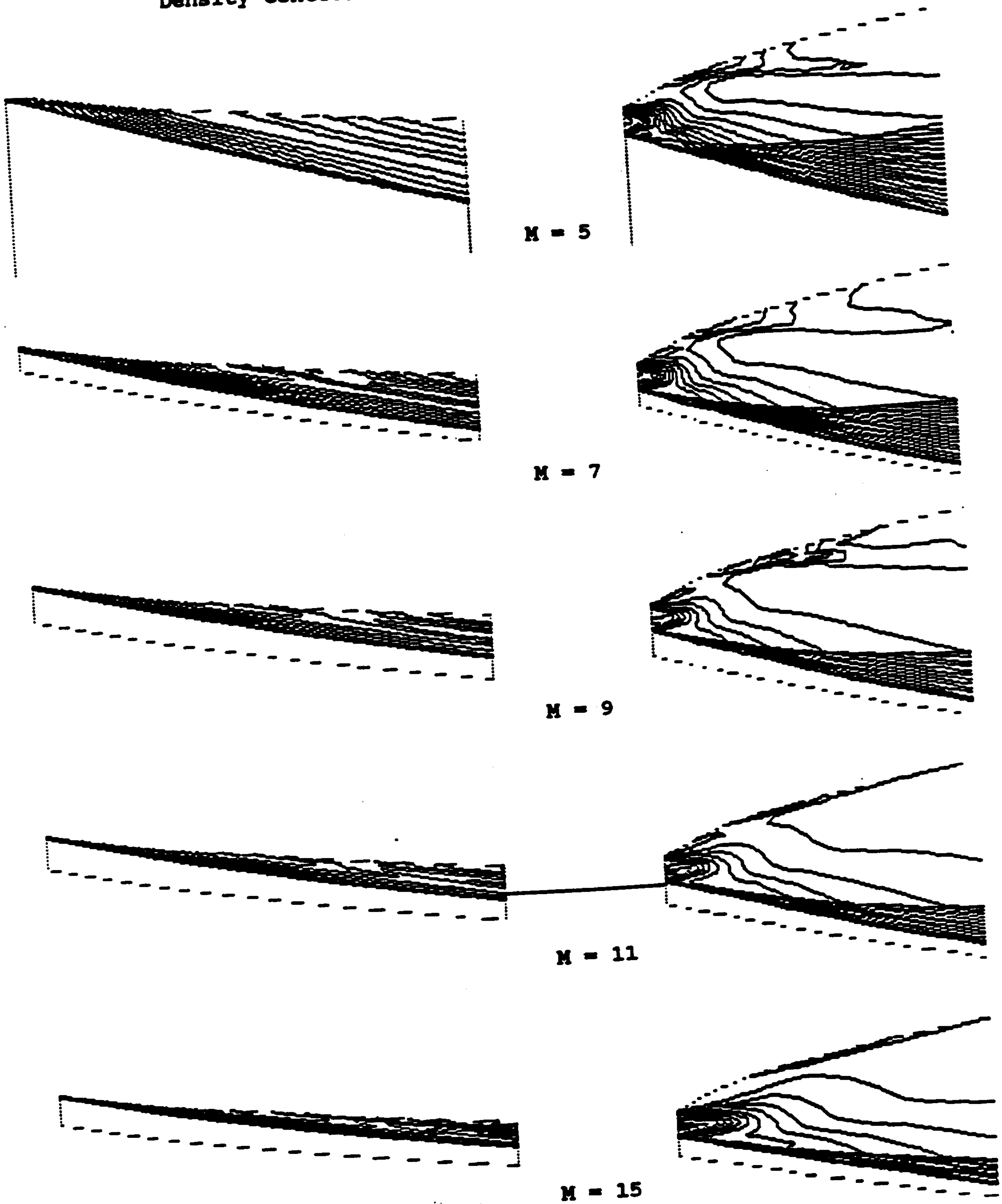
The reduction in shock angle with increasing Mach number is clearly seen in each of the figures and is evident in both fore and aft regions. The expansion around the convex portions of the forebody can also be seen as lines emanating from the body as if they were expansion waves. It may recalled that expansions are line of constant properties whose angle is equal to the local Mach wave angle. Hence, many of the contours can be thought of as Mach waves.

In each of the forebody figures, the bow shock is seen to be curved as would be appropriate for an expanding flow over the roughly convex shape. The small compression caused by the presence of the dent near the center of the forebody is seen most clearly in the temperature and density contours and to a lessor extent in the Mach number contours.

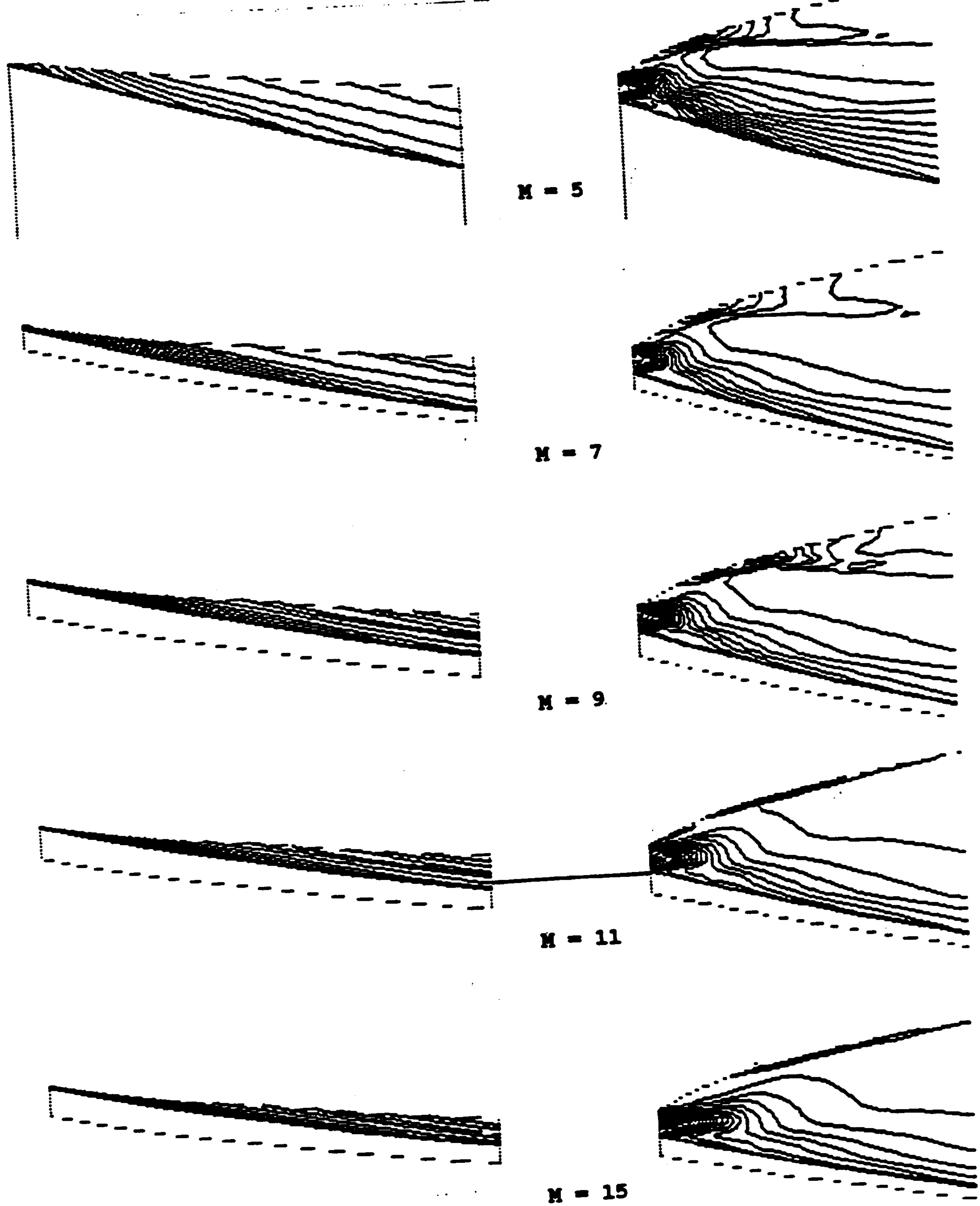
In the aftbody regions, the external shock wave is evident in all figures. It is also curved as expected. In addition, the slip line can be seen in the temperature, density, and Mach number figures. It does not show up in the pressure contours since the pressure across a slip line is constant and is thus not remarkable for that variable.

The other main feature visible in these figures is the compression caused by the concave aftbody surface. This compression is most clearly seen in the Mach number contours. At the lower Mach numbers, the compression is seen to be emanating from the location of severest curvature on the aft body. At the higher Mach numbers is has become more closely aligned with the aftbody surface. If the compression becomes too strong, then it can cause some stability problems with the Method of Characteristics routine. The user should be aware that the fix is relatively easy by simply reducing the distance between stations. Other flow features are seen more clearly in the close-ups of these regions in the next two series.

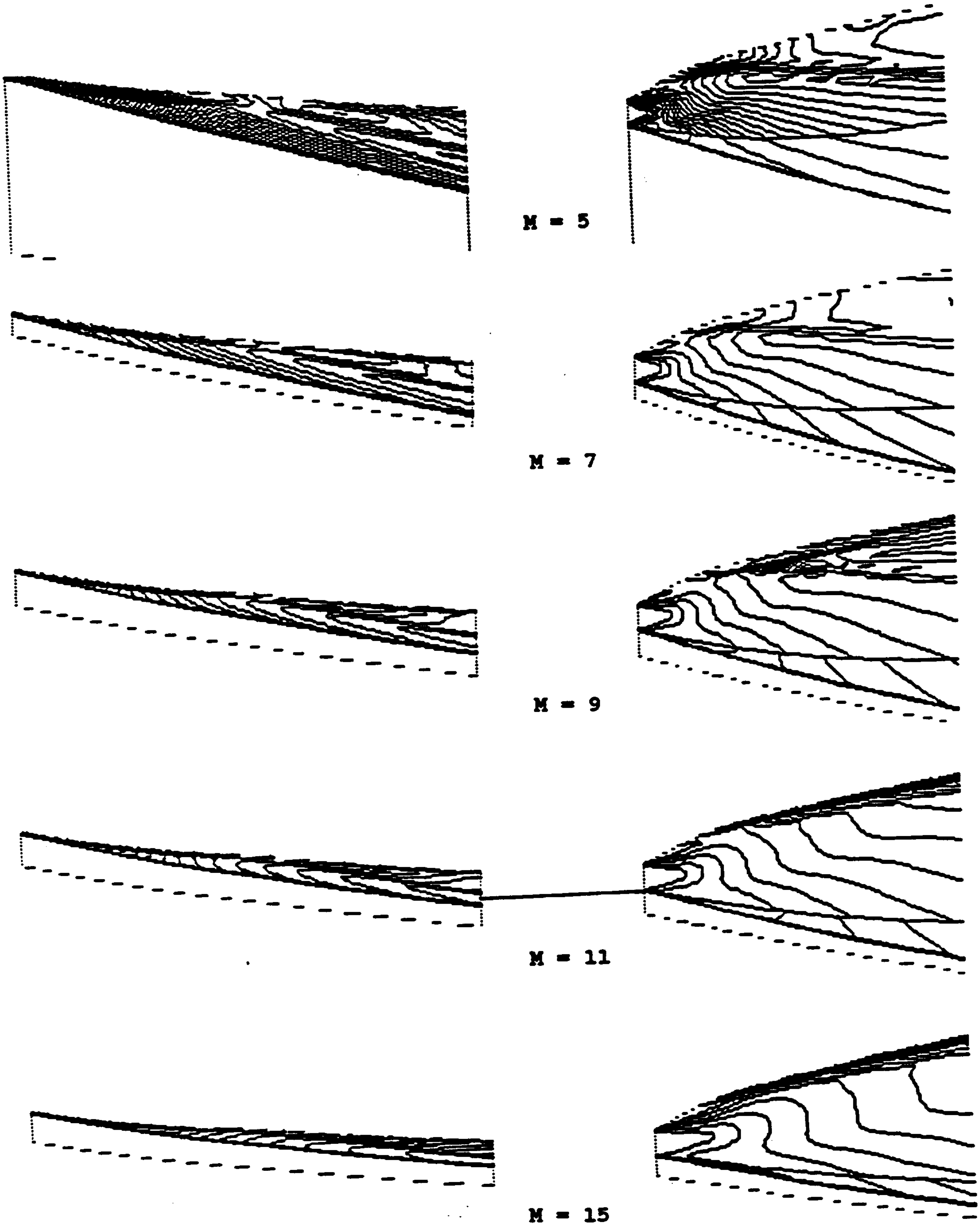
Density Contours in the Fore and Aft Body Regions



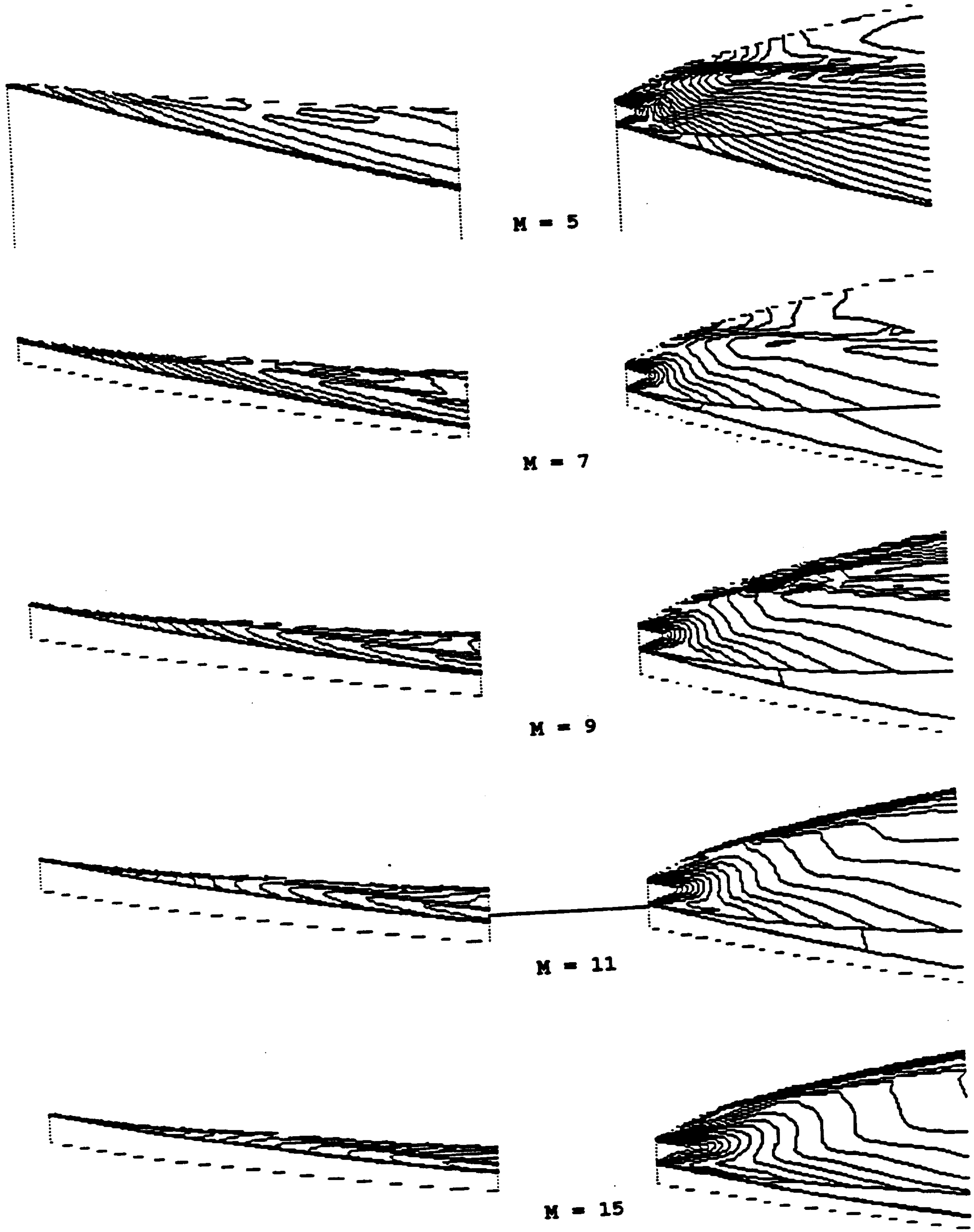
Pressure Contours in the Fore and Aft Body Regions



Temperature Contours in the Fore and Aft Body Regions



Mach Number Contours in the Fore and Aft Body Regions



Group IV. Contours in the Forebody Region for varying Mach numbers

Here, only figures of the pressure and Mach number were made. These figures were made to show the features in these regions more clearly. In the Mach number contours, a small mismatch between the coneflow routine and the forebody MOC results in a wave near the nose which appears to reflect off the bow shock. The wave diminishes rapidly after that and is essentially gone by the time the flow reaches the middle of the forebody. It does seem to hit the surface near the dent and may be affecting the flow near the dent, though it is likely a weak event.

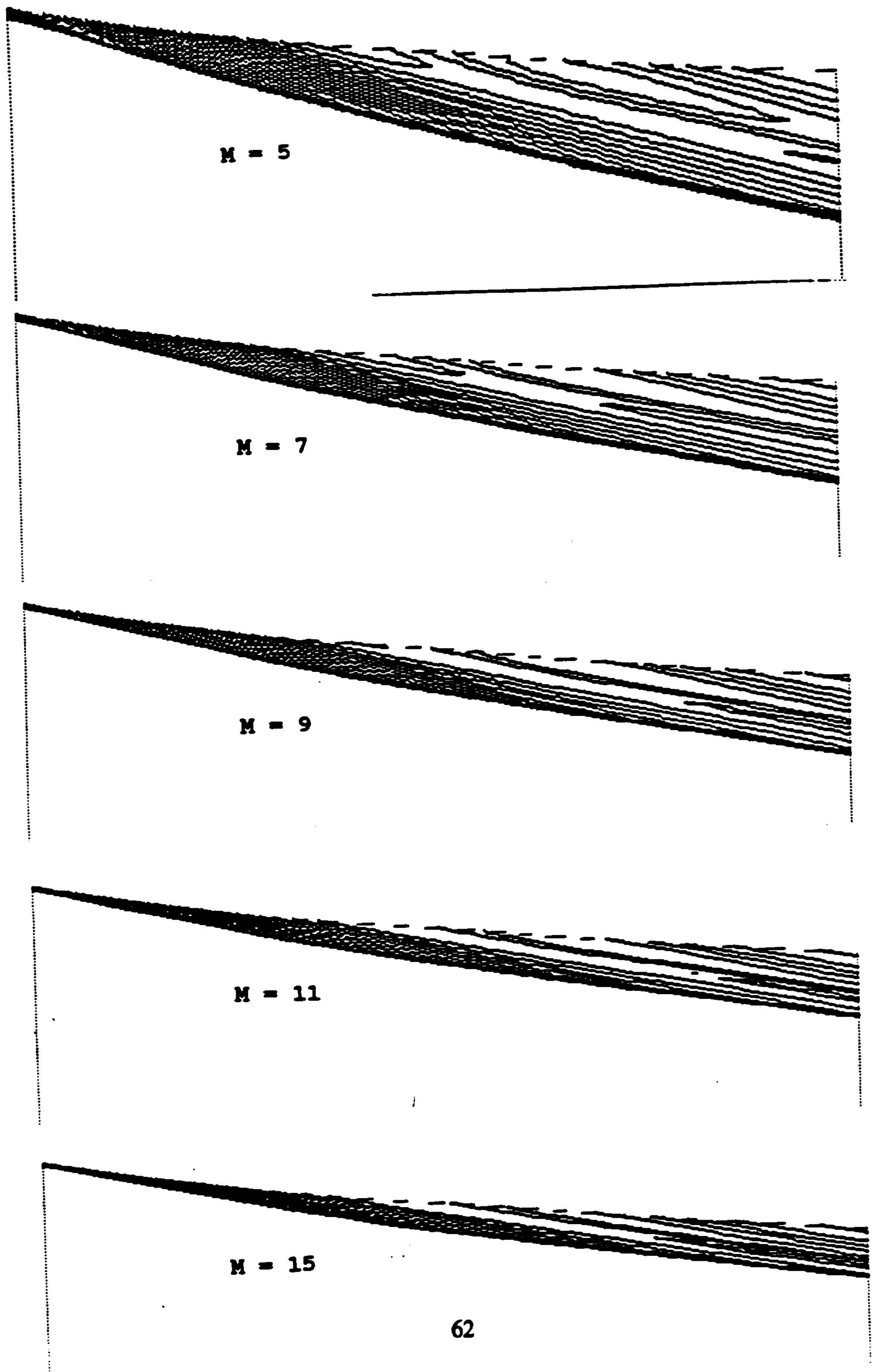
Group V. Contours in the Aftbody Region for varying Mach numbers

In the aftbody contours, the expansion at the combustor exhaust is clearly evident. At the outer edge, the pressure of the freestream is much less than the energized flow from the engine and an expansion is taking place at the outer edge for this reason. Near the body surface, the intersection of the aftbody with the combustor is at a 27 degree angle causing the flow to turn by that amount. In both cases, the expansion fans are clearly visible. These fans cross each other and reflect off the body surface and the slip line. The reflection from the slip line can be seen to produce a slight compression as the expansion waves reflect from the slip line in an unlike sense. The effect of the compression is to cause a deflection of the contour lines away from the slip line and is most visible in the Mach number contours at the higher Mach numbers.

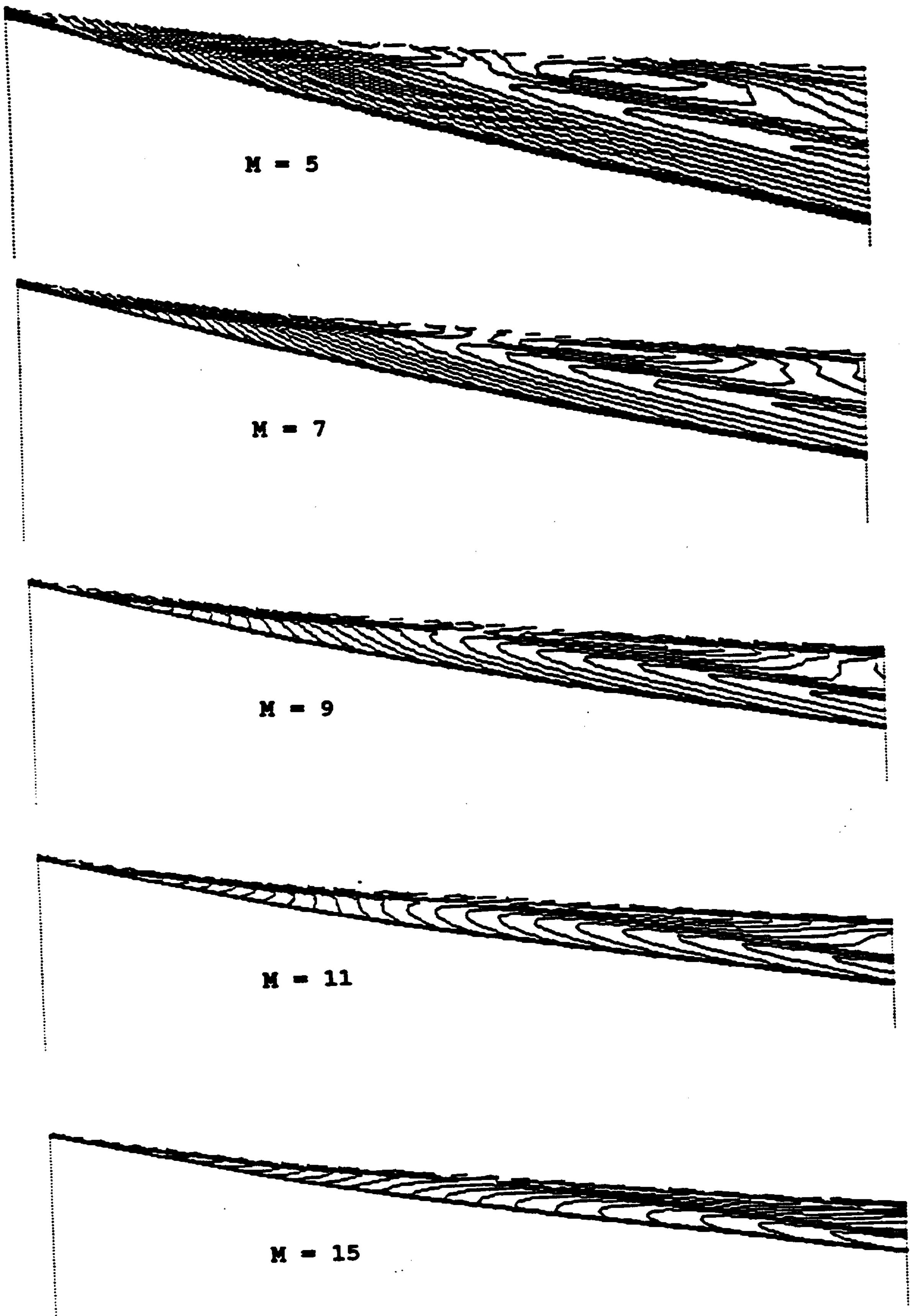
The slip line is seen to curve upward in response to the continuous series of expansions hitting it. This, in turn causes the flow between the slip line and the shock to also undergo its own expansion. This is most clearly seen in the Mach number contours at the lower Mach numbers. At the higher Mach numbers, the expansion fan from the inner corner does not hit the slip line until further downstream. Hence there is less slip line curvature at the higher Mach numbers.

Another interesting feature are the angles of the expansion waves nearest the core of the exhaust flow. As the Mach number increases, the Mach wave angle becomes smaller which lengthens the triangular core region between the two expansion fans.

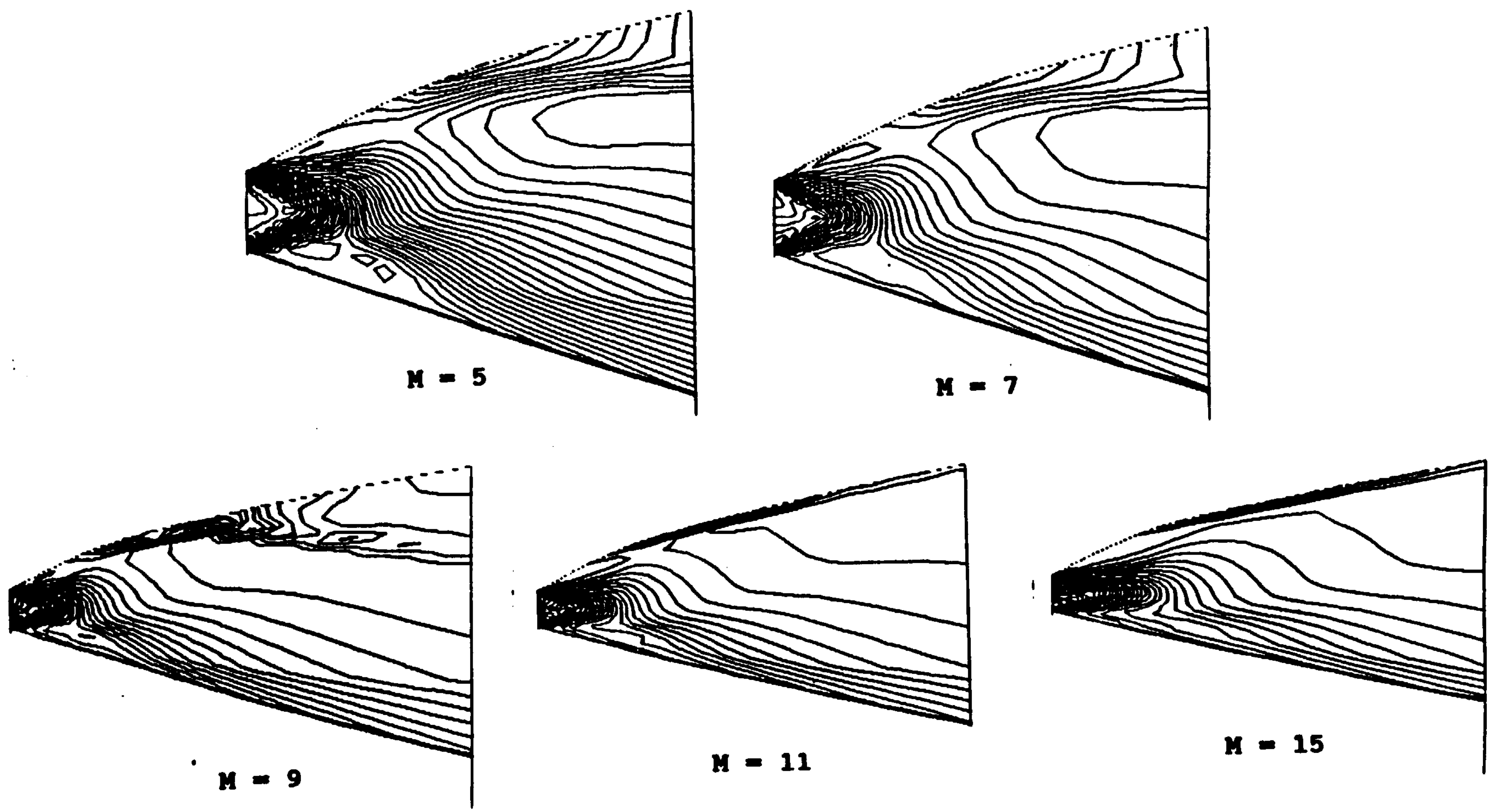
Pressure Contours in the Forebody Region



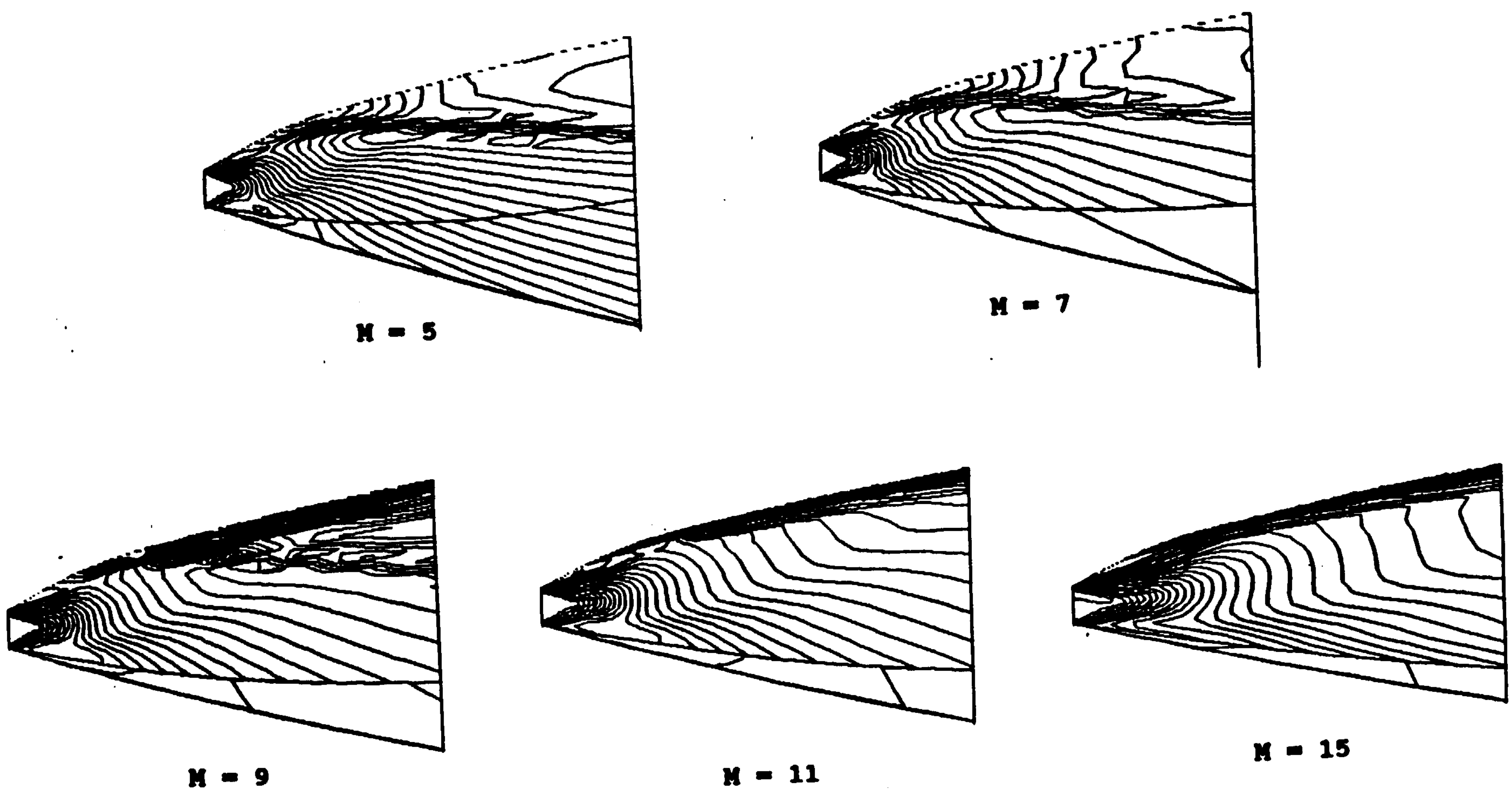
Mach Number Contours in the Forebody Region



Pressure Contours in the Aftbody Region



Mach Number Contours in the Aftbody Region



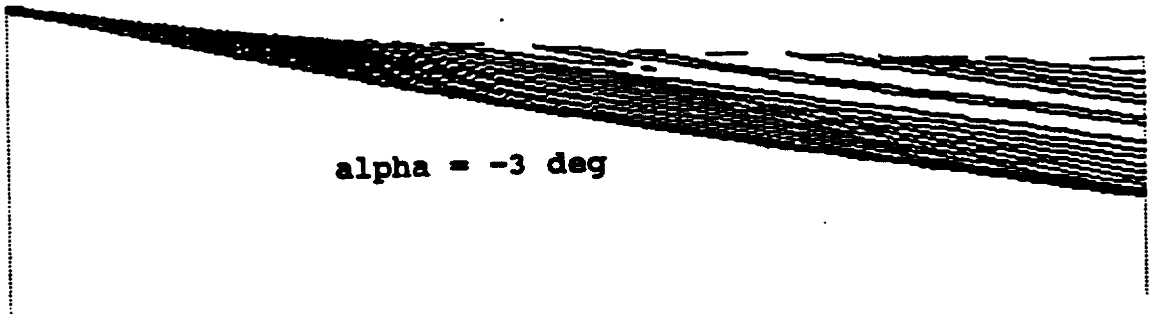
VI. Contours in the forebody region for varying angles of attack

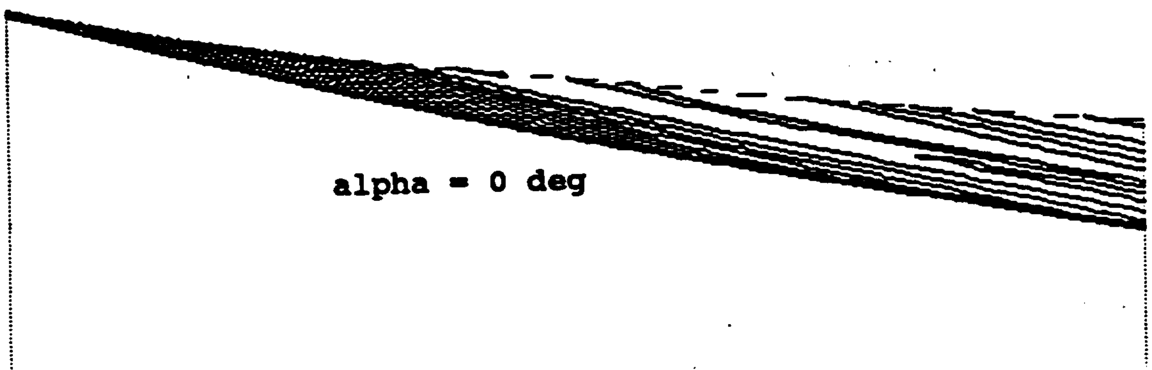
The most evident effect on the flow near the forebody of changing the angle of attack is to cause the shock to move closer to the surface. The pressure contours are rather unremarkable in any other way other than to point out that the small compression is seen near the indent at about the midpoint of the forebody. The Mach number contours more clearly show the compression and also show that the compression moves close to the shock with angle of attack and at 6 degrees actually intersects it. In addition, the slope of the Mach number contours in the nose region near the shock change direction.

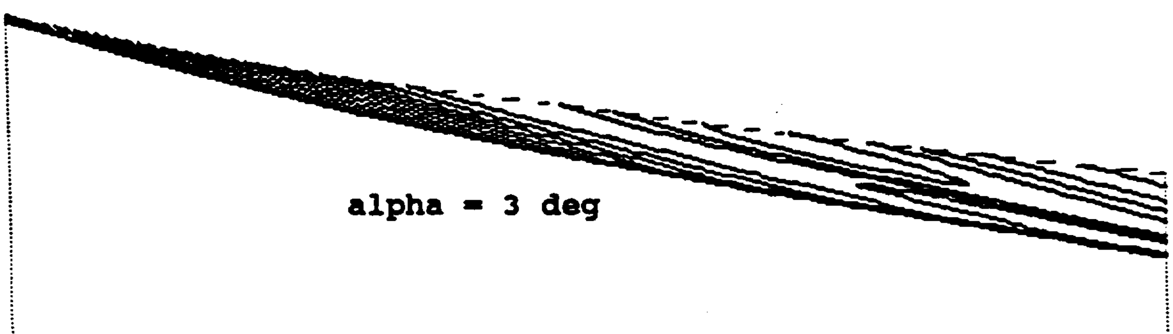
VII. Contours in the Aftbody Region for varying Angles of Attack

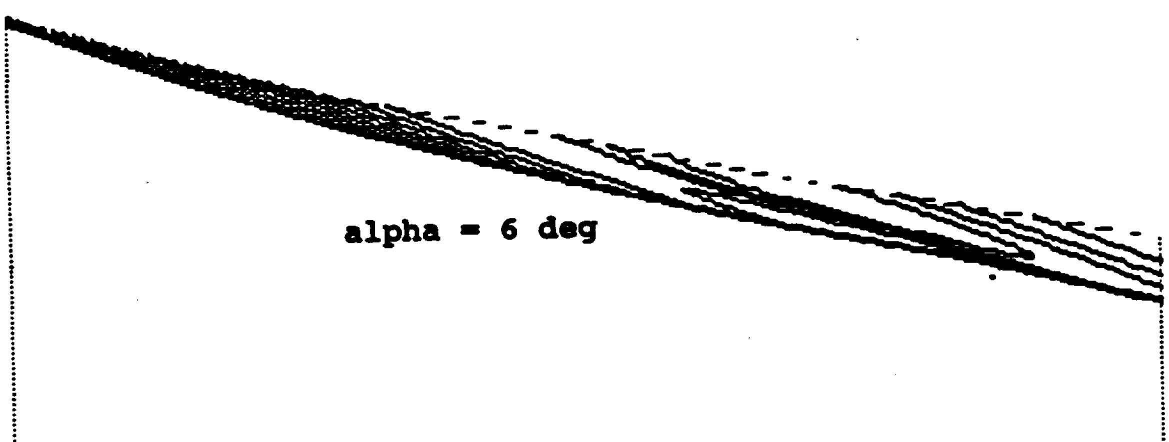
The final figure shows the pressure and Mach number contours in the aft body region at angles of attack of -3, 0, 3, and 6 degrees. As before, the first noticeable change is that the regions diminish in size slightly as the angle of attack increases. The distance between the body and the slip line at the aft end, for example, decreases by about 50% with angle of attack change from -3 to 6 degrees. That the remaining qualitative features of the flow change little is consistent with the relative insensitivity to angle of attack changes of the variables on the surface seen above.

Pressure contours in the Forebody region for varying angles of attack



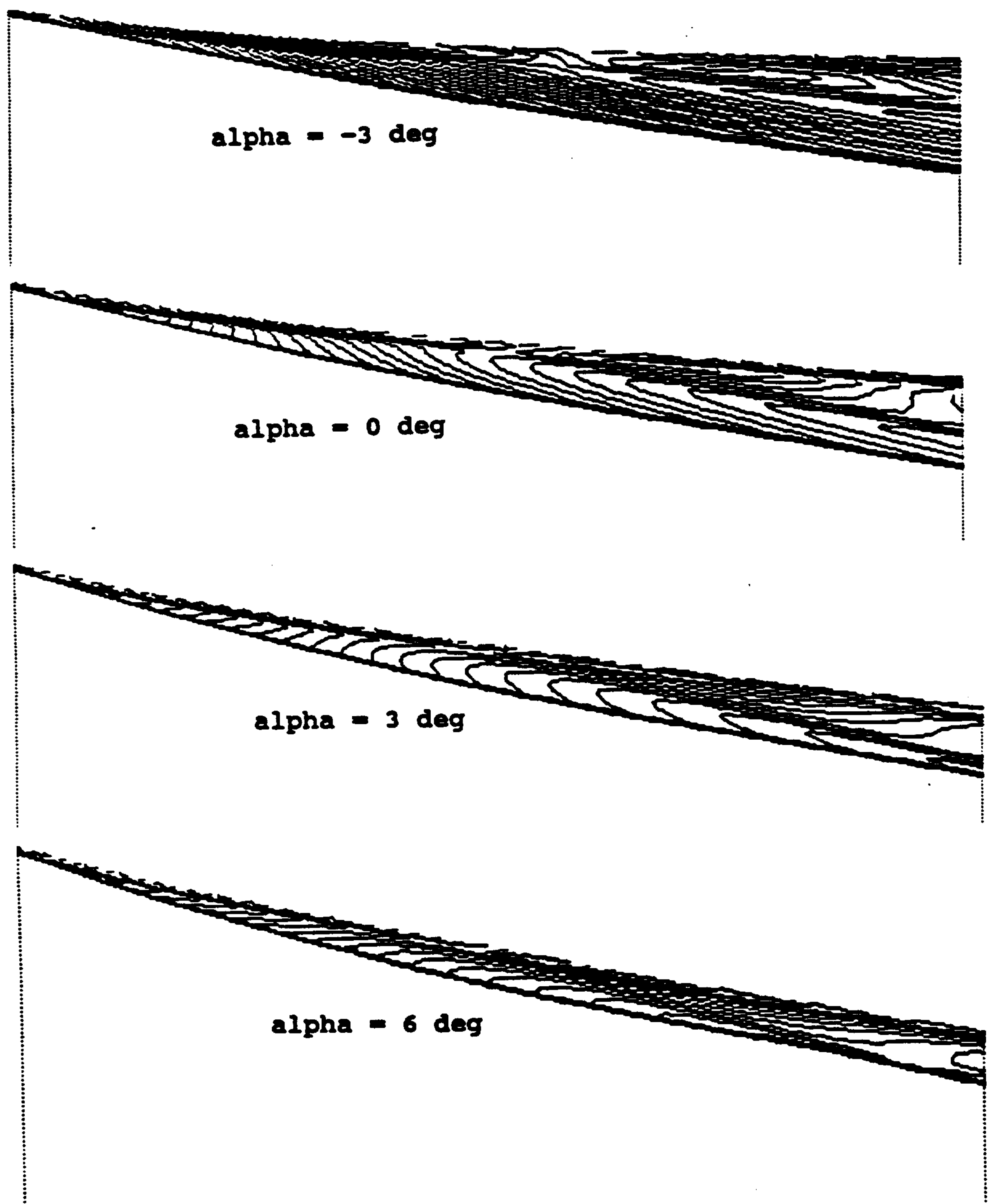




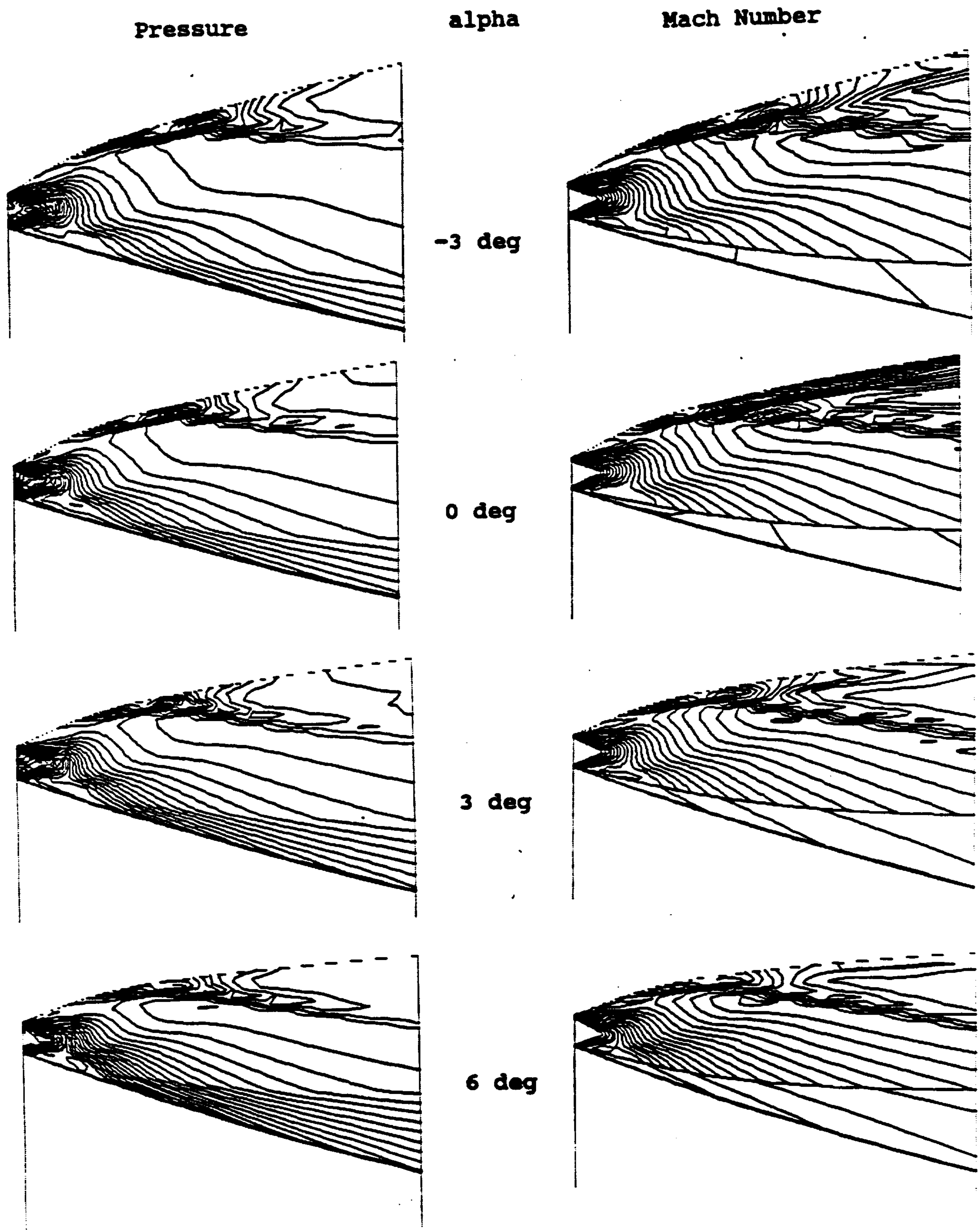


Mach Number contours in the Forebody region

for varying angles of attack



Pressure and Mach Number Contours in the Aft Body Region for varying Angles of Attack



CONTINUED DEVELOPMENT OF THE PROGRAM

The program was designed to provide the user with an estimate of the change in forces and moments on a hypersonic vehicle due to changes in pitch and changes in various geometric and propulsion system parameters. The code was primarily oriented toward finding the solution for the flow field as quickly and as accurately as possible. Various methods were chosen for distinct regions of the flow based on these criteria. Thus the code does solve the entire flow field on a hypersonic vehicle underbody, including the propulsion system, in an extremely short time (less than one minute CPU time on a small mainframe and about one minute on a workstation class machine).

As development of the code proceeded, though, it became clear that a number of changes or additions would result in greatly improved flexibility and, in some cases accuracy, without sacrificing the great speed at which the procedure computes the flowfield. This document outlines the major changes to the program which would implement these improvements. A major goal will be to incorporate models that can accommodate imbedded shock waves. Only in this way can realistic vehicle geometries be analyzed accurately.

1. A fully 3D Version - The present code was mainly oriented toward vehicle configurations which used either axisymmetric or flat forebodies so that a two dimensional Method of Characteristics procedure was used. It was somewhat fortuitous that it became possible to do other shapes as long as a knowledge of the local body curvature was available and there was no crossflow. Thus virtually any shape may be analyzed at any angle of attack as long as there is no yaw. In order to be able to do a problem with yaw, however, requires use of three dimensional equations. Thus a three dimensional form of the Method of Characteristics is recommended.

To implement the three dimensional form of the MOC, essentially the same code will be used in projected streamline planes as in Rakich [2]. Modifications to the present MOC will include the addition of the third momentum equation and changes in the geometric quantities in the other two momentum equations. Ordinarily, one would expect a significant slowing of the computational procedure since there are now multiple streamwise planes whereas the original MOC code had only one each across the fore and aft bodies. In addition, the time should be greatly increased because of additional iterations to convergence on all these planes. However, Rakich indicates that iteration is rarely required to reach convergence and can be dropped as a requirement. Since the present program already solves multiple streamwise planes (one for each propulsion system module) one at a time, the new procedure would solve all of them simultaneously. This would result only in increasing the memory requirement of the procedure and the execution of the coupling terms in the equations, resulting in an estimated increase of 20-25% CPU time.

2. Imbedded shock version of the MOC procedures - In the currently envisioned design, the exhaust portion of the cowl extends past the exhaust/body intersection. This extension is to be used to redirect the flow to change the direction of the thrust vector. Any movement of the extension or flap, will result in a shock or an expansion. As a consequence, shocks may exist in the aftbody region. As a consequence, it would be worthwhile to be able to add a shock tracking feature to the method of characteristics computation. As previously mentioned, this is not a conceptually difficult exercise and should be a straightforward extension of the Hartree MOC procedure.

3. Ramjets version - Since the current propulsion system is assumed to use scramjets, the code is restricted to hypersonic vehicles in mid-supersonic through hypersonic cruise configurations. There are, however, a number of interesting configurations at the lower supersonic speeds which may be using Ramjet propulsion systems, such as the High Speed Civil Transport and the SANGER vehicles. In this case, the inlet and combustion systems need to be modified to allow subsonic flows. The changes required would be to allow for the terminal normal shock in the inlet to be located at an appropriate minimum area and to allow for subsonic combustion in the combustor. Since these are fairly easy to implement within the framework of the present code, this enhancement should be fairly straightforward.

4. Addition of boundary layer effects - These are currently ignored in the inviscid analysis. However, the effects of viscosity and turbulent heat transfer can be quite significant since the boundary layer is rather thick on the body surface under the flight conditions being stated. Thus it is deemed an important addition to be able to provide reasonable estimates of the viscous heat transfer at the surface. This may be done very easily by incorporating any of a number of boundary layer programs. All are fairly rapid and provide excellent estimates of the wall shear stress and heat transfer. If shocks are present, however, a method which can compute the flow separation and reattachment may be necessary.

5. Inclusion of an on-board air liquefaction system - The advantage of liquefying captured air for improvements in vehicle specific impulse is well known. In order to assess a configuration using such a device, the capability must exist to include its effects. A heat exchanger design package currently exists [5] as a separate module and needs to be incorporated (along with a couple of improvements) to the main code. The only inputs to the code are the flow capture rate (which depends on the boundary layer velocity profile) and the heat sink properties of the fuel (in this case liquid hydrogen has been assumed). This improvement is coupled with the addition of a boundary layer treatment since most LACE systems use captured boundary layer air.

6. Computation of the flow on the upper surface - Currently, the program computes only the flow through the four regions of the underbody. The force and moment computations are then computed from these. While, these represent the contributions from the propulsion system, it would be convenient to know the total body forces and moments particularly if the cost would

be low. To get these would only require the computation of the flow on the upper surface and on the wings. It is envisioned that the upper surface would be particularly simple to compute since this would require only slight modification of the MOC procedure used on the undersurface forebody. The wings would be computed by a simple shock expansion program or by some impact theory such as Modified Newtonian. In each case, the added cost of the computation would be about 20-25%.

7. Flexible structure capability - In this case a model for the vehicle structure must be included so that the effect of the pressure forces on the structure may be assessed. This would then feed back to the aerodynamic computation to give a new flow field and pressure distribution. The process repeats to a converged structural shape.

8. Improved internal cowl lip analysis - The present cowl lip analysis on the interior of the inlet consists of simply applying two dimensional shock expansion theory at the cowl lip using only the lip incidence angle with the oncoming flow. The intersection with the sidewall shocks in the corners is not currently taken into account. While this region is small, it still accounts for the largest error in this part of the code. Solving for the corner intersections of two shocks requires a full three-dimensional solution. There are several methods which could be used, all of which will require considerably more computation time. Each of these methods use the methods of finite differences and solves some form of the Euler equations such as the methods of Kutler and Lomax [6] or Warming, et. al. [7]. Should such a method be written, it might also replace the module written for the rest of the inlet if it were sufficiently fast. The difficulty with this idea, though, is that all finite difference methods to date have the tendency to smear shocks and could thus might not provide a reasonable solution in a complex shock interaction region. The present inlet program is a shock/expansion method and, though rather cumbersome in its complexity and its inability to do three dimensional problems, it is extremely fast and is near exact for the inviscid equations.

9. Real gas effects and heat transfer computations on exposed surfaces - Since the Mach numbers in realistic situations can be quite high, it may be necessary to make some accounting for high temperature effects such as dissociation and the appearance of ionic forms of the air molecules. One way to do this is to use either equilibrium or frozen gas assumptions. In either case an adjustment is made of the local ratio of specific heats and z-factor in the state equation based on the per cent dissociation expected at a certain temperature [8]. Since the temperature is known locally, the adjustment of the z-factor can be made an extra step in the iteration process at each point. With surface thermodynamic properties available from the Method of Characteristics, there are several procedures available to make estimates of the heat transfer coefficients.

10. Improved Combustion Model - Many improvements in the combustion model can be made at relatively low cost. In particular, effects of the chemistry of teh combustion products should

be a accounted for. Some correction for effects of fuel injection spray on the flow structure should be incorporated. This can be done by utilizing extensions of the shock/expansion routines already available in the program. Effects of variable combutor channel duct area should be included. This is already being done in the present code by a simple extension of the Rayleigh line algorithm.

CONCLUSIONS

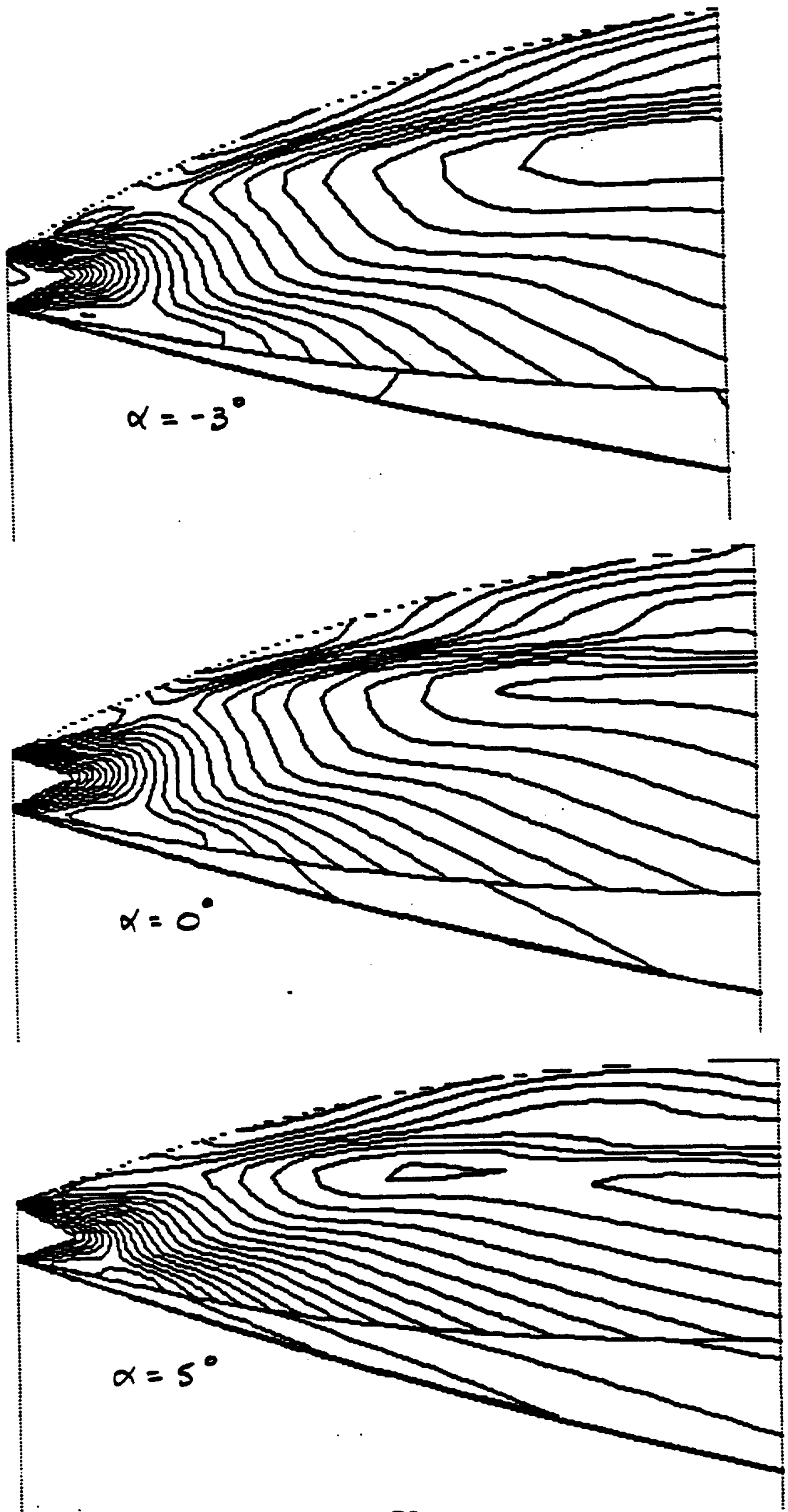
The hypersonic code has been used to compute several cases for this document though comparisons with experimental or other analytical data for validation purposes has not been performed. Considerable improvement of the code will be needed if it is to work for more realistic hypersonic vehicle geometries of the type currently under study by NASA. To every extent possible, validations of individual elements of the code were performed (e.g. frequent use of the isentropic and shock tables - computer generated, of course) and these gave us confidence that the individual modules of the code were faithfully executing their assigned tasks. Many development runs have been made on small computers. The standalone implementations were generated on a Macintosh FX. After extensive debugging, run time has decreased to less than 30 seconds for the method of characteristics forebody solutions and less than 15 seconds for the shock/expansion forebody calculations. Some calculations were performed on a Gateway 2000 486/33 where typical run times were 25 CPUsec. Faster runs are possible since some of the time is used to writeextensive diagnostics and data files. As many as 8400 grid points in the fore and aft body regions were used.

There were, however, several configurations in which the MOC procedure in the aftbody region had difficulties. These tended to be for cases where the Mach number was in the range of 10-13 for the AWST configuration. The reason is believed to have been the compressions resulting from the choice of geometry rather than from an inherent instability in the code. The present configuration has high curvature relatively close to the beginning of the exhaust plane and, as a consequence, resembles a compression curve. This is neither a desirable feature to have in the middle of an MOC procedure since it can lead to instabilities, or worse, nor is this a desirable feature for an aerodynamically optimum vehicle. It should be noted that without a shock imbedding scheme, strong compressions can defeat an MOC calculation by causing instabilities, slow or non-existent convergence, or hanging of the machine. Such compressions have already been seen in some of the contour plots (see Fig. 52 on page 65, for example). Runs at a freestream Mach number of 10 would diverge in the aftbody region as the compression was approached and penetrated. Runs for cases with similar conditions which did manage to finish frequently had losses in mass flux due to the inaccurate capture of the nonisentropic shock. Thus, it seemed that it was the fault of the body shape rather than the method.

To check this, a few runs were made with an alternate aftbody (courtesy of D. McMinn, NASA-Langley). This corresponds to the UX-30 geometry described in the results section. The new nozzle contour has a milder curvature throughout the length of the aftbody significantly reducing the ramp-like structure. As a consequence, the MOCA routines had no problems computing this flow. The contour plots for M=10 at angles of attack of -3, 0, and 5 degrees are shown in Fig. 56. A mild compression still exists in each of the plots, but is properly computed by the MOCA routine which can handle weak compressions. Further, there was no difficulty in preserving mass conservation throughout the entire domain.

We conclude that as long as the non-shock-imbedded Method of Characteristics is used, then the user will have some difficulties with cases where the bodies are not particularly aerodynamically efficient (i.e. the presence of strong compressions cause problems for an MOC code without provision for arbitrary shocks). Other than this difficulty, the code appears to be working for a fairly wide range of Mach numbers (to 25) and angles of attack (as high as possible before the bow shock enters the inlet). We have made preliminary runs throughout these ranges as presented in the results section. A temporary shock/expansion method is used in the current software to handle compressive forebody shapes such as the UX-30 configuration.

Fig. 56. Mach number contours for the UX-30 aftbody at M=10 for various angles of attack.



REFERENCES

1. Chushkin, P.I., "Numerical Method of Characteristics for Three-Dimensional Supersonic Flows," Progress in the Aeronautical Sciences, Vol. 9, 1968, pp. 41-122.

2. Rakich, J.V., "A Method of Characteristics for Steady Three-Dimensional Supersonic Flow with Application to Inclined Bodies of Revolution," NASA TN D-5341, 1969.

3. Fox, L., Numerical Solution of Ordinary and Partial Differential Equations, Pergamon Press, 1962.

4. Doty, R.T. and M.L. Rasmussen, "Approximation for Hypersonic Flow past an inclined Cone," AIAA Journal, Vol. 11, No. 9, Sept. 1973, pp. 1310-1315.

5. Sakowski, B., "Heat Exchanger Design Analysis for In-Flight Liquification of Air in Hypersonic Vehicles," MS Special Problem, School of Aerospace Engineering, Georgia Tech, March 1990.

6. Kutler, P. and H,. Lomax, "Shock-Capturing, Finite-Difference Approach to Supersonic Flows," AIAA J., Vol. 8, No. 12, Dec. 1971, pp. 1175-1182.

7. Warming, R.F., P. Kutler, and H. Lomax, "Second- and Third-Order Noncentered Difference Schemes for Nonlinear Hyperbolic Equations," AIAA J., Vol. 11, No. 2, Feb. 1973, pp. 189-196.

8. Vincenti, W.G. and C.H. Kruger, "An Introduction to Physical Gas Dynamics," J. Wiley & Sons, N.Y., 1965.

9. Waltrup, P. J., Billig, F. S., and Stockbridge, R. D., "A Procedure for Optimizing the Design of Scramjet Engines," J. Spacecraft, Vol. 16, No. 3, May-June 1979.

10. Hsia, Y. H., Daso, E. O., and Padhye, V. A., "FNS Analysis of a 3-D Scramjet Inlet," AIAA Paper No. 91-0129, 29th Aerospace Sciences Meeting, January 1991.

11. Weissmann, D., "Representation of Two-Dimensional Hypersonic Inlet Flows for One-Dimensional Scramjet Cycle Analysis," AIAA Paper No. 90-0527, 28th Aerospace Sciences Meeting, January 1990.

12. Buchmann, O. A., "Thermal-Structural Design Study of an Airframe-Integrated Scramjet," NASA CR-3141, Langley Research Center, 1979.

13. Shaughnessy, J. D., Pinckney, S. Z., McMinn, J. D., et al, "Hypersonic Vehicle Simulation Model: Winged-Cone Configuration," NASA TM-102610, Langley Research Center, 1990.

14. Dash, S. M, Sinha, R. N., Wolf, D. E., and York, B. J., "Computational Model for the Analysis/Design of Hypersonic Scramjet Components - Part 1: Combustor and Nozzle Models," AIAA Paper 86-1595, June 1986.

15. Krawczyk, W. J., Sinha, R. N., Harris, T. B, York, B. J., and Dash, S. M., "Computational Model for the Analysis/Design of Hypersonic Scramjet Components - Part 2: Inlet and Ramp/ Forebody Models," AIAA Paper 86-1596, June 1986.

16. Tsujiakawa, Y., Tsukamoto, Y., and Fujii, S., "Performance Analysis of Scramjet Engines," Bulletin of University of Osake Prefecture. Series A. Engineering and Natural Sciences, Vol. 37, No. 1, 1988.

17. White, M. E., Drummond, J. P, and Kumar, A., "Evolution and Application of CFD Techniques for Scramjet Applications," J. Propulsion Vol. 3, No. 5, Sept.-Oct. 1987.

18. Northam, G. B., and Anderson, G. Y., "Ramjet Research at Langley," AIAA Paper No. 86-0159, 24th Aerospace Sciences Meeting, January 1986.

19. Thomas, J. L., Rudy, D. H., Chakravarthy, S. R., and Walters, R. W., "Patched-Grid Computations of High-Speed Inlet Flows," FED-Vol. 66, Advances and Applications in Computational Fluid Dynamics, ASME Winter Annual Meeting, November 1988.

20. Trexler, C. A., "Inlet Performance of the Integrated Langley Scramjet Module (Mach 2.3 to 7.6)," AIAA Paper No. 75-1212, 11th Propulsion Conference, September 1975.

21. Broadbent, E. G., "Flowfield Calculations for some Supersonic Sections with Ducted Heat Addition," Ingenieur-Archiv Vol. 42, Springer-Verlag 1973.

22. Broadbent, E. G., "Flows With Heat Addition," Progress in Aerospace Science Vol. 17, No. 2, Pergamon Press, 1976.

23. Kumar, A., "Numerical Simulation of Flow Through a Two-Strut Scramjet Inlet," J. Propulsion, Vol. 5, No. 3, May-June 1989.

24. Cheng, Sin-I, "Hypersonic Propulsion," Prog. Energ. Combust. Sci, Vol. 15, 1989.

25. Stalker, R. J., Morgan, R. G., and Netterfield, M. P., "Wave Processes in Scramjet Thrust Generation," Combustion and Flame, Vol. 71, 1988.

26. Schetz, J. A., and Billig, F. S., "Modular Analysis of Scramjet Flowfields," J. Propulsion, Vol. 5, No. 2, Mar.-Apr. 1989.

27. Schindel, L. H., "Design Model of High-Performance Ramjet or Scramjet-Powered Vehicles," J. Spacecraft, Vol. 27, No. 6, Nov.-Dec. 1990.

28. Burkardt, L. A., and Franciscus, L. C., "RAMSCAN - A Flexible Ramjet/Scramjet Engine Simulation Program," ASME 90-GT-323, Gas Turbine and Aeroengine Congress, June 1990

APPENDIX 1
Users Manual for Program HYTHRUST

Forward

This document gives a brief description of the HYTHRUST computer program and defines many of the symbols used. A users guide is also included along with a sample input files and sample outputs.

Abstract

The HYTHRUST program is designed to be used to compute the forces and moments generated on the underside of a hypersonic vehicle including the propulsion system module. The present configuration is based on the NASA baseline vehicle and assumes that the propulsion system consists of a number of SCRAMjet modules which use hydrogen fuel. The program must be run once for each module as the solution is generated for a strip along the length. A number of geometric, propulsion system, and flow parameters may be varied to examine the effect on the force distribution. The aim of the program was to be able to give a relatively accurate assessment of the flow and yet be very fast computationally. The present program runs in about 1-2 minutes on a 486-class microcomputer.

Approach

The computer program is divided into roughly four parts. One each for the forebody, inlet, combustor, and aftbody. Each of these parts is called from the main program which reads the input file, and directs the output. As all variables are stored, the user may examine any number of flow properties by placing appropriate write statements in the corresponding locations in the main program. The present outputs are a breakdown of the forces generated on the main parts of the underbody and one or more data files with flow information in them for graphics output. Some figures are given which used these files are included as an example.

Generally the user's main interaction with the program is through the input data file. This file, HYTH.DAT, contains the freestream information, the forebody geometry, the inlet and strut (up to five struts) geometries, combustion parameters, and the aftbody geometry.

<u>Brief description of the computational methods</u>

A. Vehicle Forebody

The vehicle forebody acts as the primary inlet ramp and provides the first compression of the freestream. As such it is necessarily streamlined to avoid severe stagnation pressure losses. It may also have a complex form if some sort of waveriding analysis was used to generate it's shape. The flow is generally rotational and the location and shape of the bow shock is unknown. Hence, the shock must be computed along with the flow. At present, a 2D Method of Characteristics (MOC) is used to compute the flow and shock. The procedure allows for angle of attack to be included as well as radius of curvature in the crossflow direction. The MOC procedure only needs freestream information, the body geometry, and the data along some initial line normal to the body to get started. The method then marches downstream using compatibility relations derived from the inviscid, rotational equations (Euler) to compute the flow at the next station. The main routines responsible for this part of the code are MOCF, CONEFLOW, and STEP. The routine CONEFLOW is responsible for generating the starting conditions for the MOCF routine and is called as soon as the forebody geometry is known. The routine MOCF does the main part of the work in applying the method of characteristics along the forebody. It accomplishes this by distributing points along vertical lines between the body and the shock whose location is initially guessed by extrapolating its position from the previous station. At each point, the compatibility relations are solved in an iterative fashion to get the properties. At the shock, the oblique shock relations are used along with the appropriate compatibility relation to determine the post shock properties and shock location. The process continues until the inlet face is reached. Generally, the inlet is inclined at an angle and is not coincident with the last station. The routine STEP does the final MOC computations to get the properties at the inlet face.

B. Inlet

The inlet is responsible for the flow compression and deceleration required for the combustor. It tries to accomplish this using oblique shocks in such a fashion as to minimize the stagnation pressure loss. The least expensive method which retains a great deal of accuracy is the oblique/shock expansion theory. This method assumes locally one-dimensional flow between lines represented by shocks, expansions, slip lines, and solid surfaces. To the extent that the flow is characterized by a relatively small number of these regions, the method is an extremely close approximation to an exact solution of the inviscid equations. The main difficulty is the logic in keeping track of locations of the lines, their next intersection and the computation of the properties downstream of the intersections. The present version of the code divides the SCRAMjet inlet into a number of vertical streamtubes. The number of streamtubes depends on how much the stagnation pressure varies normal to the body surface. In each streamtube, the flow is assumed to enter the inlet with constant Mach number and stagnation pressure. From there, the method computes the sidewall shock angles and the first intersection of these shocks with each other or with one of the struts. The downstream marching procedure keeps track of the locations of all lines, flow properties in between the lines, and where the next intersection of any kind will occur. The process continues until the beginning of the combustor (user specified location) is reached. At this point, the stagnation pressure and Mach number are area-averaged for the streamtube for use by the combustor section. While the idea is a relatively simple one, the present implementation of the numerous possible intersections, computation of the propetries downstream and so forth take up about 60% of the code. This part of the code starts at the subroutine INLET. This routine, makes calls to a number of utility routines (namely UPEXP, LOWEXP, BSTRUT, ESTRUT, SSDF, SSSF, SEDFA, SEDFB, SESF, EEDF, EESF, SRSPA, SRSPB, SRW, ERSPA, ERSPB, and ERW) which mainly compute properties downstream of the various kinds of intersections. The names of these utility routines are dervied from the kind of intersection which has occured. For example, SSDF means "Shock-Shock, Different Family," and refers to the intersection of two shocks of different families. Two other routines, SHANG and PM are utilities which compute the angles and properties downstream of an oblique shock and a Prandtl-Meyer expansion respectively. These routines are also called by some of the other sections.

External to the inlet, at the cowl lip is a Prandtl-Meyer expansion. The flow properties are computed in MAIN since this is a simple procedure and results are held for the aftbody computation.

C. Combustor

In keeping with the simplicity of methods, the simplest manner in which a flow may be influenced by the addition of energy from chemical reaction is through the Rayleigh line analysis. This is also a one dimensional flow computation with a specified amount of heat addition in the routine RAYLEIGH. Alternatively, the outflow Mach number can be specified and the amount of heat addition computed. In either case, the amount of heat addition is related to the fuel-air equivalence ratio (ratio of mass flow rates of the fuel and air). A number of parameters may be varied in this section, such as the mixture gas properties, combustor outflow Mach number or equivalence ratio. A maximum temperature is also specified and the procedure does not allow this to be exceeded. Once the heat addition has been applied to each of the stream tubes, the combustor outflow Mach number, stagnation pressure, stagnation temperature, and static quantities are available for the computaton of the flow over the aftbody.

D. Aftbody

The aftbody computation is performed totally in MOCA. Starting conditions are computed from the outflow of the combustor and from the properties computed from the expansion around the cowl lip. The basic procedure is similar to the forebdy with the exception that the external stream has a much different stagnation enthalpy than the combustor exhaust as well as a different pressure. The pressure difference and Mach numbers are used to determine the initial shock orientation from the combustor exhaust along with the slip line angle between the two streams. The routine then marches downstream in a manner similar to the forebody with the addition of an iterative procedure for the slip line orientation as well as the shock. The procedure ceases when the end of the vehicle is reached.

E. Force and moment computation

After the MOCA routine has completed the aftbody flow computation, the MAIN program presents a summary of the forces and moments applied to the vehicle by the propulsion system. Two methods are available within the program to determine these quantities. The first employs integration of the pressure forces over all wetted surface areas defined as part of the propulsion system. This includes inlet wedges, struts, and cowl surfaces. The second method follows and extension of the scramjet propulsion/ airframe integration definitions of Billig (Reference). The momentum method is used to determine the forces on a control volume described by the stream tube passing through the combustor duct. The fore and aft boundaries are the inlet surface and a plane normal to the flight direction passing through the most rearward point of the airframe. The nozzle slip line emanating from the lower edge of the cowl forms the lower boundary of the control volume.

The program computes the components of the thrust vector defined in the standard way. Thrust is positive in the direction of flight. The lift component of the thrust vector is positive upward. The thrust vector angle is positive if the vector is above the flight plane. The line of action of the thrust vector is computed as the point of intersection of the vector with the vehicle reference x-axis. The moment of the thrust force about the origin (located at the nose) of the vehicle x-y system is also determined. The moment is positive in the pitchup direction.

Input Data File

As mentioned, the main interface the user has is with the input data file. A sample data file is shown in Appendix 2. There are a several other places where the user may wish to change some default choices and these will be described in a later section. The input data file contains most of the parameters that the user may wish to vary in any parametric study and hence the program can usually be treated as a stand-alone module. Below a description of the input module is given. Variable names appropriate to each of the variables is included at the bottom of the data file for the convenience of the user.

Freestream conditions - Line 1. AMINF, TINF, PINF, ALPHAD

These are, respectively, the freestream Mach number, static temperature in degrees Rankin, static pressure in pound-force per square foot, and vehicle angle of attack in degrees. The stagnation conditions will be computed from these quantities and the gas properties given in line 2. The angle of attack is the angle between the freestream and the main wing chord which would be the horizontal axis if the angle of attack were zero. In the sample data file, the temperature and pressure were those typical at about 120,000 ft altitude.

Freestream gas properties - Line 2. G, RG

These are the ratio of specific heats and the gas constant for the freestream gas. The gas constant is the universal gas constant divided by the molecular weight of the gas. For air, the values are 1.4 and 1716 ft-pound-force/slug-degree Rankin.

Number of forebody points - Line 3. NPF

This is the number of coordinate points the user wishes to use to describe the geometry of the forebody on the strip (Number of Points on Forebody). These points should lie directly on the surface. The program will use a parametric cubic spline routine to interpolate between these points which is a rather smooth procedure. Thus, it does not require many points to describe a shape with few inflection points. in the sample data file, only about 5 points were used.

Forebody data - Lines 4 (there will be NPF of these). X(i),Y(i),C(i)

These are the forebody coordinate points (in feet) and body curvature. The coordinates are defined with respect to the nose of the vehicle with the x-axis parallel to the main wing chord. The vehicle is assumed to be upside down in order to have the vertical coordinates positive, though this did not turn out to be the advantage originally thought. Thus the y-axis may be thought of as being downward. The curvature of the surface, defined as simply the inverse of the radius of curvature in a plane perpendicular to the freestream. This is necessary in order to allow the MOC procedure to handle angles of attack. Note that is the body were a cone, the curvature would be the inverse of the y-coordinate. If the forebody is flat in cross section, the local curvature is zero (since the radius of curvature for a flat surface is infinite). All coordinates and curvatures are redefined in the program for the given angle of attack and leaves the freestream parallel to the horizontal axis. That way the user never needs to change the surface coordinates while changing the angle of attack.

Inlet print logical - Line 5. LOUT

This is a logical variable used to turn off or on the generation of various output data files in the inlet. If LOUT is true, then the files are written, if false, they are not. As these files were mainly usd for debug purposes, the user does not generally need the information contained in them and should set LOUT to false.

Inlet dimensions - Line 6. H, XW1, XW2

The variable H is the width of the combustor inlet (feet). XW1 and XW2 are the distances from the inlet face to the max thickness points of the sidewall ramps (feet).

Sidewall wedge angles - Line 7. D1,D1ST,D2,D2ST

D1 and D2 are the initial sidewall ramp angles. D1ST and D2ST are the sidewall ramp angles after the max thickness points. All angles are in degrees and are specified with respect to a vertical plane parallel to the freestream.

Cowl Lip and combustor exhaust locations - Line 8. XCL,YCL,XEX,YEX
These are the coordinates of the cowl lip and the outer edge of the combustor exhaust (feet).

Combustion chamber location, number of struts, strut angle - Line 9. XCC,NSTRUT,ALSTD
XCC is the distance (feet) from the inlet face of the beginning of the combustion chamber. It is this location that determines when the shock-expansion computation of the inlet ceases and the Rayleigh line analysis begins. NSTRUT is the number of struts in the SCRAMjet up to a maximum of 5. ALSTD is the angle that the struts make with respect to the vertical axis in degrees.

Strut configuration - Lines 10 (NSTRUT of these). XD(i),YD(i),DT(i),DD(i),DELST(i)
XD and YD are the relative positions of the strut leading edges with respect to the leading edge of sidewall 1 (feet). DT and DD are the length and thickness of the struts (feet). In this version of the INLET subroutine, the struts are considered to be rhombi. DELST is the angle that the major axis of the strut makes with respect to a vertical plane parallel to the freestream.

Combustion gas properties - Line 11. G2,RG2,QR,ETAB,FCRIT
These are the mixture ratio of specific heats and gas constant (ft-pound force/slug-degree Rankin). QR is the heating value of the fuel (.................), ETAB is the combustor efficiency factor, and FCRIT is the maximum fuel-air equivalence ratio for the fuel.

Combustor operation parameters - Line 12. F,AM4
These are the fuel-air equivalence ratio and the combustor outflow Mach number. The user specifies one of these. The idea of the combustor is to add the maximum amount of heat possible. This drives the flow toward a Mach number of unity (adding more heat will choke the flow, reducing the mass flow rate and possibly causing an unstart or, at least, greater inlet spillage resulting in high drag). Thus, it is desirable to have AM4 be as close to unity as possible. The user, though, more often has control over the fuel-air ratio and may wish to specify F. If F is to be given, then the value of AM4 is ignored. If the user wishes to specify AM4, then the value of F should be set to zero.

Number of aftbody points - Line 13. NPA
Self-explanatory. Analogous to NPF, the number of forebody points.

Aftbody geometry - Lines 14 (NPA of these). XA(i),YA(i),CA(i)
Completely analogous to the forebody data. Again, parametric cubic splines will be used to interpolate the data.

Other User-control Variables

In addition to the input data file, there are a few places in the program where the user may wish to modify some of the program control parameters. These are described below roughly in order of appearance in the program.

1. Number of vertical stations for the MOC procedures. This is the integer JM set in a PARAMETER statement in MAIN and in several other routines. This number depends on the level of accuracy desired. The current use of 15 represents a reasonable number of vertical stations. Naturally, more stations could be used and may be necessary if the geometry is more rapidly varying that the sample data. However, if the forebody data is no more complex than the present configuration, then 15 has been shown to give reasonable results. More stations does not appear to greatly change the overall result.

2. Equivalent cone angle for starting the forebody MOC. Currently, the program is set to create a cone tangent to the forebody at 5% of the forebody length. This is a completely arbitrary choice. It was chosen to be far enough downstream so that the number of MOC streamwise stations would not be excessive. The number does depend, to some extent, on the closeness of the shock to the body. If a smaller % were chosen, then there would necessarily be more stations. If this is desirable, then the user may need to enlarge the dimensions of the stored variables.

The Routines in HYTHRUST

Most of the routines in the program have some descriptive comment at the beginning. The following list briefly summarizes these routines for the convenience of the user.

MAIN - overall program control. This part calls the routines which compute the four main segments of the flow field.

MOCF - computes the forebody flow and bow shock by using a Method of Characteristics procdedure similar to the Hartree method.

CONEFLOW - called by MOCF. Provides the starting cone and flow conditions for the MOCF procedure.

LAGRANGE - an interpolation routine used in providing a means to evaluate flow variables in between the values at the previous station.

CUBIC - a parametric cubic spline interpolation procedure. In this routine, a single dimensioned array (like X(i)) is given to the routine. The coefficients of the cubic spline interpolating functions are returned. The interpolating functions are, of course cubics, in the parameter, t, which varies between zero and one on each cubic.

INTRPC - finds the value of the parameter, t, and between which two points X(i) some arbitrary value of X exists. This value of t can then be used in the cubic splines of the other variables to find their corresponding values.

STEP - advances the forebody MOC net to the face of the inlet. This routine is necessary since the MOCF routine only advances the forebody flow calculation using vertical stations and the inlet face is, in general, not vertical. Most of the same logic as MOCF is used in STEP, but the geometry is different.

INLET - computes the flow in the inlet by the shock-expansion method. This is the largest part of the program using 60% of the coding. A number of utility routines are called by INLET since there are a large number of possible types of intersections between shocks of two families, expansions of two families, slip lines, and solid surfaces.

APPENDIX 2
Typical Data File
(Represents Configuration Shown in Figure 1)

NASP.dat	
5. 418.79 23.085 8.	AMinf Tinf Pinf Alpha(deg)
1.4 1716.	Gamma Rgas
5	NPF
0. 0. 0.	XF(1) YF(1) CF(1)
28.31 5.08 .19685	XF(2) YF(2) CF(2)
52.54 8.6 .11628	
76.27 11.01 .090827	... ...
98.31 11.86 .084317	XF(NPF) YF(NPF) CF(NPF)
FALSE	Lout
7.35 15.385 15.385	H XW1 XW2
6. -6. -6. 6.	D1 D1ST D2 D2ST
103.5 17.06 134.27 17.06	XCL YCL XEX YEX
15.553 3 35.	XCC NST ALST
13.561 3.675 2.727 .2867 0.	XD(1) YD(1) DT(1) DD(1) DELST(1)
12.796 2.345 4.101 .3587 3.	XD(2) YD(2) DT(2) DD(2) DELST(2)
12.796 2.345 4.101 .3587 -3	XD(NST) YD(NST) DT(NST) DD(NST) DELST(NST)
1.4 1716. 1.2E9 .99 .0292	G2 RG2 QR EtaB FCRIT
0. 1.15	F AM4
6	NPA
134.27 11.86 0.	XA(1) YA(1) CA(1)
146.47 6.53 0.	XA(2) YA(2) CA(2)
158.67 2.14 0.	... ...
170.87 -1.15 0.	
183.07 -3.37 0.	
195.27 -4.74 0.	XA(NPA) YA(NPA) CA(NPA)

Key:

Symbol Table corresponds to values given above that are read by the program. Propellant is liquid hydrogen, ER is equivalence ratio. Program computes mixture properties, but burns only enough H2 to stay within temperature limits. Static Temperature is not allowed to exceed 5500 degrees inside propulsion system duct. Flight condition shown corresponds to standard atmosphere at 100,000 ft.

APPENDIX 2 (continued)
Typical Data File
(Represents Configuration Shown in Figure 21)

UX-30.dat

5. 406.44 27.83 8.	AMinf Tinf Pinf Alpha(deg)
1.4 1716.	Gamma Rgas
5	NPF
0. 0. 0.	XF(1) YF(1) CF(1)
53.367 4.667 0.0	XF(2) YF(2) CF(2)
70.392 6.983 0.0	
79.575 8.742 0.0	... ...
92.833 12.117 0.0	XF(NPF) YF(NPF) CF(NPF)
FALSE	Lout
33.333 1.0 1.0	H XW1 XW2
0.5 -0.5 -0.5 0.5	D1 D1ST D2 D2ST
92.833 12.875 100.2 12.875	XCL YCL XEX YEX
4.50 3 35.	XCC NST ALST
1.0 1.0 1.0 1.0 0.0	XD(1) YD(1) DT(1) DD(1) DELST(1)
1.0 1.0 1.0 1.0 1.0	XD(2) YD(2) DT(2) DD(2) DELST(2)
1.0 1.0 1.0 1.0 -1.0	XD(NST) YD(NST) DT(NST) DD(NST) DELST(NST)
1.26 1716. 1.2E9 1.00 .02916	G2 RG2 QR EtaB FCRIT
0. 1.15	F AM4
7	NPA
100.2 12.117 0.	XA(1) YA(1) CA(1)
108.51 8.7932 0.	XA(2) YA(2) CA(2)
116.80 5.9963 0.	... ...
125.10 3.7167 0.	
133.40 1.9574 0.	
141.70 0.71852 0.	XA(NPA) YA(NPA) CA(NPA)
150.00 0.00 0.	

Key:

Symbol Table corresponds to values given above that are read by the program. Propellant is liquid hydrogen, ER is equivalence ratio. Program computes mixture properties, but burns only enough H2 to stay within temperature limits. Static Temperature is not allowed to exceed 5500 degrees inside propulsion system duct. Flight condition shown corresponds to standard atmosphere at 95,636 ft.

APPENDIX 3
Program Source Listing

```
*______________________________________________________________________
                         PROGRAM HYTHRUST
*
*                          VERSION 1.0
*                           July 1992
*
*        This program determines the effect of angle of attack on thrust
*        vector of a scramjet propelled vehicle. Method of characteristics
*        is used for the fore and aft body regions with shock-expansion and
*        Rayleigh line methods for the scramjet duct. Output includes
*        thrust vector magnitude, direction, and line of action, fuel ISP,
*        Pitching moment due to thrust (positive pitch up), and effective
*        equivalence ratio (portion of the fuel flow actually burned).
*
*        The equivalence ratio and a range of flight Mach number and angle
*        of attack are chosen by the user. Propellant is liquid hydrogen.
*        Properties of the atmosphere at selected altitude, vehicle geometry,
*        and gas properties are read from a data file that can be modified
*        as needed by the user.
*
*
*    Written by:
*
*                 Robert L. Roach and Harald Buschek,
*                  Georgia Insitute of Technology
*                            Atlanta, GA
*
*                          Gary A. Flandro,
*                 University of Tennessee Space Institute
*                            Tullahoma, TN
*
*    For:         NASA Langley Research Center
*                            Hampton, VA
*______________________________________________________________________
*
*  Main Externals:
*   CUBIC     gets the cubic spline coeffs
*   INTERPC   interpolates cubic
*   CONEFLOW  starting conditions based on Taylor-Maccoll theory
*   LAGRANGE  4 pt Lagrange interpolation
*
*______________________________________________________________________

      PARAMETER    (IM=501,JM=21,JI=21,NST=5,NGMAX=23)

      COMMON/CBSPF/ AFX(IM),BFX(IM),CFX(IM),AFY(IM),BFY(IM),CFY(IM),
     1         AFC(IM),BFC(IM),CFC(IM),NPF
      COMMON/CBSPA/ AAX(IM),BAX(IM),CAX(IM),AAY(IM),BAY(IM),CAY(IM),
     1         AAC(IM),BAC(IM),CAC(IM),NPA
      COMMON/DEPVR/ RHO(IM,JM),VT(IM,JM),T(IM,JM),P(IM,JM),
     1         TH(IM,JM),AM(IM,JM),WS(IM),H0(IM,JM),P0(IM,JM)
```

```
      COMMON/CMBST/ G2,RG2,CP2,G2M,G2P,QR,ETAB,FCRIT
      COMMON/GASPR/ G,RG,GM,GP,CP
      COMMON/GRIDS/ X(IM),Y(IM,JM)
      COMMON/GRIDF/ XF(IM),YF(IM),CF(IM),ILF
      COMMON/GRIDA/ XA(IM),YA(IM),CA(IM),ILA
      COMMON/INFTY/ AMINF,PINF,TINF,ALPHA
      COMMON/INLTI/ P0IN(JI),AMIN(JI),PIN(JI),TIN(JI),RHIN(JI),
     1          VTIN(JI),THIN(JI),XIN(JI),YIN(JI)
      COMMON/INLTC/ P0CC(JI),AMCC(JI),PCC(JI),TCC(JI),RHCC(JI),
     1          VTCC(JI),H0CC(JI)
      COMMON/EXITC/ P0CX(JI),AMCX(JI),PCX(JI),TCX(JI),RHCX(JI),
     1          VTCX(JI),H0CX(JI)

      COMMON/INLTF/ DFUW,DFLW
      DIMENSION JSTRUT(NGMAX)
      COMMON/GEO/ H,H1,XW1,XW2,D1,D1ST,D2,D2ST,XD(NST),YD(NST),
     1          DT(NST),DD(NST),XCC,DELST(NST),JSTRUT,ALST
      COMMON/SCRMX/ P0EX(JI),AMEX(JI),PEX(JI),TEX(JI),RHEX(JI),
     1          VTEX(JI),THEX(JI),YCX(JI),H0EX(JI)
      COMMON/STAGN/ T0INF,H0INF
      COMMON/STRUT/ NSTRUT

      DIMENSION FF(JI),XS(IM),YS(IM)
              DIMENSION  PUTOUT(14)
      CHARACTER  DATNAM*21, FILNAM*21

      LOGICAL  LOUT,REVY

      write(6,*) ' Enter Minimum Equivalence Ratio (ERmin)'
      read(6,*) ER     write(6,*) ' Enter Mach Number Range (Mstart,Mfinal,DeltaM)'

      read(6,*) AMSTART,AMFINAL,AMINCREMENT
      write(6,*) ' Enter Angle of Attack Range (Astart,Afinal,DeltaA)'

      read(6,*) ASTART,AFINAL,AINCREMENT
      write(6,*) ' Enter Plot File Name [type NONE if none reqd.] '

      READ(6,'(A21)') FILNAM
      IF (FILNAM.EQ.'NONE') THEN
         NOPE = 1
      ELSE
         OPEN(12,FILE=FILNAM,STATUS='UNKNOWN')
      END IF
      WRITE(6,*),' Enter Input data file name: '
      READ(6,'(A21)') DATNAM
      WRITE(6,*), '     SCRAMJET THRUST VECTOR ANALYSIS'
      WRITE(6,*) '______________________________________________________'

      DO AMACHNUM = AMSTART,AMFINAL,AMINCREMENT
         DO ANGLEOFATK = ASTART,AFINAL,AINCREMENT
         CALL SCRAMJET(AMACHNUM,ANGLEOFATK,ER,lever,DATNAM)
              IF (lever.GT.0) EXIT
         END DO
      END DO *________________________________________________________
         IF (NOPE.EQ.0) THEN
```

```
                CLOSE(12,STATUS='KEEP')
          END IF
          STOP
      END

*-----------------------------------------------------------------------
          SUBROUTINE SCRAMJET(AMACHNUM,ANGLEOFATK,ER,lever,DATNAM)
*
*
*  This subprogram computes effect of angle of attack on thrust vector
*  of a scramjet propelled vehicle. MOC is used for the
*  fore and aft body regions with shock-expansion and
*  Rayleigh line methods for the scramjet duct. Output includes
*      thrust vector magnitude, direction, and line of action, fuel ISP,
*      Pitching moment due to thrust (positive pitch up), and equivalence
*      ratio.
*
*     Main Externals:
*      CUBIC     gets the cubic spline coeffs
*      INTERPC   interpolates cubic
*      CONEFLOW  starting conditions based on Taylor-Maccoll theory
*      LAGRANGE  4 pt Lagrange interpolation
*
*-----------------------------------------------------------------------

      PARAMETER  (IM=501,JM=15,JI=21,NST=5,NGMAX=23)

COMMON/CBSPF/ AFX(IM),BFX(IM),CFX(IM),AFY(IM),BFY(IM),CFY(IM),
   1          AFC(IM),BFC(IM),CFC(IM),NPF
COMMON/CBSPA/ AAX(IM),BAX(IM),CAX(IM),AAY(IM),BAY(IM),CAY(IM),
   1          AAC(IM),BAC(IM),CAC(IM),NPA
      COMMON/DEPVR/ RHO(IM,JM),VT(IM,JM),T(IM,JM),P(IM,JM),
   1          TH(IM,JM),AM(IM,JM),WS(IM),H0(IM,JM),P0(IM,JM)
      COMMON/CMBST/  G2,RG2,CP2,G2M,G2P,QR,ETAB,FCRIT
      COMMON/GASPR/ G,RG,GM,GP,CP
      COMMON/GRIDS/ X(IM),Y(IM,JM)
      COMMON/GRIDF/ XF(IM),YF(IM),CF(IM),ILF
      COMMON/GRIDA/ XA(IM),YA(IM),CA(IM),ILA
      COMMON/INFTY/ AMINF,PINF,TINF,ALPHA
COMMON/INLTI/ P0IN(JI),AMIN(JI),PIN(JI),TIN(JI),RHIN(JI),
   1          VTIN(JI),THIN(JI),XIN(JI),YIN(JI)
COMMON/INLTC/ P0CC(JI),AMCC(JI),PCC(JI),TCC(JI),RHCC(JI),
   1          VTCC(JI),H0CC(JI)
COMMON/EXITC/ P0CX(JI),AMCX(JI),PCX(JI),TCX(JI),RHCX(JI),
   1          VTCX(JI),H0CX(JI)
      COMMON/INLTF/ DFUW,DFLW
      DIMENSION JSTRUT(NGMAX)
      COMMON/GEO/ H,H1,XW1,XW2,D1,D1ST,D2,D2ST,XD(NST),YD(NST),
   1          DT(NST),DD(NST),XCC,DELST(NST),JSTRUT,ALST
COMMON/SCRMX/ P0EX(JI),AMEX(JI),PEX(JI),TEX(JI),RHEX(JI),
   1          VTEX(JI),THEX(JI),YCX(JI),H0EX(JI)
      COMMON/STAGN/ T0INF,H0INF
      COMMON/STRUT/ NSTRUT
```

```
      DIMENSION FF(JI),XS(IM),YS(IM)
         DIMENSION      PUTOUT(14)
      CHARACTER*1 ZZZZ
      CHARACTER  DATNAM*21,FILNAM*21

      LOGICAL     LOUT,REVY

      CALL InitBeachBall(512)
      CALL ShowBeachBall

*-------------- USEFUL STUFF ----------------------
      PI = 3.141592654
         lever = 0 *---------------------------------------------------*
   READ INPUT DATA
*--------------------------------------------------

      OPEN(UNIT=2,FILE= DATNAM ,STATUS='UNKNOWN')

*---------- READ FREESTREAM CONDITIONS
      READ(2,*) AMINF,TINF,PINF,ALPHAD

*---------- READ FREESTREAM GAS PROPERTIES READ(2,*) G,RG

*---------- READ FOREBODY DATA
    READ(2,*) NPF
    DO 50 I = 1,NPF
          READ(2,*) XF(I),YF(I),CF(I)

   50     CONTINUE

*---------- READ INLET & COMBUSTOR GEOMETRY READ(2,*) LOUT
    READ(2,*) H,XW1,XW2
READ(2,*) D1,D1ST,D2,D2ST
               WIDTH = H
    D1    = PI*D1/180.
    D1ST = PI*D1ST/180.
    D2    = PI*D2/180.
    D2ST = PI*D2ST/180.

    READ(2,*) XCL,YCL,XEX,YEX
    READ(2,*) XCC,NSTRUT,ALSTD
*    WRITE(6,*),'   ALST = ',ALSTD
    DO 54 I = 1,NSTRUT
    READ (2,*) XD(I),YD(I),DT(I),DD(I),DELST(I) 54    CONTINUE

    ALST = PI*ALSTD/180.
    EL    = XEX - XCL

*---------- NORMALIZE INTERNAL INLET DIMENSIONS WRT H H1      = H
    XW1  = XW1/H
    XW2  = XW2/H
    XCC  = XCC/H
    DO 56 I = 1,NSTRUT
          XD(I) = XD(I)/H
```

```
          YD(I) = YD(I)/H
          DT(I) = DT(I)/H
          DD(I) = DD(I)/H
   56     CONTINUE
    H =   1.

*—————READ COMBUSTION GAS PROPERTIES READ(2,*) G2,RG2,QR,ETAB,FCRIT

*————— READ COMBUSTOR OUTFLOW CONDITIONS READ(2,*) F,AM4

*————— READ AFTBODY GEOMETRY
    READ(2,*) NPA
    DO 60 I = 1,NPA
READ(2,*) XA(I),YA(I),CA(I)
   60     CONTINUE
          YHITE = ABS(YA(2)-YA(1)) CLOSE(2,STATUS='KEEP')

                AMINF = AMACHNUM
                ALPHAD = ANGLEOFATK

ALPHA = PI*ALPHAD/180.
                FLTMACH = AMINF
                DYNAMP = 0.5*G*PINF*FLTMACH**2.
                T0INF = TINF*(1. + .5*(G-1.)*FLTMACH**2)

                WRITE(6,*) ' '
WRITE(6,*) 'Flight Conditions:' WRITE(6,*) '———————————— ' WRITE(6,950)
FLTMACH,ALPHAD
    WRITE(6,951) TINF,T0INF,PINF
                WRITE(6,970) DYNAMP
                WRITE(6,*) ' '
  950 FORMAT ('',' Minf =',F5.2,7X,' Alpha =',F5.2,'°')
951 FORMAT ('',' Tinf =',F7.2,' °R',3X,' Toinf =',F8.0,' °R',3X,' Pinf =',F7.3,' psf')
  970 FORMAT ('',' Dynamic Pressure = q =',F10.3,' psf')


*————————————————————————————————
*           SOME USEFUL QUANTITIES
*————————————————————————————————
    GM   = G - 1.
    GP   = G + 1.
    G2M  = G2 - 1.
    G2P  = G2 + 1.
    RHOINF = PINF/(RG*TINF)
    V1   = AMINF*SQRT(G*RG*TINF)
    VINF = V1
    T0INF  = TINF*(1. + .5*GM*AMINF**2)
    P0INF  = PINF*(1. + .5*GM*AMINF**2)**(G/GM) CP  = G*RG/GM
    H0INF  = CP*T0INF

*——————————————————————————————————————————* ROTATE
ALL COORDINATES THROUGH ANGLE OF ATTACK
*——————————————————————————————————————————CAL = COS(ALPHA)
```

```
      SAL = SIN(ALPHA)

*-------- FOREBODY POINTS
      DO 70 I = 1,NPF
            XX = XF(I)
            YY = YF(I)
            XF(I) = XX*CAL - YY*SAL
            YF(I) = XX*SAL + YY*CAL
                  CALL SpinBeachBall(1)

   70       CONTINUE

*-------- COMBUSTOR POINTS
      XX = XCL
      YY = YCL
      XCL = XX*CAL - YY*SAL
      YCL = XX*SAL + YY*CAL
      XX = XEX
      YY = YEX
      XEX = XX*CAL - YY*SAL
      YEX = XX*SAL + YY*CAL

*-------- AFTBODY POINTS
      DO 78 I = 1,NPA
            XX = XA(I)
            YY = YA(I)
            XA(I) = XX*CAL - YY*SAL
            YA(I) = XX*SAL + YY*CAL
                  CALL SpinBeachBall(1)

   78       CONTINUE

*---------------------------------------------------------------
*     ADJUST SURFACE CURVATURES FOR ANGLE OF ATTACK
*---------------------------------------------------------------
*      WRITE(6,*),'  Adjusting surface curvatures for angle of attack..'

*-------------- FOREBODY
      DO 82 I = 1,NPF
CF(I) = CF(I)/CAL
   82       CONTINUE

*-------------- AFTBODY
            DO 84 I = 1,NPA
            CA(I) = CA(I)/CAL
   84       CONTINUE

*---------------------------------------------------------------
*     GET CUBIC SPLINE COEFFS FOR THE FORE & AFT BODIES
*---------------------------------------------------------------
*     WRITE(6,*),'  Getting cubic spline coeffs for the fore & aft..'
            CALL CUBIC(NPF,XF,AFX,BFX,CFX)
            CALL CUBIC(NPF,YF,AFY,BFY,CFY)
            CALL CUBIC(NPF,CF,AFC,BFC,CFC)
```

```
          CALL CUBIC(NPA,XA,AAX,BAX,CAX)
          CALL CUBIC(NPA,YA,AAY,BAY,CAY)
          CALL CUBIC(NPA,CA,AAC,BAC,CAC)

*------------- OPEN BODY SURFACE DATA FILE -------------
*         OPEN(55,FILE='surface.dat',STATUS='UNKNOWN')

*---------------------------------------------------------------------*
     VEHICLE FOREBODY FLOW COMPUTATION
*---------------------------------------------------------------------*
WRITE(6,*),'  Calling MOCF....'
          CALL MOCF
*    WRITE(6,*),'  Back from MOCF...'

*-------- CHECK TO SEE IF SHOCK INTERSECTS COWL LIP ------YSHK = Y(ILF,JM)
+ (XCL - X(ILF))*TAN(WS(ILF)) IF(YSHK.LT.YCL) THEN
WRITE(6,*),'  Bow Shock is Inside Cowl Lip' lever = 1
              GOTO 3334
          ENDIF

*---------------------------------------------------------------------
*              SCRAMJET INLET
*---------------------------------------------------------------------*------ STEP FROM
FOREBODY FLOW TO COWL LIP USING MOC
          DXINLT = XCL - X(ILF)
          DYINLT = ABS(YCL - Y(ILF,1))
          YP     = YCL

*   WRITE(6,*),'  Calling STEP for Cowl Lip...'
CALL STEPMOC(DXINLT,YP,AMCL,PCL,TCL,RHCL,THCL,VTCL)
*   WRITE(6,*),'  Back from STEP...'

          P0CL   = PCL*(1. + .5*GM*AMCL**2)**(G/GM)

*------ DETERMINE THE NUMBER OF SCRAMJET STREAMTUBES NEEDED
*   Note: NSTR = Number of streamtubes, user selectable or automatic.
*          Determined by percent change in P0, 1 per SLPC change.
*   WRITE(6,*),'  Finding number of streamtubes...'
          SLPC = .005
          DLP0 = ABS(P0CL - P0(ILF,1))/P0(ILF,1)
          NSTR  = INT(DLP0/SLPC)
          IF(NSTR.LT.6)  NSTR = 6
          IF(NSTR.GT.15) NSTR = 15

*   WRITE(6,*),'    There will be ',NSTR,' streamtubes...'

*---------------------------------------------------------------------*
    STREAMTUBE LOOP
*---------------------------------------------------------------------AREAINLET = 0.
   FLXMSS =    0.
   FLXMTMX = 0.
   FLXMTMY = 0.
   FLXMTMA = 0.
PFORCEI = 0.
```

```
        AREAST = DYINLT*H1/NSTR

        DO 200 N = 1,NSTR
              FN = (N - .5)/NSTR
XIN(N) = X(ILF) + FN*(XCL - X(ILF)) YIN(N) = Y(ILF,1) + FN*(YCL - Y(ILF,1)) YCX(N) =
YIN(N)

*—— STEP TO INLET FACE OF THIS STREAMTUBE USING MOC *     Note: Here, the
conditions at the center of the *          streamtube are computed using MOC.
              YY = YIN(N)
              DX = XIN(N) - X(ILF)
              CALL STEPMOC(DX,YY,AMN,PN,TN,RHN,THN,VTN)

P0IN(N) = PN*(1. + .5*GM*AMN**2)**(G/GM) AMIN(N) = AMN
              PIN(N)  = PN
              TIN(N)  = TN
              RHIN(N) = RHN
              THIN(N) = THN
              VTIN(N) = VTN

*———— Compute Mass & Momentum Fluxes

DMASS  = RHN*AREAST*VTN*COS(THN) DMSSX = -DMASS*VTN*COS(THN) DMSSY
= -DMASS*VTN*SIN(THN)
DMMA = XIN(N)*DMSSY - YIN(N)*(DMSSX + PN*AREAST)

              FLXMSS  = FLXMSS + DMASS
              FLXMTMX  = FLXMTMX + DMSSX
              FLXMTMY  = FLXMTMY + DMSSY
              FLXMTMA = FLXMTMA + DMMA

PFORCEI = PFORCEI + PN*AREAST
                    AREAINLET = AREAINLET + AREAST

*———— CALL INLET FOR THIS STREAMTUBE
              AML = AMIN(N)
              PTL = P0IN(N)
              THDG = 180.*THN/PI
*   WRITE(6,*),N,RHN,VTN,THDG,DMASS

*  WRITE(6,*),'        AML = ',AML,'  PTL = ',PTL
              F1  = (1. + .5*GM*AML**2)**(.5*GP/GM)
              SQSTF = SQRT(G/(RG*T0INF))
              FLX    = PTL*AREAST*AML*SQSTF/F1
*  WRITE(6,*),'  Flux for streamtube at start of inlet = ',FLX
*  WRITE(6,*),'  DMASS = ',DMASS

              CALL INLET(N,AML,PTL,AREAST,AM3,P03,AREA3)

*——— COMPUTE INLET EXIT CONDITIONS FOR THIS STREAMTUBE
*  WRITE(6,*),'        AM3 = ',AM3,'  P03 = ',P03
              AMCC(N) = AM3
              TRCC     = 1. + .5*GM*AM3**2
              TCC(N)  = T0INF/TRCC
```

```
  P0CC(N) = P03
PCC(N)  = P03/TRCC**(G/GM) RHCC(N) = PCC(N)/(RG*TCC(N)) ACC
   = SQRT(G*RG*TCC(N))
    VTCC(N) = AM3*ACC
    H0CC(N) = H0INF

*———— SCRAMJET COMBUSTOR
*      WRITE(6,*),'      Calling RAYLEIGH...'

CALL RAYLEIGH(N,AM3,P03,T0INF,F,AM4,P04,T04,T4,P4) AREA4 = AREA3
*                  WRITE(6,*) N,AM4,T04,T4,P4,P04,F

                   FF(N)  = F
    AMCX(N) = AM4
    TRCX         = 1. + .5*G2M*AM4**2
    TCX(N)  = T4
    P0CX(N) = P04
    PCX(N)  = P04/TRCX**(G2/G2M)
    RHCX(N) = PEX(N)/(RG*TCX(N))
    ACX           = SQRT(G2*RG*TCX(N))
    VTCX(N) = AM5*ACX


*———— WRITE TO BODY DATA FILE IF FIRST STREAMTUBE *
            IF(N.EQ.1) THEN
*              TR  =1. +  .5*GM*AM4**2
*              T4  = T04/TR
*              PR  = TR**(G/GM)
*              P4  = P04/PR
*              RH4 = P4/(RG2*T4)
*                 XX = X(ILF) + XCC
*              WRITE(55,*) XX,AM4,P4,RH4
*   ENDIF

*———— COMBUSTOR EXPANSION
*———— WRITE TO BODY SURFACE DATA FILE IF FIRST STREAMTUBE *
    FOR THE FIRST STREAM TUBE, EXPAND IN INCREMENTS
 AREA5 = AREAST
    IF(N.EQ.1) THEN
            AREA = AREA4
            AMA  = AM4
            DO 170 I = 1,10
                 AREAL = AREA
                 AML   = AMA
 ARA  = (2.*(1. + .5*G2M*AML**2)/G2P)**(.5*G2P/G2M)/AML FI  = .1*I
   XX=X(ILF)+XCC+FI*(X(ILF+1)-X(ILF)-XCC) AREA=AREA4+FI*(AREA5-AREA4)
                 ARR  = ARA*AREA/AREAL
                 CALL MSUP(ARR,G2,AMA)
                  TRA  = 1. + .5*G2M*AMA**2
                   TST = T04/TRA
                   PR  =  TRA**(G2/G2M)
                   PST = P04/PR
                   RHST = PST/(RG2*TST)
*                   WRITE(55,*) XX,AMA,PST,RHST
```

```
170     CONTINUE
        ENDIF

   AREA4 = AREA3
   AR4        = (2.*(1. + .5*G2M*AM4**2)/G2P)**(.5*G2P/G2M)/AM4
   AREA5 = AREAST
   AR5        = AR4*AREA5/AREA4
   CALL MSUP(AR5,G2,AM5)

        FF(N)  = F
        AMEX(N) = AM5
        TREX      = 1. + .5*G2M*AM5**2
        TEX(N)  = T04/TREX
        P0EX(N) = P04
PEX(N)  = P04/TREX**(G2/G2M) RHEX(N) = PEX(N)/(RG*TEX(N)) AEX              =
SQRT(G2*RG*TEX(N))
        VTEX(N) = AM5*AEX
        H0EX(N) = CP*T04

*________ COMPUTE MASS FLUX AT COMBUSTOR EXIT
        F1  = (1. + .5*G2M*AM5**2)**(.5*G2P/G2M)
        SQSTF = SQRT(G2/(RG*T04))
        FLX    = P04*AREA5*AM5*SQSTF/F1
*    WRITE(6,*),'  Comb. streamtube flux after expansion = ',FLX
        F1  = (1. + .5*G2M*AM4**2)**(.5*G2P/G2M)
        SQSTF = SQRT(G2/(RG*T04))
        FLX    = P04*AREA4*AM4*SQSTF/F1
*    WRITE(6,*),'  Comb. streamtube flux before expansion = ',FLX

*          WRITE(6,1) N,AM5,P04,T04,PEX(N),RHEX(N)
   1      FORMAT('  Rayleigh...',I4,5E12.4)
             CALL SpinBeachBall(1)
  200    CONTINUE
*______________    END OF STREAMTUBE LOOP ______________________________


*________ OUTPUT FROM RAYLEIGH LINE COMPUTATION
     FLX = 0.
     DO 202 N = 1,NSTR
          T0CC = TCC(N)*(1. + .5*GM*AMCC(N)**2)
          RHN = RHCC(N)
                 VTN = VTCC(N)
          DFLX = RHN*VTN*AREA4
          FLX = FLX + DFLX
          THD = 0.

             CALL SpinBeachBall(5)
  202    CONTINUE

*____       Average Flow Properties in Propulsion Duct    ____

                    AMIN2 =  0.0
                    P0IN2 =    0.0
                    PIN2 = 0.0
```

```
TIN2 = 0.0
RHIN2 =  0.0
VTIN2 =  0.0
THIN2 =  0.0

AMCC3 = 0.0
P0CC3 =  0.0
PCC3 = 0.0
TCC3 = 0.0
RHCC3 =  0.0
VTCC3 =  0.0

AMCX4 = 0.0
P0CX4 =  0.0
PCX4 = 0.0
TCX4 = 0.0
RHCX4 =  0.0
VTCX4 =  0.0
FCOMB = 0.0

AMEX5 = 0.0
P0EX5 =  0.0
PEX5 = 0.0 TEX5 = 0.0 RHEX5 =  0.0
VTEX5 =  0.0

DO 203 J = 1,NSTR

AMIN2 = AMIN2 + AMIN(J)/NSTR P0IN2 = P0IN2 + P0IN(J)/NSTR PIN2 = PIN2 + PIN(J)/NSTR
TIN2= TIN2 + TIN(J)/NSTR RHIN2 = RHIN2 + RHIN(J)/NSTR VTIN2 = VTIN2 + VTIN(J)/
NSTR THIN2 = THIN2 + THIN(J)/NSTR

AMCC3 = AMCC3 + AMCC(J)/NSTR P0CC3 = P0CC3 + P0CC(J)/NSTR PCC3 = PCC3 + PCC(J)/
NSTR TCC3 = TCC3 + TCC(J)/NSTR RHCC3 = RHCC3 + RHCC(J)/NSTR VTCC3 = VTCC3 +
VTCC(J)/NSTR

AMCX4 = AMCX4 + AMCX(J)/NSTR P0CX4 = P0CX4 + P0CX(J)/NSTR PCX4 = PCX4 +
PCX(J)/NSTR TCX4 = TCX4 + TCX(J)/NSTR RHCX4 = RHCX4 + RHCX(J)/NSTR VTCX4 =
VTCX4 + VTCX(J)/NSTR FCOMB = FCOMB + FF(J)/NSTR

AMEX5 = AMEX5 + AMEX(J)/NSTR P0EX5 = P0EX5 + P0EX(J)/NSTR PEX5 = PEX5 + PEX(J)/
NSTR TEX5 = TEX5 + TEX(J)/NSTR RHEX5 = RHEX5 + RHEX(J)/NSTR VTEX5 = VTEX5 +
VTEX(J)/NSTR

      CALL SpinBeachBall(5)
203   CONTINUE

T0IN2 = TIN2*(1+0.5*GM*AMIN2**2) T0CC3 = TCC3*(1+0.5*GM*AMCC3**2) T0CX4 =
TCX4*(1+0.5*G2M*AMCX4**2) T0EX5 = TEX5*(1+0.5*G2M*AMEX5**2)

*______________________________________________________________
*        SCRAMJET COWL FORCE COMPUTATION
*______________________________________________________________
```

```
*     Note: This temporary computation is based on oblique shock/
*   expansion theory and does not include the interaction
*   with the sidewall shocks.  This is hopefully a small
*   error since the intersections are confined to small
*   regions near the corners.  A more accurate computation
*   would require a lengthy 3D computation of the region.
*   If such a computation were performed, it might as well
*   do the entire inlet, but run times would be significantly
*   longer by at least an order of magnitude from the current
*   procedure.                    rlr
*
*________ INLET COWL FORCES BY OBLIQUE SHOCK THEORY COWLANGL =
ATAN2((YEX-YCL),(XEX-XCL))
        DL = THCL - COWLANGL
        LSHANG = .TRUE.
CALL SHANG (AMCL,DL,G,LSHANG,THY,PR,PTR,AMY) PINL = PCL*PR - P(ILF,1)
DXAV = .5*H1*(XW1 + XW2)
DAW1=.5*TAN(ABS(D1))*(H1*XW1)**2DAW2=.5*TAN(ABS(D2))*(H1*XW2)**2 AREA
= H1*DXAV - DAW1 - DAW2 XAVG = XCL + .5*H1*XCC
YAVG  =  YCL  +  (XAVG-XCL)*(YEX-YCL)/(XEX-XCL)   FCLXI  =  -
PINL*AREA*SIN(COWLANGL)
FCLYI = -PINL*AREA*COS(COWLANGL) DXFYCL = -XAVG*FCLYI
        DYFXCL = -YAVG*FCLXI

*  WRITE(6,*),' Computing forces on cowl inside inlet...'
*  WRITE(6,*),'   COWLANGL = ',180.*COWLANGL/PI
*  WRITE(6,*),'      PINL = ',PINL,'  AREA = ',AREA
*  WRITE(6,*),'     FCLXI = ',FCLXI,'  FCLYI = ',FCLYI
*  WRITE(6,*),'      XAVG = ',XAVG,'  YAVG = ',YAVG
*  WRITE(6,*),'    DXFYCL = ',DXFYCL,'  DYFXCL = ',DYFXCL

*________ VERTICAL FORCE IN THE COMBUSTION ZONE
*     Note:  Assume, for now that pressure contribution to overall
*            force balance is small.
        IZZZ = 1
        IF(IZZZ.EQ.1) GO TO 210
        PAVG = .5*(PCC(NSTR) + PEX(NSTR))
        XW1ST = EL - H1*XW1
        XW2ST = EL - H1*XW2
DXAV  = .5*(XW1ST + XW2ST)
DAW1=.5*TAN(ABS(D1ST))*XW1ST**2DAW2=.5*TAN(ABS(D2ST))*XW2ST**2 AREA
= H1*DXAV - DAW1 - DAW2 FCLXC = -PAVG*AREA*SIN(COWLANGL) FCLYC = -
PAVG*AREA*COS(COWLANGL)
XAVG  = XCL + XCC + (2.*PEX(NSTR)+PCC(NSTR))/(3.*PAVG) YAVG  = YCL + (XAVG-
XCL)*(YEX-YCL)/(XEX-XCL)
        DXFYCC = -XAVG*FCLYC
        DYFXCC = -YAVG*FCLXC

*  WRITE(6,*),' Computing forces on cowl in combustion zone...'
*  WRITE(6,*),'       PAVG = ',PAVG,'  AREA = ',AREA
*  WRITE(6,*),'      FCLXC = ',FCLXC,'  FCLYC = ',FCLYC
*  WRITE(6,*),'       XAVG = ',XAVG,'  YAVG = ',YAVG
*  WRITE(6,*),'     DXFYCC = ',DXFYCC,'  DYFXCC = ',DYFXCC
```

```
*________ VERTICAL FORCE ON THE OUTSIDE BY EXPANSION THEORY
*        Note: Computed below prior to computing aft body stuff
  210     CONTINUE

*_____ SHIFT VALUES FROM CENTER OF STREAMTUBE TO EDGES
*   Note: can now use arrays at combustor inlet temporarily
          DO 220 N = 1,NSTR
           RHCC(N) = RHEX(N)
           AMCC(N) = AMEX(N)
           P0CC(N) = P0EX(N)
           VTCC(N) = VTEX(N)
           PCC(N)  = PEX(N)
           TCC(N)  = TEX(N)
           H0CC(N) = H0EX(N)
  220     CONTINUE

RHEX(1)=.125*(9.*RHCC(1)-RHCC(2)) AMEX(1)=.125*(9.*AMCC(1)-AMCC(2)) P0EX(1)
= .125*(9.*P0CC(1) - P0CC(2)) VTEX(1) = .125*(9.*VTCC(1) - VTCC(2)) H0EX(1) =
.125*(9.*H0CC(1) - H0CC(2)) PEX(1) =.125*(9.*PCC(1) - PCC(2)) TEX(1) =.125*(9.*TCC(1)
- TCC(2))

RHEX(NSTR+1) = .125*(9.*RHCC(NSTR) - RHCC(NSTR-1)) AMEX(NSTR+1) =
.125*(9.*AMCC(NSTR) - AMCC(NSTR-1)) P0EX(NSTR+1) = .125*(9.*P0CC(NSTR) -
P0CC(NSTR-1)) VTEX(NSTR+1)=.125*(9.*VTCC(NSTR)-VTCC(NSTR-1)) H0EX(NSTR+1)
=.125*(9.*H0CC(NSTR)-H0CC(NSTR-1)) PEX(NSTR+1) =.125*(9.*PCC(NSTR) -PCC(NSTR-
1)) TEX(NSTR+1) = .125*(9.*TCC(NSTR) - TCC(NSTR-1))

    DO 230 N = 2,NSTR
          RHEX(N) = .5*(RHCC(N) + RHCC(N-1))
AMEX(N) = .5*(AMCC(N) + AMCC(N-1)) P0EX(N) = .5*(P0CC(N) + P0CC(N-1)) VTEX(N) =
.5*(VTCC(N) + VTCC(N-1)) H0EX(N) = .5*(H0CC(N) + H0CC(N-1)) PEX(N) = .5*(PCC(N) +
PCC(N-1)) TEX(N)  = .5*(TCC(N)  + TCC(N-1))
   230 CONTINUE

*__________ See if we messed up the mass flux FLX   = 0.
    DO 240 J = 2,NSTR+1
VAV  = .5*(VTEX(J) + VTEX(J-1))
RVAV  = .5*(RHEX(J)*VTEX(J) + RHEX(J-1)*VTEX(J-1)) DFLX  = RVAV*AREA5
          FLX  = FLX + DFLX
   240    CONTINUE

*     WRITE(6,*),'Net Mass Flux at combexh. after shift = ',FLX

*___________________________________________________________
*                VEHICLE AFTBODY COMPUTATION
*___________________________________________________________
*     WRITE(6,*),'    Vehicle Aft body computation...'
*________ STARTING COORDS
     I = ILF + 1
     DY = (YEX - YA(1))/(JM - 3.)
     DO 300 J = 1,JM
          X(I)  = XA(1)
          Y(I,J) = YA(1) + (J - 1.)*DY
   300    CONTINUE
```

```
*________ STARTING VALUES - Linear interpolation from combustor exhaust DO 320 J = 1,JM-
2
        DO 310 N = 2,NSTR
NZ = N - 1 IF(Y(I,J).LT.YCX(N)) GO TO 315
  310       CONTINUE

  315       YR = (Y(I,J) - YCX(NZ))/(YCX(NZ+1) - YCX(NZ))
RHO(I,J) = RHEX(NZ) + YR*(RHEX(NZ+1) - RHEX(NZ)) VT(I,J) = VTEX(NZ) +
YR*(VTEX(NZ+1) - VTEX(NZ)) AM(I,J) = AMEX(NZ) + YR*(AMEX(NZ+1) - AMEX(NZ))
H0(I,J) =H0EX(NZ)+YR*(H0EX(NZ+1)-H0EX(NZ)) P0(I,J) =P0EX(NZ)+YR*(P0EX(NZ+1)
- P0EX(NZ)) T(I,J)   = TEX(NZ) + YR*(TEX(NZ+1) - TEX(NZ))
        P(I,J) = PEX(NZ) + YR*(PEX(NZ+1) - PEX(NZ))
        TH(I,J) = ALPHA
  320   CONTINUE

*__________ EXTERNAL CONDITIONS
ANGL = THCL - COWLANGL IF(ANGL.LT.0.) THEN
WRITE(6,*),'  Shock in Inlet!'
*        WRITE(6,*),'   THCL = ',180.*THCL/PI
              GOTO 3334
  ENDIF
  CALL PM(AMCL,ANGL,G,AMCLY,TH1,TH2,THAV,PCLY)
*  WRITE(6,*),'  Cowl Lip expansion...'
*  WRITE(6,*),'      AM = ',AMCL
*  WRITE(6,*),' Turn Angle = ',180.*ANGL/PI,' deg'
*  WRITE(6,*),'     AM2 = ',AMCLY
  AMINF = AMCLY


*_________ COMPUTE SHOCK PROPERTIES AT EXHAUST LIP
  PCLY  = PCL*PCLY
  P2    =   .6*P(I,JM-2) + .4*PCLY
*  WRITE(6,*),' Iteration loop for exhaust lip conditions...'
*  WRITE(6,*),'    Mext = ',AMCLY
*  WRITE(6,*),'    Pext = ',PCLY
*  WRITE(6,*),'    Mint = ',AM(I,JM-2)
*  WRITE(6,*),'    Pint = ',P(I,JM-2) .

*__________ ITERATE
  325    CONTINUE
   AA    = SQRT(.5*(GP*P2/PCLY + GM)/(G*AMCLY**2))
WS(I) = ATAN2(AA,SQRT(1. - AA**2))
   TW    = TAN(WS(I))
   SW    = SIN(WS(I))
   F1    = TW*(.5*GP*AMCLY**2/((AMCLY*SW)**2 - 1.) - 1.)
   THL   = ATAN2(1.,F1)

*__________ CALL SHANG TO GET P2S
   LSHANG = .TRUE.
CALL SHANG(AMCLY,THL,G,LSHANG,WSHK,PRAT,PTRAT,AM2S) P2S = PRAT*PCLY

*__________ CALL PM TO GET P2E
   AME = AM(I,JM-2)
CALL PM(AME,THL,G,AM2E,TH1,TH2,THA,PRAT) P2E = PRAT*P(I,JM-2)
```

```
*___________ GET NEW P2
    P2L = P2
    P2 = .5*(P2S + P2E)
    DP2 = ABS(P2 - P2L)
*   WRITE(6,*),'  DP2 = ',DP2
    IF(DP2.GT..001) GO TO 325
*_______ CONVERGED: GET CONDITIONS OUTSIDE SLIP LINE TR =    1.          +
.5*GM*AM2S**2
    T2S = T0INF/TR
    RHO2S = P2/(RG*T2S)
    JSS = JM - 1
    DO 330 J = JSS,JM
          RHO(I,J) = RHO2S
          AM(I,J)  = AM2S
VT(I,J)  = AM2S*SQRT(G*RG*T2S) H0(I,J)  = H0INF
TH(I,J)  = THL + ALPHA
          T(I,J)   = T2S
          P(I,J)   = P2
          Y(I,J)   = Y(I,JM-2)
          P0(I,J)  = P(I,J)*TR**(G/GM)
  330     CONTINUE
*_______ SET VALUES ON LOWER PART OF SLIP LINE JSL     = JM - 2
    AM(I,JSL) = AM2E
    P(I,JSL)  = P2
TR     = 1. + .5*GM*AM2E**2 T(I,JSL)  = H0(I,JSL)/(CP*TR) RHO(I,JSL) = P2/(RG*T(I,JSL))
VT(I,JSL)  = AM2E*SQRT(G*RG*T(I,JSL)) TH(I,JSL)  = THL + ALPHA
    P0(I,JSL)  = P2*TR**(G/GM)

*   WRITE(6,*),'  Converged.  Exhaust Lip Values:'
*   WRITE(6,*),' Shock wave angle = ',180.*WS(I)/PI
*   WRITE(6,*),' Slip Line angle  = ',180.*TH(I,JSL)/PI
*   WRITE(6,*),' Mach No above SL = ',AM(I,JSS)
*   WRITE(6,*),' Mach No below SL = ',AM(I,JSL)
*   WRITE(6,*),' Pressure above SL= ',P2S
*   WRITE(6,*),' Pressure below SL= ',P2E

*_______ SET INITIAL WALL VALUES
    DY = CAY(1)
    DX = CAX(1)
    TH(I,1) = ATAN2(DY,DX)
    AMA1 = AM(I,1)
    THA1 = ABS(TH(I,1))
*   WRITE(6,*),'  Calling PM...AMA1 = ',AMA1,'  TH = ',180.*THA1/PI
    CALL PM(AMA1,THA1,G,AM2,TH1,TH2,THAV,PRAT)
    AM(I,1) = AM2
    P(I,1)  = PRAT*P(I,1)
*   WRITE(6,*),' PRAT = ',PRAT
*   WRITE(6,*),' G    = ',G
*   WRITE(6,*),' GM   = ',GM
RHO(I,1)  =  RHO(I,1)*PRAT**(1./G)  T(I,1)     =  T(I,1)*PRAT**(GM/G)  CSP        =
SQRT(G*RG*T(I,1)) VT(I,1)  = CSP*AM(I,1)

*_______ SET FLOW ANGLES
```

```
      DY = CAY(1)
      DX = CAX(1)
*     TH(I,1) = ATAN2(DY,DX)
*     DO 340 J = 2,JM-2
*       YR = (Y(I,J) - Y(I,1))/(Y(I,JM-1) - Y(I,1))
*       TH(I,J) = TH(I,1) + YR*(THL - TH(I,1))
*  340         CONTINUE

*——————————— FORCE ON OUTER COWL
*     WRITE(6,*),'  Computing forces, etc. on outer cowl...'
      XAVG = .5*(XCL + XEX)
      YAVG = .5*(YCL + YEX)
      AREA = H1*(XEX - XCL)
      FCLXE = PCLY*AREA*SIN(COWLANGL)
      FCLYE = PCLY*AREA*COS(COWLANGL)
      DXFYE = -FCLYE*XAVG
      DYFXE = -FCLXE*YAVG
*     WRITE(6,*),'     COWLANGL = ',180.*COWLANGL/PI
*     WRITE(6,*),'    PCLY = ',PCLY,'  AREA = ',AREA
*     WRITE(6,*),'   FCLXE = ',FCLXE,'  FCLYE = ',FCLYE
*     WRITE(6,*),'    XAVG = ',XAVG,'  YAVG = ',YAVG
*     WRITE(6,*),'   DXFYE = ',DXFYE,'  DYFXE = ',DYFXE

*——————————— TOTAL FORCE & MOMENT ON COWL
     FCLX = FCLXI + FCLXE + FCLXC
     FCLY = FCLYI + FCLYE + FCLYC
DXFYC = DXFYCL + DXFYCC + DXFYE DYFXC = DYFXCL + DYFXCC + DYFXE


*——————————— MARCH THROUGH THE AFTBODY REGION ————————
*       WRITE(6,*),'  Calling MOCA...'
          CALL MOCA(Kicker)
*       WRITE(6,*),'  Back from MOCA...'
               IF (Kicker.GT.0) THEN
WRITE(6,*) ' Nozzle Errors Prevented Completion of Solution' WRITE(6,*) '       ——— GOING
TO NEXT CASE ———'
                      GOTO 3334
               END IF
*          CLOSE(55,STATUS= 'KEEP')

*————————————————————————————————————————————————————————*
           FINAL SUMMARY COMPUTATIONS
*————————————————————————————————————————————————————————*———— Ef-
fective Equivalence Ratio:

               EREFF = ER
               ERCOMB=FCOMB/FCRIT
               IF (ERCOMB.GT.ER) THEN
                      EREFF = ERCOMB
               END IF
*————— Freestream Momentum Flux
        FMMINF = FLXMSS*VINF
*————— Average Properties at Nozzle Exit: NSTR6 = JSL-2
                      AMN6 = 0.0
```

```
                    PN6 = 0.0
                    RHN6 = 0.0
                    VTN6 = 0.0

        DO 439 J = 2,JSL

AMN6 = AMN6 + .5*(AM(ILA,J)+AM(ILA,J-1))/NSTR6 PN6 = PN6 + .5*(P(ILA,J)+P(ILA,J-
1))/NSTR6
RHN6 = RHN6 + .5*(RHO(ILA,J)+RHO(ILA,J-1))/NSTR6 VTN6 = VTN6 +
.5*(VT(ILA,J)*COS(TH(ILA,J)) +
   1               VT(ILA,J-1)*COS(TH(ILA,J-1)))/NSTR6
  439 CONTINUE
                    TN6 = PN6/(RG2*RHN6)
P0N6 = PN6*(1 +.5*G2M*AMN6**2)**(G2/G2M) T0N6 = TN6*(1 +.5*G2M*AMN6**2)

*——— Outflow MASS Flux
      FLX  = 0.

      DO 440 J = 2,JSL
AREA = H1*(Y(ILA,J) - Y(ILA,J-1))
   RHAV = .5*(RHO(ILA,J)+RHO(ILA,J-1)) VNAV = .5*(VT(ILA,J)*COS(TH(ILA,J)) +
   1               VT(ILA,J-1)*COS(TH(ILA,J-1)))
                    DMASS = RHAV*AREA*VNAV
         FLX   = FLX + DMASS

  440 CONTINUE
               Rat = FLXMSS*(1+EREFF*FCRIT)/FLX

*——— Outflow Momentum Flux

      FXMMA = 0.
      FYMMA = 0.
      MMA = 0.
      FLX  = 0.
           PFORCEX = 0.


            DO 441 J = 2,JSL
AREA = H1*(Y(ILA,J) - Y(ILA,J-1))
   RHAV = .5*(RHO(ILA,J)+RHO(ILA,J-1)) VXAV = .5*(VT(ILA,J)*COS(TH(ILA,J)) +
   1               VT(ILA,J-1)*COS(TH(ILA,J-1)))

            VYAV = .5*(VT(ILA,J)*SIN(TH(ILA,J))+ 1                    VT(ILA,J-
1)*SIN(TH(ILA,J-1)))
             PAV = .5*(P(ILA,J)+P(ILA,J-1))
             DMASS = Rat*RHAV*AREA*VNAV
             DFXMM = DMASS*VXAV
             DFYMM = DMASS*VYAV
             DM = XA(NPA)*DFYMM-(Y(ILA,J)+Y(ILA,J-1))*(DFXMM+AREA*PAV)/2.0

            FLX = FLX + DMASS
            FXMMA = FXMMA + DFXMM
            FYMMA = FYMMA + DFYMM
PFORCEX = PFORCEX + AREA*PAV
```

```
          MMA = MMA - DM
    441 CONTINUE
*             WRITE(6,*) ' '
*             WRITE(6,*),'Freestream capture area  = ',CAPTUREAREA
*         WRITE(6,*),' Pfinlet  =', PFORCEI
*         WRITE(6,*),' Pfexit  =', PFORCEX
*         WRITE(6,*),' Net mass flux out = ',FLX
*             WRITE(6,*),'Momentum Fluxes:'
*         WRITE(6,*),' Freestream Momentum Flux = ',FMMINF
*         WRITE(6,*),' Inflow x-momentum flux   = ',FLXMTMX
*         WRITE(6,*),' Inflow y-momentum flux   = ',FLXMTMY
*         WRITE(6,*),' Outflow x-momentum flux  = ',FXMMA
*         WRITE(6,*),' Outflow y-momentum flux  = ',FYMMA

*   Force on Vehicle (FX Positive Forward; FY Positive Upward):

        FX = (FXMMA + FLXMTMX) + PFORCEX - PFORCEI
        FY = (FYMMA + FLXMTMY)
            FMAG = SQRT(FX*FX+FY*FY)
            TANG = (180./PI)*ATAN2(FY,FX)

*       Moment on Vehicle (Positive Pitch up):

            PMOMENT = MMA + FLXMTMA
        XB = -PMOMENT/FY
XB = SIN(PI*TANG/180.)*XB/SIN(PI*(TANG-ALPHAD)/180.) XB = XB/XA(NPA)

*   Thrust and Moment Coefficients:

            CAPTUREAREA = FLXMSS/(RHOINF*VINF)
AIAC = AREAINLET/CAPTUREAREA CT = FMAG/(AREAINLET*DYNAMP)
CM = PMOMENT/(AREAINLET*DYNAMP*XA(NPA))

SPF = FX/(32.17*EREFF*FCRIT*FLXMSS) IF (SPF.LT.0.) THEN
SPF = 0.0
        END IF

*   Write results to terminal

WRITE(6,*) 'Vehicle Geometry:' WRITE(6,*) '—————————' WRITE(6,958) XA(NPA)
WRITE(6,964) AREAINLET WRITE(6,963) AIAC

WRITE(6,*) ' '
WRITE(6,*) 'System Performance:' WRITE(6,*) '—————————'
   WRITE(6,957) CT,CM
WRITE(6,959) SPF
WRITE(6,962) EREFF
WRITE(6,956) FLXMSS
   WRITE(6,974) FLXMSS*ER*FCRIT WRITE(6,952) FX,FY
WRITE(6,953) FMAG
WRITE(6,971) TANG
```

```
WRITE(6,961) PMOMENT WRITE(6,960) XB
WRITE(6,*) ' '
         WRITE(6,*) 'Average Flow Properties in Duct:'
         WRITE(6,*) '________________________________'
         WRITE(6,975)
W        R        I        T        E        (        6        ,        *        )
'_______________________________________________________________________
' WRITE(6,979) FLTMACH,VINF,TINF,T0INF,PINF,P0INF
         WRITE(6,976) AMIN2,VTIN2,TIN2,T0IN2,PIN2,P0IN2
         WRITE(6,977) AMCC3,VTCC3,TCC3,T0CC3,PCC3,P0CC3
         WRITE(6,978) AMCX4,VTCX4,TCX4,T0CX4,PCX4,P0CX4
         WRITE(6,980) AMEX5,VTEX5,TEX5,T0EX5,PEX5,P0EX5
         WRITE(6,981) AMN6,VTN6,TN6,T0N6,PN6,P0N6
W        R        I        T        E        (        6        ,        *        )
'_______________________________________________________________________
' WRITE(6,*) ' '

  952 FORMAT ('',' Fx =',F9.1,' lbf',3X,' Fy =',F9.1,' lbf')
  953 FORMAT ('',' Thrust =',F9.1,' lbf')
  956 FORMAT ('',' Air Mass Flow Rate =',F7.2,' slug/sec',3X)
  957 FORMAT ('',' Ct =',F5.3,5X,' Cm =',F5.3)
  958 FORMAT ('',' Total Length = ',F6.2,' ft')
  959 FORMAT ('',' ISPfuel =',F8.2,' sec')
  960 FORMAT ('',' Thrust line-of-action =',F6.3)
  961 FORMAT ('',' Moment about Origin =',I10,' ft-lbf')
962 FORMAT ('',' Effective Equivalence Ratio =',F7.3)
964 FORMAT ('',' Module Inlet Area =',F6.2,' ft^2') 963 FORMAT ('',' Ai/Acapture =',F5.3)
  971 FORMAT ('',' Thrust Angle =',F7.2,'°')
  972 FORMAT ('',' Ai/Acapture =',F5.3)
  974 FORMAT ('',' Fuel Mass Flow Rate =',F7.2,' slug/sec')
975 FORMAT ('',19X,'   M',3X,'V [ft/sec]',3X,'T [°R]',3X,'To [°R]',3X,' p [psf]',3X,'Po [psf]'
)
  976 FORMAT ('',' Duct Inlet Plane:',3X,F5.2,3X,F7.0,3X,F7.1,3X,F7.0,3X,F7.1,3X,F11.0)
977 FORMAT ('',' Combustor Inlet :',3X,F5.2,3X,F7.0,3X,F7.1,3X,F7.0,3X,F7.1,3X,F11.0)
978 FORMAT ('',' Combustor Exit  :',3X,F5.2,3X,F7.0,3X,F7.1,3X,F7.0,3X,F7.1,3X,F11.0) 979
FORMAT ('',' Free Stream     :',3X,F5.2,3X,F7.0,3X,F7.1,3X,F7.0,3X,F7.1,3X,F11.0) 980 FOR-
MAT ('',' Nozzle Entrance :',3X,F5.2,3X,F7.0,3X,F7.1,3X,F7.0,3X,F7.1,3X,F11.0) 981 FOR-
MAT ('',' Nozzle Exit     :',3X,F5.2,3X,F7.0,3X,F7.1,3X,F7.0,3X,F7.1,3X,F11.0)

*  Write Data to File:
         IF (NOPE.GT.0) GOTO 3334

         PUTOUT(1) = FLTMACH
         PUTOUT(2) = ALPHAD
PUTOUT(3) = EREFF PUTOUT(4) = FLXMSS PUTOUT(5) = FCOMB/FCRIT PUTOUT(6) =
FMAG PUTOUT(7) = TANG PUTOUT(8) = SPF PUTOUT(9) = PMOMENT PUTOUT(10) = XB
PUTOUT(11) = FX PUTOUT(12) = FY PUTOUT(13) = CT PUTOUT(14) = CM

       WRITE(12,'(9999(F,A1))') (PUTOUT(I),CHAR(9),I=1,13), 1
             PUTOUT(14)
*______________________________________________________
         CALL SpinBeachBall(1)
```

```
 3334  CONTINUE

       RETURN
          CALL RelBeachBall
       END
*
*
*
       SUBROUTINE MOCF *--------------------------------------------------
------
*
*      This subroutine computes the flow over the forebody
*      of a super/hypersonic vehicle using the Method of
*      characteristics.
*
*-------------------------------------------------------------PARAMETER
(IM=501,JM=15)

    COMMON/CBSPF/ AFX(IM),BFX(IM),CFX(IM),AFY(IM),BFY(IM),CFY(IM), 1
    AFC(IM),BFC(IM),CFC(IM),NPF
COMMON/DEPVR/ RHO(IM,JM),VT(IM,JM),T(IM,JM),P(IM,JM),
    1          TH(IM,JM),AM(IM,JM),WS(IM),H0(IM,JM),P0(IM,JM)
       COMMON/GASPR/ G,RG,GM,GP,CP
       COMMON/GRIDS/ X(IM),Y(IM,JM)
       COMMON/GRIDF/ XF(IM),YF(IM),CF(IM),ILF
       COMMON/INFTY/ AMINF,PINF,TINF,ALPHA

       LOGICAL     AXISYMA
       CALL InitBeachBall(512)
       CALL ShowBeachBall

*-------------------------------------------------------
*           LEADING EDGE CONE STUFF
*-------------------------------------------------------
       PI = 3.141592654
       CAL = COS(ALPHA)
T0INF = TINF*(1. + .5*GM*AMINF**2) H0INF = CP*T0INF
AINF  = SQRT(G*RG*TINF)
       VINF  = AMINF*AINF
RHOINF = PINF/(RG*TINF)
*---------- Find leading edge cone angle
*   User may specify XLC, ie. how far from nose, default is 5%
       XLC = .1
       XS  = XF(1) + XLC*(XF(NPF) - XF(1))

       CALL INTRPC(XF,XS,AFX,BFX,CFX,NPF,IZ,TT)

YS  = YF(IZ) + ((AFY(IZ)*TT + BFY(IZ))*TT + CFY(IZ))*TT DX  = (3.*AFX(IZ)*TT +
2.*BFX(IZ))*TT + CFX(IZ)
          DY  = (3.*AFY(IZ)*TT + 2.*BFY(IZ))*TT + CFY(IZ) *----------- DL IS THE
CONE ANGLE
          DL  = ATAN(ABS(DY/DX))

*---------- CALL CONEFLOW TO GET STARTING CONDITIONS --------*
```

```
WRITE(6,*),'    Calling coneflow...DL = ',180.*DL/PI
         CALL CONEFLOW(DL)
*    WRITE(6,*),'    Back from Coneflow...'

*----------------------------------------------------------------------*
      FIRST X-STATION COORDINATES
*----------------------------------------------------------------------DXS = XS - XF(1)
         X(1) = XS
         Y(1,JM) = YF(1) + DXS*TAN(WS(1))
         DY = (Y(1,JM) - YS)/(JM - 1.)
         DO 80 J = 1,JM
          Y(1,J) = YS + DY*(J - 1.)
CALL SpinBeachBall(1)

   80    CONTINUE

*----------------------------------------------------------------------
*                  MAIN LOOP
*----------------------------------------------------------------------
*        OPEN(2,FILE='mocf.dat',STATUS='UNKNOWN')
*        WRITE(2,*) X(1),Y(1,1),Y(1,JM)
*        OPEN(3,FILE='mocf.out',STATUS='UNKNOWN')
*        WRITE(3,4)
*   4  FORMAT(3X,'i',8X,'x',11X,'y',11X,'M',11X,'w',11X,'p',11X,'Vt')

         I = 1
  100    I = I + 1

*------------ DETERMINE THE STEP SIZE, DX ------------
         DXF = .5

*---------- LIMITATION AT GROUND PLANE
         DY = (Y(I-1,JM) - Y(I-1,1))/(JM-1.)
         ARG = AM(I-1,2)**2 - 1.
         IF(ARG.LT.0.) THEN
    WRITE(6,*),'    ARG<0 at loc. G1...'
    WRITE(6,*),'AM(I-1,2) = ',AM(I-1,2)
    WRITE(6,*),'I = ',I,'    X(I) = ',X(I)
          GOTO 3333
         ENDIF
EM = ATAN2(1.,SQRT(AM(I-1,2)**2 - 1.))
DXW = DXF*DY/(TAN(EM + TH(I-1,2)) - TAN(TH(I-1,2)))

*---------- LIMITATION AT SHOCK
         ARG = AM(I-1,JM)**2 - 1.
         IF(ARG.LT.0.) THEN
WRITE(6,*),'    ARG<0 at loc. G2...' WRITE(6,*),'    AM(I-1,JM) = ',AM(I-1,JM) WRITE(6,*),'
I = ',I,'    X(I) = ',X(I) GOTO 3333
         ENDIF
EM = ATAN2(1.,SQRT(AM(I-1,JM)**2 - 1.))
         DXS = DXF*DY/(TAN(EM - TH(I-1,JM)) + TAN(WS(I-1)))

         DX = DXS
         IF(DXW.LT.DXS) DX = DXW
```

```
*———— SET Y-STATIONS AT THIS I ————
XX  = X(I-1) + DX IF(XX.GE.XF(NPF)) THEN
          XX = XF(NPF)
DX = XF(NPF) - X(I-1)
          ILF = I
          Y(I,1) = YF(NPF)
          CFW = CF(NPF)
   ELSE
          CALL INTRPC(XF,XX,AFX,BFX,CFX,NPF,IZ,TT)
Y(I,1)  = YF(IZ) + ((AFY(IZ)*TT + BFY(IZ))*TT + CFY(IZ))*TT CFW      = CF(IZ) +
((AFC(IZ)*TT + BFC(IZ))*TT + CFC(IZ))*TT
   ENDIF
   X(I)   = XX
   Y(I,JM) = Y(I-1,JM) + DX*TAN(WS(I-1))

   DY = (Y(I,JM) - Y(I,1))/(JM - 1.)
   DO 120 J = 2,JM
          X(I)  = X(I)
          Y(I,J) = Y(I,J-1) + DY
             CALL SpinBeachBall(1)

   120    CONTINUE
*————————————————————————————————————* CONDI-
TIONS AT THE WALL
*————————————————————————————————————*———— SURFACE
SLOPE
DYS=(3.*AFY(IZ)*TT+2.*BFY(IZ))*TT+CFY(IZ) DXS=(3.*AFX(IZ)*TT+2.*BFX(IZ))*TT
+ CFX(IZ) TH(I,1) = ATAN2(DYS,DXS)

*———— INITIAL GUESS,VALUES AT LAST WALL STATION P(I,1)  = P(I-1,1)
   AM(I,1)  = AM(I-1,1)
   RHO(I,1) = RHO(I-1,1)
   VT(I,1)  = VT(I-1,1)

   AM1    = AM(I-1,1)
   TH1    = TH(I-1,1)

*———— ITERATION LOOP FOR THE WALL ————*           WRITE(6,*),'
Iteration loop for wall...'
*———— LOCATE YC1
   130    CONTINUE
   PWL = P(I,1)
   ARG = AM(I-1,1)**2 - 1.
   IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0 at loc. G3...' WRITE(6,*),'   AM(I-1,1) = ',AM(I-1,1) WRITE(6,*),'   I
= ',I,'  X(I) = ',X(I) GOTO 3333
   ENDIF
EM  = ATAN2(1.,SQRT(AM(I,1)**2 - 1.))
   ARG = AM1**2 - 1.
   IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0 at loc. G4...' WRITE(6,*),'   AM1 = ',AM1
WRITE(6,*),'   I = ',I,'  X(I) = ',X(I) GOTO 3333
   ENDIF
```

```
EMU = .5*(EM + ATAN2(1.,(SQRT(AM1**2 - 1.)))) PHI = EMU - .5*(TH(I,1) + TH1)
    YC1 = Y(I,1) + DX*TAN(PHI)
DC1 = SQRT(DX**2 + (Y(I,1) - YC1)**2)

    DO 140 J = 2,JM
          J1 = J - 1
             CALL SpinBeachBall(1)

          IF(YC1.LT.Y(I-1,J)) GO TO 145
   140 CONTINUE

*————————— COMPUTE PRESSURE & OTHER STUFF AT (X(I-1),YC1) *            4  pt
Lagrange interpolation
   145 CONTINUE
       JZ = J1
       YZ = YC1

       CALL LAGRANGE(I,JZ,YZ,YF1,YF2,YF3,YF4,JA,JB,JC,JD)

TH1 =  TH(I-1,JA)*YF1 + TH(I-1,JB)*YF2 +
    1              TH(I-1,JC)*YF3 + TH(I-1,JD)*YF4
AM1 =  AM(I-1,JA)*YF1 + AM(I-1,JB)*YF2 +
    1              AM(I-1,JC)*YF3 + AM(I-1,JD)*YF4
VT1 =  VT(I-1,JA)*YF1 + VT(I-1,JB)*YF2 +
    1              VT(I-1,JC)*YF3 + VT(I-1,JD)*YF4
RH1 = RHO(I-1,JA)*YF1 + RHO(I-1,JB)*YF2 +
    1              RHO(I-1,JC)*YF3 + RHO(I-1,JD)*YF4
P1  =  P(I-1,JA)*YF1 + P(I-1,JB)*YF2 +
    1              P(I-1,JC)*YF3 + P(I-1,JD)*YF4

       ARG = AM1**2 - 1.
       IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0 at loc. G5...' WRITE(6,*),'      AM1 = ',AM1
          WRITE(6,*),'      I = ',I,'  X(I) = ',X(I)
          GOTO 3333
       ENDIF
       BT1 = SQRT(AM1**2 - 1.)
BRV1 = BT1/(RH1*VT1**2)

*———— AVERAGES BETWEEN YC1 & Y(I,1) ARG = AM(I,1)**2 - 1. IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0 at loc. G6...' WRITE(6,*),'      AM(I,1) = ',AM(I,1)
          WRITE(6,*),'      I = ',I,'  X(I) = ',X(I)
          GOTO 3333
       ENDIF
BRV  = .5*(BRV1 + SQRT(AM(I,1)**2-1.)/(RHO(I,1)*VT(I,1)**2)) DY1  = YC1 - Y(I,1)
       CF1  = CFW/(1. + CFW*DY1*CAL)
       RHS1 = CF1*SIN(TH1)/AM1
RHS  = -.5*DC1*(RHS1 + CFW*SIN(TH(I,1))/AM(I,1))

*———— GET PRESSURE
       P(I,1) = P1 + (RHS + (TH(I,1) - TH1))/BRV

*———— GET DENSITY FROM SOUND SPEED EQN
AA     = .5*G*(P(I,1)/RHO(I,1) + P(I-1,1)/RHO(I-1,1))
```

```
        RHO(I,1) = RHO(I-1,1) + (P(I,1) - P(I-1,1))/AA

*--------- GET VT FROM STAGNATION ENTHALPY
        ARG = H0INF - G*P(I,1)/(GM*RHO(I,1))
        IF(ARG.LT.0.) THEN
           WRITE(6,*),'  ARG<0 at loc. G7...'
           WRITE(6,*),'     H0INF = ',H0INF,'  P(I,1) = ',P(I,1)
           WRITE(6,*),'     RHO(I,1) = ',RHO(I,1)
           WRITE(6,*),'     I = ',I,'  X(I) = ',X(I)
           GOTO 3333
        ENDIF
        VT(I,1) = SQRT(2.*(H0INF - G*P(I,1)/(GM*RHO(I,1))))

*--------- MACH NUMBER
AM(I,1) = VT(I,1)/SQRT(G*P(I,1)/RHO(I,1)) TR      = 1. + .5*GM*AM(I,1)**2
   P0(I,1) = P(I,1)*TR**(G/GM)

*-------------- CHECK TO SEE IF CONVERGED
   DP = ABS(PWL - P(I,1))
*  WRITE(6,*),'    DP = ',DP
   IF(DP.GT..001) GO TO 130

*----------------------------------------------------------------------*
             DO INTERIOR POINTS
*----------------------------------------------------------------------*
WRITE(6,*),' Interior points....'
   DO 260 J = 2,JM-1
              CALL SpinBeachBall(1)

         DYJ = Y(I,J) - Y(I,1)
         CFJ = CFW/(1.+CFW*DYJ*CAL)

*------------- INITIAL GUESS FOR THIS J
         P(I,J)   = P(I-1,J)
         AM(I,J)  = AM(I-1,J)
         RHO(I,J) = RHO(I-1,J)
         VT(I,J)  = VT(I-1,J)
         TH(I,J)  = TH(I-1,J)

         AM1 = AM(I,J)
         TH1 = TH(I,J)
         AM2 = AM(I,J)
         TH2 = TH(I,J)
         AMS = AM(I,J)
         THS = TH(I,J)

*----------------- ITERATION LOOP AT THIS POINT -----------210          CON-
TINUE
         PL = P(I,J)

*-------- LOCATE YC1
         ARG = AM(I,J)**2 - 1.
         IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0 at loc. G8...' WRITE(6,*),'   AM(I,J) = ',AM(I,J) WRITE(6,*),'  I = ',I,'
```

```
 X(I) = ',X(I) GOTO 3333
          ENDIF
 EM  = ATAN2(1.,SQRT(AM(I,J)**2 - 1.)) ARG = AM1**2 - 1.
          IF(ARG.LT.0.) THEN
 WRITE(6,*),'  ARG<0 at loc. G9...' WRITE(6,*),'   AM1 = ',AM1 WRITE(6,*),'  I = ',I,' X(I)
 = ',X(I) GOTO 3333
          ENDIF
          EMU =    .5*(EM + ATAN2(1.,(SQRT(AM1**2 - 1.))))
 PHI = EMU - .5*(TH(I,J) + TH1)
          YC1 = Y(I,J) + DX*TAN(PHI)
 DC1 = SQRT(DX**2 + (Y(I,J) - YC1)**2)

*———————— FIND NEAREST J-INDEX BELOW YC1
         DO 220 JZ = 2,JM
            J1 = JZ - 1
IF(YC1.LT.Y(I-1,JZ)) GO TO 225
                CALL SpinBeachBall(1)

  220 CONTINUE

*———————— COMPUTE PRESSURE & OTHER STUFF AT YC1 *                    4 pt Lagrange
interpolation
  225  CONTINUE
       JZ = J1
       YZ = YC1

       CALL LAGRANGE(I,JZ,YZ,YF1,YF2,YF3,YF4,JA,JB,JC,JD)

TH1 =  TH(I-1,JA)*YF1 + TH(I-1,JB)*YF2 +
    1                  TH(I-1,JC)*YF3 + TH(I-1,JD)*YF4
AM1 =  AM(I-1,JA)*YF1 + AM(I-1,JB)*YF2 +
    1                  AM(I-1,JC)*YF3 + AM(I-1,JD)*YF4
VT1 =  VT(I-1,JA)*YF1 + VT(I-1,JB)*YF2 +
    1                  VT(I-1,JC)*YF3 + VT(I-1,JD)*YF4
RH1 = RHO(I-1,JA)*YF1 + RHO(I-1,JB)*YF2 +
    1                  RHO(I-1,JC)*YF3 + RHO(I-1,JD)*YF4
      P1  =  P(I-1,JA)*YF1 + P(I-1,JB)*YF2 +
    1                  P(I-1,JC)*YF3 + P(I-1,JD)*YF4

       ARG = AM1**2 - 1.
       IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0 at loc. G10...' WRITE(6,*),'   AM1 = ',AM1 WRITE(6,*),'  I = ',I,' X(I)
= ',X(I) GOTO 3333
       ENDIF
       BT1  = SQRT(AM1**2 - 1.)
       BRV1 = BT1/(RH1*VT1**2)
       DY   = YC1 - Y(I,1)
CF1  = CFW/(1. + CFW*DY1*CAL)
       RHS1 = CF1*SIN(TH1)/AM1

*———————— LOCATE YC2
       ARG = AM2**2 - 1.
       IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0 at loc. G11...' WRITE(6,*),'   AM2 = ',AM2 WRITE(6,*),'  I = ',I,' X(I)
```

```
= ',X(I) GOTO 3333
        ENDIF
EMU = .5*(EM + ATAN2(1.,(SQRT(AM2**2 - 1.)))) PHI = EMU + .5*(TH(I,J) + TH2)
        YC2 = Y(I,J) - DX*TAN(PHI)
DC2 = SQRT(DX**2 + (Y(I,J) - YC2)**2)

        DO 230 JZ = 2,JM
           J2 = JZ - 1
        IF(YC2.LT.Y(I-1,JZ)) GO TO 235 230   CONTINUE

*———— COMPUTE PRESSURE & OTHER STUFF AT YC2 *                    Use   4   pt
Lagrange interpolation
   235 CONTINUE
        JZ = J2
        YZ = YC2

        CALL LAGRANGE(I,JZ,YZ,YF1,YF2,YF3,YF4,JA,JB,JC,JD)

        TH2 = TH(I-1,JA)*YF1 + TH(I-1,JB)*YF2 +
     1                TH(I-1,JC)*YF3 + TH(I-1,JD)*YF4
        AM2 = AM(I-1,JA)*YF1 + AM(I-1,JB)*YF2 +
     1                AM(I-1,JC)*YF3 + AM(I-1,JD)*YF4
        VT2 = VT(I-1,JA)*YF1 + VT(I-1,JB)*YF2 +
     1                VT(I-1,JC)*YF3 + VT(I-1,JD)*YF4
        RH2 = RHO(I-1,JA)*YF1 + RHO(I-1,JB)*YF2 +
     1                RHO(I-1,JC)*YF3 + RHO(I-1,JD)*YF4
        P2  =  P(I-1,JA)*YF1 + P(I-1,JB)*YF2 +
     1                P(I-1,JC)*YF3 + P(I-1,JD)*YF4

        ARG = AM2**2 - 1.
        IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0 at loc. G12...' WRITE(6,*),'   AM2 = ',AM2 WRITE(6,*),'   I = ',I,' X(I)
= ',X(I) GOTO 3333
        ENDIF
        BT2  = SQRT(AM2**2 - 1.)
        BRV2 = BT2/(RH2*VT2**2)

        DY2 = YC2 - Y(I,1)
CF2 = CFW/(1. + CFW*DY2*CAL)
        RHS2 = CF2*SIN(TH2)/AM2

*————————— LOCATE YS
        ARG = AMS**2 - 1.
        IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0 at loc. G13...' WRITE(6,*),'   AMS = ',AMS WRITE(6,*),'   I = ',I,' X(I)
= ',X(I) GOTO 3333
        ENDIF
EMU = .5*(EM + ATAN2(1.,(SQRT(AMS**2 - 1.)))) PHI = .5*(TH(I,J) + THS)
YS  = Y(I,J) - DX*TAN(PHI)
        DS  = SQRT(DX**2 + (Y(I,J) - YS)**2)

        DO 240 JZ = 2,JM
           JS = JZ - 1
        IF(YS.LT.Y(I-1,JZ)) GO TO 245 240    CONTINUE
```

```
*———— COMPUTE PRESSURE & OTHER STUFF AT YS *          4 pt Lagrange interpo-
lation
  245 CONTINUE
        JZ = JS
        YZ = YS

        CALL LAGRANGE(I,JZ,YZ,YF1,YF2,YF3,YF4,JA,JB,JC,JD)

        THS = TH(I-1,JA)*YF1 + TH(I-1,JB)*YF2 +
     1                  TH(I-1,JC)*YF3 + TH(I-1,JD)*YF4
        AMS = AM(I-1,JA)*YF1 + AM(I-1,JB)*YF2 +
     1                  AM(I-1,JC)*YF3 + AM(I-1,JD)*YF4
        VTS = VT(I-1,JA)*YF1 + VT(I-1,JB)*YF2 +
     1                  VT(I-1,JC)*YF3 + VT(I-1,JD)*YF4
RHOS = RHO(I-1,JA)*YF1 + RHO(I-1,JB)*YF2 +
     1                  RHO(I-1,JC)*YF3 + RHO(I-1,JD)*YF4
        PS  =  P(I-1,JA)*YF1 + P(I-1,JB)*YF2 +
     1                  P(I-1,JC)*YF3 + P(I-1,JD)*YF4

*———————— SOLVE FOR P(I,J) ————————
        ARG = AM(I,J)**2 - 1.
        IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0 at loc. G14...' WRITE(6,*),'   AM(I,J) = ',AM(I,J) WRITE(6,*),'  I = ',I,'
X(I) = ',X(I) GOTO 3333
        ENDIF
BRV = SQRT(AM(I,J)**2 - 1.)/(RHO(I,J)*VT(I,J)**2) BRV1 = .5*(BRV + BRV1)
BRV2 = .5*(BRV + BRV2)

RHS  = CFI*SIN(TH(I,J))/AM(I,J) RHS1 = .5*DC1*(RHS + RHS1) RHS2 = .5*DC2*(RHS +
RHS2)

           DTH = TH1 - TH2
           P(I,J) = (BRV1*P1 + BRV2*P2 - RHS1 - RHS2 - DTH)/(BRV1 + BRV2)

*———————— SOLVE FOR TH(I,J) ————————
           TH(I,J) = TH1 + BRV1*(P(I,J) - P1) + RHS1

*———————— SOLVE FOR RHO(I,J) ————————AA = .5*G*(P(I,J)/RHO(I,J) +
PS/RHOS) RHO(I,J) = RHOS + (P(I,J) - PS)/AA

*———————— SOLVE FOR VT(I,J) ————————ARG = H0INF - G*P(I,J)/
(GM*RHO(I,J)) IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0 at loc. G15...'
WRITE(6,*),'   H0INF = ',H0INF,'  P(I,J) = ',P(I,J) WRITE(6,*),'   RHO(I,J) = ',RHO(I,J)
WRITE(6,*),'  I = ',I,' X(I) = ',X(I)
              GOTO 3333
           ENDIF
           VT(I,J) = SQRT(2.*(H0INF - G*P(I,J)/(GM*RHO(I,J))))

*———————— SOLVE FOR OTHER STUFF ————————AM(I,J) = VT(I,J)/
SQRT(G*P(I,J)/RHO(I,J))
           TR  = 1. + .5*GM*AM(I,J)**2
           P0(I,J) = P(I,J)*TR**(G/GM)
```

```
*————— CHECK CONVERGENCE AT THIS POINT —————DP = ABS(P(I,J) - PL)
        IF(DP.GT..001) GO TO 210
  260   CONTINUE
*  WRITE(6,*),'  Done with interior points...starting shock...'
*———————————————————————————————————————
*                 POINT AT THE SHOCK
*———————————————————————————————————————
   DYI = Y(I,JM) - Y(I,1)
   CFI = CFW/(1. + CFW*DYI*CAL)

*————— INITIAL SHOCK ANGLE = LAST STATION
   WS(I) = WS(I-1)
   AM1  = AMINF
   AM2  = AM(I-1,JM)
   TH2  = TH(I-1,JM)

*—— GET QUANTITIES FROM OBLIQUE SHOCK RELATIONS
  410   CONTINUE
   KS       = KS + 1
   Y(I,JM) = Y(I-1,JM) + DX*TAN(WS(I))
   AMN      = AMINF*SIN(WS(I))
   PR       = (2.*G*AMN**2 - GM)/GP
   DEN      = .5*(GP*AMN)**2/GM
   TR       = (1. + .5*GM*AMN**2)*(2.*G*AMN**2/GM - 1.)/DEN
   RR       = PR/TR

   PL  = PINF*PR
   RHOL = RHOINF*RR
   F1  = AMN**2 - 1.
   F2  = G*AMN**2 + 1.
   F3  = (GP*AM1*AMN)**2
   ARG =    1. - 4.*F1*F2/F3
   IF(ARG.LT.0.) THEN
WRITE(6,*),' ARG<0. at loc. 1...' WRITE(6,*),'  AMN = ',AMN,'  AM1 = ',AM1 WRITE(6,*),'
I = ',I,'  X(I) = ',X(I) GOTO 3333
      ENDIF
      VTL  = VINF*SQRT(1. - 4.*F1*F2/F3)
      THL  = ATAN2(2.*F1,(TAN(WS(I))*
   1              (2. + (GP - 2.*SIN(WS(I))**2)*AM1**2)))
      AML  = SQRT((GM*AMN**2 + 2.)/
   1              ((2.*G*AMN**2 - GM)*SIN(WS(I) - THL)**2))
      VT(I,JM)  = VTL
      RHO(I,JM) = RHOL
      P(I,JM)   = PL
      AM(I,JM)  = AML
      TH(I,JM)  = THL

*————— NOW GET P(I,JM) FROM MOC —————*————— LOCATE
YC2
      ARG = AM(I,JM)**2 - 1.
      IF(ARG.LT.0.) THEN
WRITE(6,*),' ARG<0. at loc. 2...' WRITE(6,*),'  AM(I,JM) = ',AM(I,JM) WRITE(6,*),'  I = ',I,'
X(I) = ',X(I)
         GOTO 3333
```

```
        ENDIF
EM  = ATAN2(1.,SQRT(AM(I,JM)**2 - 1.)) ARG = AM2**2 - 1.
        IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0. at loc. 3...' WRITE(6,*),'   AM2 = ',AM2
WRITE(6,*),'   I = ',I,'  X(I) = ',X(I) GOTO 3333
        ENDIF
EMU = .5*(EM + ATAN2(1.,(SQRT(AM2**2 - 1.)))) PHI = EMU + .5*(TH(I,JM) + TH2)
        YC2 = Y(I,JM) - DX*TAN(PHI)
DC2 = SQRT(DX**2 + (Y(I,JM) - YC2)**2)

*————————— FIND J-LOCATION OF YC2
        DO 420 JZ = 2,JM
J2 = JZ - 1 IF(YC2.LT.Y(I-1,JZ)) GO TO 425
    420 CONTINUE

*————————— GET VALUES AT YC2
*    4 pt Lagrange interpolation
    425 CONTINUE
        JZ = J2
        YZ = YC2

CALL LAGRANGE(I,JZ,YZ,YF1,YF2,YF3,YF4,JA,JB,JC,JD)

        TH2 = TH(I-1,JA)*YF1 + TH(I-1,JB)*YF2 +
     1                 TH(I-1,JC)*YF3 + TH(I-1,JD)*YF4
        AM2 = AM(I-1,JA)*YF1 + AM(I-1,JB)*YF2 +
     1                 AM(I-1,JC)*YF3 + AM(I-1,JD)*YF4
        VT2 = VT(I-1,JA)*YF1 + VT(I-1,JB)*YF2 +
     1                 VT(I-1,JC)*YF3 + VT(I-1,JD)*YF4
        RH2 = RHO(I-1,JA)*YF1 + RHO(I-1,JB)*YF2 +
     1                 RHO(I-1,JC)*YF3 + RHO(I-1,JD)*YF4
        P2  =  P(I-1,JA)*YF1 + P(I-1,JB)*YF2 +
     1                 P(I-1,JC)*YF3 + P(I-1,JD)*YF4

        ARG = AM2**2 - 1.
        IF(ARG.LT.0.) THEN
          WRITE(6,*),'  ARG<0. at loc. 4...'
WRITE(6,*),'   AM2 = ',AM2
WRITE(6,*),'   I = ',I,'  X(I) = ',X(I) GOTO 3333
     ENDIF
     BT2  = SQRT(AM2**2 - 1.)
     BRV2 = BT2/(RH2*VT2**2)

*—————GET AVERAGED VALUES BETWEEN YC2 AND SHOCK AT I ARG = AM(I,JM)**2
- 1.
     IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0. at loc. 5...' WRITE(6,*),'   AM(I,JM) = ',AM(I,JM) WRITE(6,*),'   I = ',I,'
X(I) = ',X(I)
          GOTO 3333
     ENDIF
BRV  = SQRT(AM(I,JM)**2 - 1.)/(RHO(I,JM)*VT(I,JM)**2) BRV2 = .5*(BRV2 + BRV)

     DY2 = YC2 - Y(I,1)
     CF2 = CFW/(1. + CFW*DY2*CAL)
```

```
      RHS2 = CF2*SIN(TH2)/AM2
      RHS  = CFI*SIN(TH(I,JM))/AM(I,JM)
      RHS2 = .5*DC2*(RHS + RHS2)

*————COMPUTE CORRESPONDING VALUE OF P(I,JM) P(I,JM) = P2 - ((TH(I,JM)
- TH2) + RHS2)/BRV2
      TR     = 1. + .5*GM*AM(I,JM)**2
      P0(I,JM) = P(I,JM)*TR**(G/GM)

*————COMPARE WITH SHOCK VALUE DP = P(I,JM) - PL
*     WRITE(6,*),'  DP = ',DP
      IF(ABS(DP).LT..01) GO TO 450

*————PRESSURE NOT CONVERGED,GET NEW SHOCK ANGLE & Y-LOCATION PA
= P(I,JM)/PINF
      F1   = .5*(GP*PA + GM)/(G*AM1**2)
      ARG = 1. - F1
      IF(ARG.LT.0.) THEN
WRITE(6,*),'  ARG<0. at loc. 6...' WRITE(6,*),'  F1 = ',F1
WRITE(6,*),'  I = ',I,'  X(I) = ',X(I)
            GOTO 3333
      ENDIF
      WS(I) =    .5*(WS(I) + ATAN(SQRT(F1/(1. - F1))))
      GO TO 410

*————PRESSURE CONVERGED,SHOCK LOCATION SET
  450    CONTINUE
*     WRITE(2,*) X(I),Y(I,1),Y(I,JM)
*     WRITE(3,2) I,X(I),Y(I,1),AM(I,1),WS(I),P(I,1),VT(I,1)
*     2  FORMAT(I4,6F12.4)

*————WRITE TO BODY SURFACE DATA FILE ————

*            WRITE(55,*) X(I),AM(I,1),P(I,1),RHO(I,1)

*————GO TO NEXT I-STATION ————IF(X(I).LT.XF(NPF)) GO TO
100
*     CLOSE(2,STATUS='KEEP')
*     CLOSE(3,STATUS='KEEP')

*————THAT'S ALL ————990      CONTINUE
3333
            CALL RelBeachBall

       RETURN

       END
*
*
       SUBROUTINE CONEFLOW(DL) *————————————
————
PARAMETER     (IM=501,JM=15)

COMMON/DEPVR/ RHO(IM,JM),VT(IM,JM),T(IM,JM),P(IM,JM),
```

```
     1            TH(IM,JM),AM(IM,JM),WS(IM),H0(IM,JM),P0(IM,JM)
      COMMON/GASPR/ G,RG,GM,GP,CP
      COMMON/INFTY/ AMINF,PINF,TINF,ALPHA

      DIMENSION VR(JM),VW(JM),W(JM)

      CHARACTER*1 ZZZ
*-----------------------------------------------------------------
      PI  = 3.141592654

      T0 = TINF*(1. + .5*GM*AMINF**2)
      VMX  = SQRT(2.*CP*T0)
      CS = SQRT(G*RG*TINF)
SIG  = ATAN2(1.,SQRT(AMINF**2 - 1.)) DWS  = 0.
      VINF = AMINF*CS
      RHOINF = PINF/(RG*TINF)
      DLD  = 180.*DL/PI
*    WRITE(6,*),' In CONEFLOW...'
*    WRITE(6,*),' Tinf = ',TINF
*    WRITE(6,*),'  T0 = ',T0
*    WRITE(6,*),'  DL = ',DLD
*    WRITE(6,*),'  AMinf = ',AMINF

*------------------- FIRST GUESS ----------------------------
*  Note: If angle small enough, use Mach wave, otherwise, Doty's formula
*------------- DOTY'S FORMULA (for cone angle > .1 deg.)
      IF(DLD.GT..1) THEN
THS   = SIN(DL)*SQRT(.5*GP + 1./(AMINF*SIN(DL))**2)
        WS(1) = ATAN2(THS,SQRT(1. - THS**2))
      ELSE
*------------- MACH WAVE
        WS(1) = ATAN2(1.,SQRT(AMINF**2 - 1.))
      END IF *--------------------------------------------------------
*               MAIN LOOP
*-----------------------------------------------------------------
      ISTP = 0
  100 ISTP = ISTP + 1

*----------- GET STUFF FOR NEW SHOCK ANGLE ---------------
*  Ray angles
      DW = (WS(1) - DL)/(JM - 1.)
      DWI = 1./DW
      DO 120 J = 1,JM
        W(J) = DL + (J - 1.)*DW
  120 CONTINUE

*--------- GET INITIAL VR & VW AT SHOCK ---------------
*--------- VRS = tangential component. = on both sides of shock.
      VRS = VINF*COS(WS(1))

*--------- VWS = normal component. Use normal shock relations.
      VW1   = -VINF*SIN(WS(1))
      AMN   = AMINF*SIN(WS(1))
VWS   = VW1*(GM*AMN**2 + 2.)/(GP*AMN**2)
```

```
      VW(JM) = VWS

*———— SET UP FOR MARCHING ————
      VR(JM) = VRS
      VR(JM - 1) = VR(JM) - DW*VWS

*———————— MARCH TO SURFACE ————————DO 250 JB = 1,JM - 2
            J   = JM - JB
            VRL  = VR(J)
            KSTP = 0

*——— ITERATE ON VR AT THIS POINT
  200       CONTINUE
            KSTP = KSTP + 1
            Z = VMX**2 - VR(J)**2
            IF(KSTP.EQ.1) THEN
               DVR = DWI*(VR(J+1) - VR(J))
            ELSE
DVR = .5*DWI*(VR(J+1) - VR(J-1))
            END IF
            VW(J) = DVR
*           VR(J) = VR(J+1) - .5*DW*(VW(J) + VW(J+1))

            F1 = .5*GM*Z
            F2 = DVR
            TW = TAN(W(J))

            AK1 = F1 - .5*GP*F2**2
            AK2 = F1/TW
            AK3 = 2.*F1 - G*F2**2

            CC = (AK1*DWI - .5*AK2)*DWI
            BB = AK3 - 2.*AK1*DWI**2
            AA = (AK1*DWI + .5*AK2)*DWI
            DD = .5*GM*F2**3/TW

VR(J-1) = (DD - BB*VR(J) - AA*VR(J+1))/CC DVRK   = ABS(VR(J) - VRL)
*           VRL    = VR(J)
VV    = SQRT(VR(J-1)**2 + DVR**2)
*           IF(KSTP.EQ.1) GO TO 200
*           IF(DVRK.GT..001) GO TO 200

  250   CONTINUE

*———— CHECK CONVERGENCE (IS VW AT WALL = 0?) ———VWLL = VWL
    VWL  = VWC
    VWC  = .5*DWI*(4.*VR(2) - 3.*VR(1) - VR(3))
*     WRITE(6,*),'   Vwall = ',VWC,'  Wshock = ',180.*WS(1)/PI,' deg'
*     WRITE(6,*),'      Vwall(1st-order) = ',DWI*(VR(2) - VR(1))
    IF(ISTP.GT.5) GO TO 300
    IF(ABS(VWC).LT..5) GO TO 300

*————NOT CONVERGED, SET UP FOR CONTINUE LOOP————*———Use computed
VWS to find VW1 and corresponding shock angle.
```

```
OM   = DL/SIG
    WSLL = WSL
    WSL  = WS(1)
    IF(ISTP.LT.3) THEN
*           WS(1) = DL + .5*SIG
            VWS = VWS + .15*VWC
VW1 = VWS*(GP*AMN**2)/(GM*AMN**2 + 2.) WS(1) = ATAN2(-VW1,SQRT(VINF**2 -
VW1**2))
    ELSE
            EM     = (VWC - VWL)/(WSL - WSLL)
   WS(1) = .7*WS(1) + .3*(WSL - VWC/EM) END IF
*   IF(WS(1).LT.SIG) WS(1) = DL + .5*SIG

    GO TO 100

*----------- DONE ---------------
   300     CONTINUE

*------------ GET RATIOS ACROSS SHOCK ------------
   AMST = AMINF*SIN(WS(1))
   PRS  = (2.*G*AMST**2 - GM)/GP
   RRS  = GP*AMST**2/(GM*AMST**2 + 2.)
   TRS  = PRS/RRS

*-------- NOW GET PROPERTIES FOR FIRST STATION ----------
*  WRITE(6,*),'    Getting properties for first station...'
   DO 400 J = 1,JM
           ST      = SIN(W(J))
           CT      = COS(W(J))
           UU      = VR(J)*CT - VW(J)*ST
           VV      = VR(J)*ST + VW(J)*CT
VT(1,J)  = SQRT(UU**2 + VV**2) TH(1,J)  = ATAN2(VV,UU)
           CS      = .5*GM*(VMX**2 - VT(1,J)**2)
           AM(1,J)  = VT(1,J)*SQRT(1./CS)
  400      CONTINUE
   AMY = AM(1,JM)

   DO 420 J = 1,JM
           TR      = (1. + .5*GM*AMY**2)/(1. + .5*GM*AM(1,J)**2)
           RHOR      = TR**(1./GM)
           RHO(1,J) = RHOINF*RHOR*RRS
           PR      = RHOR**G
           P(1,J)  = PINF*PR*PRS
           T(1,J)  = TINF*TR*TRS
  420      CONTINUE
*----------------------------------------------------------
3333
              CALL RelBeachBall

   RETURN
   END
*
*
```

```
*-------------------------------------------------------------------SUBROUTINE
LAGRANGE(I,JZ,YZ,YF1,YF2,YF3,YF4,JA,JB,JC,JD)
*--------------------------------------------------------------------PARAMETER
(IM=501,JM=15)

   COMMON/GRIDS/ X(IM),Y(IM,JM)

*---------------------------------------------------------------JPM = 0
   IF(JZ.EQ.1) JPM = 1
   IF(JZ.EQ.(JM-1)) JPM = -1

   JA = JZ - 1 + JPM
   JB = JZ + JPM
   JC = JZ + 1 + JPM
   JD = JZ + 2 + JPM

   YA = Y(I-1,JA)
      YB = Y(I-1,JB)
      YC = Y(I-1,JC)
      YD = Y(I-1,JD)

      YF1 = (YZ - YB)*(YZ - YC)*(YZ - YD)/
   1              ((YA - YB)*(YA - YC)*(YA - YD))
      YF2 = (YZ - YA)*(YZ - YC)*(YZ - YD)/
   1              ((YB - YA)*(YB - YC)*(YB - YD))
      YF3 = (YZ - YA)*(YZ - YB)*(YZ - YD)/
   1              ((YC - YA)*(YC - YB)*(YC - YD))
      YF4 = (YZ - YA)*(YZ - YB)*(YZ - YC)/
   1              ((YD - YA)*(YD - YB)*(YD - YC))

      RETURN
      END
*
* *-----------------------------------------------------------------
SUBROUTINE CUBIC(N,X,AX,BX,CX)
*----------------------------------------------------------------*
*     This subroutine computes the (N-1) parametric cubic spline
*     coefficients for an array of N numbers. The parameter, t,
*     varies between 0 and 1 on each cubic. Natural end condi-
*     tions are used here.
* *-----------------------------------------------------------------
PARAMETER (IM=501)

      DIMENSION X(IM),AX(IM),BX(IM),CX(IM)
      DIMENSION A(IM),B(IM),C(IM),D(IM)
      CALL InitBeachBall(512)
      CALL ShowBeachBall

*------------------ SET UP MATRIX ------------------
      DO 100 I = 2,N-2
         C(I) = 1.
         B(I) = 4.
         A(I) = 1.
  100 CONTINUE
```

```
*------------------------ RHS
      DO 120 I = 2,N-1
         D(I) =  3.*(X(I+1) - 2.*X(I) + X(I-1))
  120 CONTINUE

*------------- END CONDITIONS - NATURAL --------------B(1)  =  1.
      A(1)  =   0.
      D(1)  =   0.
      C(N-1) =  1.
      B(N-1) =  4.

*--------------- SOLVE MATRIX ---------------------DO 200 I = 2,N-1
         CBI = C(I) / B(I-1)
         B(I) = B(I) - CBI * A(I-1)
D(I) = D(I) - CBI * D(I-1)
               CALL SpinBeachBall(1)

  200 CONTINUE

D(N-1) = D(N-1) / B(N-1)

      DO 220 IR = 2,N-1
         I = N - IR
   D(I) = (D(I) - A(I) * D(I + 1)) / B(I) 220 CONTINUE

*--------------- NOW GET COEFFS -------------------DO 240 I = 1,N-1
         BX(I) = D(I)
  240    CONTINUE
   DO 260 I = 1,N-2
         AX(I) = (BX(I+1) - BX(I)) / 3.
CX(I) = X(I+1) - X(I) - AX(I) - BX(I)
  260    CONTINUE
CX(N-1) = 3.*AX(N-2) + 2.*BX(N-2) + CX(N-2) AX(N-1) = X(N) - BX(N-1) - CX(N-1) - X(N-
1)

*--------------------------------------------------CALL RelBeachBall

   RETURN

   END
*
*
SUBROUTINE INTRPC(X,XS,AX,BX,CX,N,IT,T)
*-----------------------------------------------*
*  This subroutine finds where XS is located within
*  an array of N points X.  The subroutine first
*  finds which two X's, X(IT) & X(IT+1), XS is in
*  between and then finds the value of the parameter,
*  t, which locates XS on the cubic between them.
*  This value of t may then be used for finding the
*  corresponding interpolated values of related arrays.
* *-----------------------------------------------
PARAMETER   (IM=501)
```

```
      DIMENSION   X(IM),AX(IM),BX(IM),CX(IM)

*———————— FIND WHICH TWO POINTS XS IS BETWEEN ————————DO 100 I = 2,N
            IT = I - 1
            IF(XS.LT.X(I)) GOTO 200
  100       CONTINUE

*——————————— NOW FIND THE PARAMETER VALUE ——————————200       CON-
TINUE

*—————————— FIRST GUESS BY LINEAR INTERPOLATION
     DX = X(IT+1) - X(IT)
     T  = (XS - X(IT))/DX

*—————————— NEWTON ITERATION
   300     TL = T
F = ((AX(IT)*T + BX(IT))*T + CX(IT))*T + X(IT) - XS FP = (3.*AX(IT)*T + 2.*BX(IT))*T +
CX(IT)
     T  = T - F/FP
     IF(ABS(T-TL).GE.1.E-5) GO TO 300

*——————————— ALL DONE ————————————
     RETURN
     END
*
*
     SUBROUTINE STEPMOC(DX,YY,AMN,PN,TN,RHN,THN,VTN) *——————————
———————————————————————————————*
*       This subroutine computes properties at a station downstream of
*       the last forebody station using the Method of Characteristics.
* *————————————————————————————————————————
PARAMETER    (IM=501,JM=15,JI=21,NST=5)

COMMON/DEPVR/ RHO(IM,JM),VT(IM,JM),T(IM,JM),P(IM,JM),
    1              TH(IM,JM),AM(IM,JM),WS(IM),H0(IM,JM),P0(IM,JM)
        COMMON/GASPR/ G,RG,GM,GP,CP
        COMMON/GRIDS/ X(IM),Y(IM,JM)
        COMMON/GRIDF/ XF(IM),YF(IM),CF(IM),ILF
        COMMON/INFTY/ AMINF,PINF,TINF,ALPHA

*—————————————————————————————————————————————*——————
— USEFUL STUFF ——————
RHOINF = PINF/(RG*TINF)
V1    = AMINF*SQRT(G*RG*TINF) VINF  = V1
T0INF = TINF*(1. + .5*GM*AMINF**2) CP    = G*RG/GM
        H0INF  = CP*T0INF

*—————————— FIND NEAREST J
        DO 100 J = 2,JM
            JZ = J - 1
        IF(YY.LT.Y(ILF,J)) GO TO 105 100    CONTINUE

*—————————— INITIAL GUESS
   105 CONTINUE
```

```
        PN  = P(ILF,JZ)
        AMN = AM(ILF,JZ)
        RHN = RHO(ILF,JZ)
        VTN = VT(ILF,JZ)
        THN = TH(ILF,JZ)

        AM1 = AMN
        TH1 = THN
        AM2 = AMN
        TH2 = THN
        AMS = AMN
        THS = THN

*—————————— ITERATION LOOP AT THIS POINT ——————————210    CONTINUE
        PL = PN
*———— LOCATE YC1
EM = ATAN2(1.,SQRT(AMN**2 - 1.))
EMU = .5*(EM + ATAN2(1.,(SQRT(AM1**2 - 1.)))) PHI = EMU - .5*(THN + TH1)
        YC1 = YY + DX*TAN(PHI)
        DC1 = SQRT(DX**2 + (YY - YC1)**2) *———— FIND NEAREST J-INDEX
BELOW YC1
        DO 220 JZ = 2,JM
J1 = JZ - 1 IF(YC1.LT.Y(ILF,JZ)) GO TO 225
   220 CONTINUE

*———— COMPUTE PRESSURE & OTHER STUFF AT YC1 *                4 pt Lagrange
interpolation
   225 CONTINUE
        JZ = J1
        YZ = YC1
        I = ILF + 1
        CALL LAGRANGE(I,JZ,YZ,YF1,YF2,YF3,YF4,JA,JB,JC,JD)

   P1 =  P(ILF,JA)*YF1 +  P(ILF,JB)*YF2 + 1          P(ILF,JC)*YF3 +  P(ILF,JD)*YF4
   TH1 = TH(ILF,JA)*YF1 + TH(ILF,JB)*YF2 + 1          TH(ILF,JC)*YF3        +
TH(ILF,JD)*YF4
   AM1 = AM(ILF,JA)*YF1 + AM(ILF,JB)*YF2 + 1          AM(ILF,JC)*YF3        +
AM(ILF,JD)*YF4
   VT1 = VT(ILF,JA)*YF1 + VT(ILF,JB)*YF2 + 1          VT(ILF,JC)*YF3        +
VT(ILF,JD)*YF4
   RH1 = RHO(I-1,JA)*YF1 + RHO(ILF,JB)*YF2 + 1             RHO(ILF,JC)*YF3   +
RHO(ILF,JD)*YF4

        BT1  = SQRT(AM1**2 - 1.)
        BRV1 = BT1/(RH1*VT1**2)
        RHS1 = SIN(TH1)/(AM1*YC1)

*—————————— LOCATE YC2
EMU = .5*(EM + ATAN2(1.,(SQRT(AM2**2 - 1.)))) PHI = EMU + .5*(THN + TH2)
        YC2 = YY - DX*TAN(PHI)
DC2 = SQRT(DX**2 + (YY - YC2)**2)
        DO 230 JZ = 2,JM
           J2 = JZ - 1
           IF(YC2.LT.Y(ILF,JZ)) GO TO 235
```

```
  230 CONTINUE

*——— COMPUTE PRESSURE & OTHER STUFF AT YC2 *                Use 4 pt
Lagrange interpolation
  235 CONTINUE
      JZ = J2
      YZ = YC2
      CALL LAGRANGE(I,JZ,YZ,YF1,YF2,YF3,YF4,JA,JB,JC,JD)

      P2 = P(ILF,JA)*YF1 + P(ILF,JB)*YF2 +
  1        P(ILF,JC)*YF3 + P(ILF,JD)*YF4
  TH2 = TH(ILF,JA)*YF1 + TH(ILF,JB)*YF2 + 1         TH(ILF,JC)*YF3      +
TH(ILF,JD)*YF4
  AM2 = AM(ILF,JA)*YF1 + AM(ILF,JB)*YF2 + 1         AM(ILF,JC)*YF3      +
AM(ILF,JD)*YF4
  VT2 = VT(ILF,JA)*YF1 + VT(ILF,JB)*YF2 + 1         VT(ILF,JC)*YF3      +
VT(ILF,JD)*YF4
  RH2 = RHO(ILF,JA)*YF1 + RHO(ILF,JB)*YF2 + 1             RHO(ILF,JC)*YF3 +
RHO(ILF,JD)*YF4

      BT2 = SQRT(AM2**2 - 1.)
      BRV2 = BT2/(RH2*VT2**2)
      RHS2 = SIN(TH2)/(AM2*YC2)

*———— LOCATE YS
EMU = .5*(EM + ATAN2(1.,(SQRT(AMS**2 - 1.)))) PHI = .5*(THN + THS)
YS = YY - DX*TAN(PHI)
DS = SQRT(DX**2 + (YY - YS)**2) DO 240 JZ = 2,JM
JS = JZ - 1 IF(YS.LT.Y(ILF,JZ)) GO TO 245
  240 CONTINUE

*——— COMPUTE PRESSURE & OTHER STUFF AT YS *          4 pt Lagrange interpo-
lation
  245 CONTINUE
      JZ = JS
      YZ = YS
      CALL LAGRANGE(I,JZ,YZ,YF1,YF2,YF3,YF4,JA,JB,JC,JD)

      PS = P(ILF,JA)*YF1 + P(ILF,JB)*YF2 +
  1        P(ILF,JC)*YF3 + P(ILF,JD)*YF4
  THS = TH(ILF,JA)*YF1 + TH(ILF,JB)*YF2 + 1         TH(ILF,JC)*YF3      +
TH(ILF,JD)*YF4
  AMS = AM(ILF,JA)*YF1 + AM(ILF,JB)*YF2 + 1            AM(ILF,JC)*YF3   +
AM(ILF,JD)*YF4
  VTS = VT(ILF,JA)*YF1 + VT(ILF,JB)*YF2 + 1         VT(ILF,JC)*YF3      +
VT(ILF,JD)*YF4
  RHOS = RHO(ILF,JA)*YF1 + RHO(ILF,JB)*YF2 + 1      RHO(ILF,JC)*YF3     +
RHO(ILF,JD)*YF4

*———— SOLVE FOR PN ————BRV = SQRT(AMN**2 - 1.)/
(RHN*VTN**2)
      RHS = SIN(THN)/(AMN*YY)

      BRV1 =  .5*(BRV + BRV1)
```

```
        RHS1 =   .5*DC1*(RHS + RHS1)

        BRV2 =   .5*(BRV + BRV2)
        RHS2 =   .5*DC2*(RHS + RHS2)

        DTH = TH1 - TH2
        PN  = (BRV1*P1 + BRV2*P2 - RHS1 - RHS2 - DTH)/(BRV1 + BRV2)

*---------------- SOLVE FOR THN ----------------
THN = TH1 + BRV1*(PN - P1) + RHS1

*---------------- SOLVE FOR RHN ----------------AA = .5*G*(PN/RHN + PS/
RHOS)
        RHN = RHOS + (PN - PS)/AA

*----------------SOLVE FOR VTN----------------ARG = H0INF - G*PN/(GM*RHN)
        VTN = SQRT(2.*ARG)

*---------------- SOLVE FOR OTHER STUFF ----------------AMN = VTN/SQRT(G*PN/
RHN)
                TN = PN/(RG*RHN)

*---------- CHECK CONVERGENCE AT THIS POINT ----------DP = ABS(PN - PL)
        IF(DP.GT..001) GO TO 210

*--------------------------------------------------------------------RETURN
        END
*
*
SUBROUTINE INLET(ISTR,AMI,PTI,AREAI,AMAV,PTAV,AREAO)
*------------------------------------------------------------------------*
          SCRAMJET - INLET  ANALYSIS
*
*   This routine determines the internal flow field in the inlet area
*   of a hypersonic SCRAMjet engine under the assumption of inviscid
*   airflow using two-dimensional oblique shock/expansion relations.
*   The inlet flow involves sidewall compression and diamond-shaped
*   struts of arbitrary location and orientation.
*
*   Written by Harald Buschek.
*
*                    * ** GEOMETRICAL DATA**
*
*   H Width of inlet entrance
*   XW1 Location of change of sidewedge 1
*   XW2 Location of change of sidewedge 2
*   D1    Sidewedge angle 1 before change (deg)
*   D1ST Sidewedge angle 1 after change (deg)
*   D2    Sidewedge angle 2 before change (deg)
*      D2ST      Sidewedge angle 2 after change (deg)
*      NSTRUT  Number of struts
*      XD()       Horizontal locations of struts
*      YD()       Vertical locations of struts
*      DT()       Horizontal lengths of the struts (2*DT)
```

```
*      DD()     Vertical widths of the struts (2*DD)
*      XCC      Entrance of the combustion chamber
*      DELST    Angle of mean chord of strut to horizontal axis
*      ALST     Sweep angle of the struts (compliment to angle wrt horiz)
**___________________________________________________________________
      PARAMETER (NGMAX=23,NLMAX=201,NFMAX=200,NST=5)

            DIMENSION XG(NGMAX),ITGEO(NGMAX),JSTRUT(NGMAX)
      DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
     1        ALF(NFMAX),ITLIN(NLMAX)
       DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),1
        PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

           LOGICAL LSHANG,LOUT,LNW2,LNW6 .

      COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
     1         THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT
           COMMON/INLTF/ DFUW,DFLW
      COMMON/GEO/  H,H1,XW1,XW2,D1,D1ST,D2,D2ST,XD(NST),YD(NST),
     1         DT(NST),DD(NST),XCC,DELST(NST),JSTRUT,ALST
           COMMON/GASPR/ G,RG,GM,GP,CP
           COMMON/STAGN/ T0INF,H0INF
           COMMON/STRUT/ NSTRUT

*____________________________________________________________________
      FORMAT (/,17HGEOMETRICAL DATA:/,10HINLET: H= ,F7.3/,17HLOWER W___901
     &EDGE: D1= ,F7.3,7H D1ST= ,F7.3,6H XW1= ,F7.3/,17HUPPER WEDGE: D2= & ,F7.3,7H
     D2ST= ,F7.3,6H XW2= ,F7.3/,34HCOMBUSTION CHAMBER ENTRAN &CE AT X= ,F5.2)
 902 FORMAT (17HINCOMING AIRFLOW:/,3HM= ,F7.3,4H P= ,E10.3,5H PT= ,E10
     &.3,4H G= ,F5.2/)
 903 FORMAT (5HSTRUT,I2/,4HXD= ,F7.3,5H YD= ,F7.3,11H LENGTH= 2*,F6.3,
     &14H THICKNESS= 2*,F6.3,8H ANGLE= ,F7.3)
 911 FORMAT (/,5HAREA ,I3)
 912 FORMAT (53HINLET ENTRANCE, FLOW AFFECTED BY SIDEWALL-COMPRES-
     SION)
 913 FORMAT (15HNEW FLOW FIELD:)
 914 FORMAT (5HLINE ,I3,5H YL= ,F7.3,8H THETA= ,F7.3,7H TYPE= ,I1)
 915 FORMAT (5HFIELD,I3,5H M=  ,F7.3,8H ALFA=  ,F7.3,4H P= ,E11.4,5H PT
     &= ,E12.5)
 920 FORMAT (34HCOMBUSTION CHAMBER ENTRANCE AT X= ,F7.3/,16HFINAL FLOW
     &FIELD:)
 930 FORMAT (/,/,21HFINAL PT-DISTRIBUTION)
 931 FORMAT (11HWALLS AT Y=,F7.3,7H AND Y=,F7.3/,4H   Y,10X,2HPT)
 932 FORMAT (F7.3,2X,E12.5)
 933 FORMAT (/,15H  AVERAGE PT=  ,E12.5,19H  AVER. MACH NO.=  ,F7.3)
*
*          OPEN (19,FILE='INLETG.OUT',STATUS='UNKNOWN')
           OPEN (2,FILE='FLOW1.DAT',STATUS='UNKNOWN')
*___________________________________________________________________
0
*     WRITE(6,*),'   In INLET.....NSTRUT = ',NSTRUT________NSTRUT=
        PI = 3.141592654
        NG = 3 + 4*NSTRUT
```

```
      DZ = AREAI/H1

      AMINF = AMI
      PTINF = PTI

*     WRITE (19,901) H,D1,D1ST,XW1,D2,D2ST,XW2,XCC
*   DO 5 I = 1,NSTRUT
*     WRITE (19,903) I,XD(I),YD(I),DT(I),DD(I),DELST(I)
*  5CONTINUE

*——— GEOMETRICAL EVENT TYPES
*   ITGEO = 1: CHANGE OF WEDGE 1
*           2: CHANGE OF WEDGE 2
*           3: BEGINNING OF STRUT
*           4: CHANGE OF UPPER PART OF STRUT
*           5: CHANGE OF LOWER PART OF STRUT
*           6: END OF STRUT
*           7: COMBUSTION CHAMBER ENTRANCE

*——— WEDGE 1
    XG(1) = XW1
    ITGEO(1) = 1
    JSTRUT(1) = 0
*——— WEDGE 2
    XG(2) = XW2
    ITGEO(2) = 2
    JSTRUT(2) = 0
*——— COMBUSTION CHAMBER
    XG(3) = XCC
    ITGEO(3) = 7
    JSTRUT(3) = 0
*——— STRUTS
    DO 6 I = 1,NSTRUT
        K      = 4*I
        DELST(I) = DELST(I)*PI/180.
        XG(K) = XD(I)
        ITGEO(K)= 3
        JSTRUT(K)   = I
        XG(K+1)  = XD(I)+DT(I)*COS(DELST(I))-DD(I)*SIN(DELST(I))
        ITGEO(K+1)  = 4
        JSTRUT(K+1) = I
        XG(K+2)  = XD(I)+DT(I)*COS(DELST(I))+DD(I)*SIN(DELST(I))
        ITGEO(K+2)  = 5
        JSTRUT(K+2) = I
        XG(K+3)  = XD(I)+2.*DT(I)*COS(DELST(I))
        ITGEO(K+3)  = 6
        JSTRUT(K+3) = I
  6CONTINUE

*——— SORT GEOMETRICAL ARRAY XG IN ASCENDING NUMERICAL ORDER DO 12
J = 2,NG
        A  = XG(J)
        K  = ITGEO(J)
        KS = JSTRUT(J)
```

```
          DO 10 I = J-1,1,-1
             IF (XG(I).LE.A) GO TO 11
             XG(I+1)      = XG(I)
             ITGEO(I+1)  = ITGEO(I)
             JSTRUT(I+1) = JSTRUT(I)
 10          CONTINUE
          I       = 0
 11          XG(I+1)    = A
          ITGEO(I+1)  = K
          JSTRUT(I+1) = KS
 12       CONTINUE

*------------ INCOMING AIRFLOW
*     AM  MACH NUMBER
*     P   STATIC PRESSURE
*     PT  TOTAL PRESSURE
*     TT  TOTAL TEMPERATURE
*     ALF FLOW ANGLE (DEG)
*   G   RATIO OF SPECIFIC HEATS
*   TH  ANGLE OF LINES AT BEGINNING OF AREA
*   YL  VERTICAL LOCATION OF LINES AT BEGINNING OF AREA
*
*   ITLIN, CHARACTERISATION OF LINE-TYPES
*              = 1 SHOCK WAVE, FAMILY 1 (DOWNSTREAM RIGHT RUNNING)
*              = 2 SHOCK WAVE, FAMILY 2 (DOWNSTREAM LEFT RUNNING)
*              = 3 EXPANSION WAVE, FAMILY 1
*              = 4 EXPANSION WAVE, FAMILY 2
*              = 5 SLIP LINE
*              = 6 SOLID LINE, WALL

   PINF = PTINF/(1.+ .5*GM*AMINF**2)**(G/GM) *        WRITE                 (19,902)
AMINF,PINF,PTINF,G

   LSHANG =    .TRUE.

*---------- UPPER WEDGE (WEDGE 2)
   D2  = ABS(D2)
*          WRITE(19,*) ' SHANG loc 1...D2 = ',180.*D2/PI,' AM = ',AMINF
   CALL SHANG (AMINF,D2,G,LSHANG,THY,PY,PTY,AMY)
*          WRITE(19,*) '    THY = ',180.*THY/PI
*          WRITE(19,*) '     PY = ',PY
*          WRITE(19,*) '    PTY = ',PTY
*          WRITE(19,*) '    AMY = ',AMY
   D2        = -D2
   AM(1)   = AMY
   P(1)     = PINF*PY
   PT(1)    = PTINF*PTY
   ALF(1)  = D2
   TH(1)    = D2
   TH(2)    = -THY
   ITLIN(1) = 6
   ITLIN(2) = 1
   YL(1)    = H
   YL(2)    = H
```

```
*——— UNAFFECTED FIELD
    AM(2)  = AMINF
    P(2)   = PINF
    PT(2)  = PTINF
    ALF(2) = 0.

*——— LOWER WEDGE (WEDGE 1)
*         WRITE(19,*) ' SHANG loc 2...D1 = ',180.*D1/PI,' AM = ',AMINF
    CALL SHANG (AMINF,D1,G,LSHANG,THY,PY,PTY,AMY)
    AM(3)    = AMY
    P(3)     = PINF*PY
    PT(3)    = PTINF*PTY
    ALF(3)   = D1
    TH(3)    = THY
    TH(4)    = D1
    ITLIN(3) = 2
    ITLIN(4) = 6
    YL(3)    = 0.
    YL(4)    = 0.

*——— PRINT DATA OF FIRST AREA
*         LOUT = .TRUE.
    NAREA = 1
    NL    = 4
    IF (LOUT) THEN
*            WRITE (19,911) NAREA
*            WRITE (19,912)
*            WRITE (19,913)
            DO 50 I = 1,4
              THDG = TH(I)*180./PI
              ALDG = ALF(I)*180./PI
*              WRITE (19,914) I,YL(I),THDG,ITLIN(I)
              IF (I.EQ.4) GO TO 50
*              WRITE (19,915) I,AM(I),ALDG,P(I),PT(I)
 50           CONTINUE
    END IF
*    WRITE (2,*) NL,X0
    DO 51 I = 1,NL
*         WRITE (2,*) YL(I),TH(I),ITLIN(I)
 51      CONTINUE

*——————————————————————————————————————————————————————*
  MAIN LOOP TO CALCULATE THE INLET FLOWFIELD AREA BY AREA
*————————————————————————————————————————————————————JLOOP =
-1
    IPRINT = 1
    IG     = 1
    X0     = 0.
  100      CONTINUE
    JUSE  = 1
    NAREA = NAREA+1
    NWALL = 0
*    IF (LOUT) WRITE (19,911) NAREA
```

```
*---------- STORE DATA OF PREVIOUS AREA X0OLD = X0
    DO 105 I = 1,NL
YLOLD(I)  = YL(I)
          THOLD(I)  = TH(I)
          ITLOLD(I) = ITLIN(I)
  105     CONTINUE
    DO 106 I = 1,NL-1
          AMOLD(I)  = AM(I)
          POLD(I)   = P(I)
          PTOLD(I)  = PT(I)
          ALFOLD(I) = ALF(I)
  106     CONTINUE
*------------ DETERMINATION OF NEXT INTERSECTION XMIN = XG(IG)
    IMIN = 0
    DO 110 I = 1,NL-1
          IF (TH(I).GE.TH(I+1)) GO TO 110
TTT =TAN(TH(I))-TAN(TH(I+1)) XINT = (TTT*X0+YL(I+1)-YL(I))/TTT IF (XINT.LT.XMIN)
THEN
             XMIN = XINT
             IMIN = I
          END IF
  110     CONTINUE
    X0 = XMIN

*--------- SUMMING UP THE FORCES
*------- UPPER WEDGE
    DFUW = DFUW+POLD(1)*(X0-X0OLD)/COS(THOLD(1))

*------- LOWER WEDGE
    DFLW = DFLW+POLD(NL-1)*(X0-X0OLD)/COS(THOLD(NL))

    DO 120 I=2,NL-1
IF (ITLIN(I).EQ.6) NWALL = NWALL+1
  120     CONTINUE
    IF (NWALL.EQ.0) GO TO 140
    IF (NWALL.EQ.2) GO TO 125
    IF (NWALL.EQ.6) GO TO 130

*------- ONE STRUT
  125     CONTINUE
    DO 126 I=2,NL-1
          IF (ITLIN(I).EQ.6) THEN
FST3 = FST3+POLD(I-1)*(X0-X0OLD)/COS(TH(I)) FST4 = FST4+POLD(I+1)*(X0-X0OLD)/
COS(TH(I+1)) GO TO 140
          END IF
  126     CONTINUE
    GO TO 140

*------- TWO STRUTS
  130     CONTINUE
    K = 1
    DO 131 I=2,NL-1
          IF (ITLIN(I).EQ.6) THEN
```

```
              K = K+1
              IF (K.EQ.2) THEN
FST1 = FST1+POLD(I-1)*(X0-X0OLD)/COS(TH(I)) FST2 = FST2+POLD(I+1)*(X0-X0OLD)/
COS(TH(I+1))
              ELSE IF (K.EQ.4) THEN
FST3 = FST3+POLD(I-1)*(X0-X0OLD)/COS(TH(I)) FST4 = FST4+POLD(I+1)*(X0-X0OLD)/
COS(TH(I+1))
              ELSE IF (K.EQ.6) THEN
FST5 = FST5+POLD(I-1)*(X0-X0OLD)/COS(TH(I)) FST6 = FST6+POLD(I+1)*(X0-X0OLD)/
COS(TH(I+1)) GO TO 140
              END IF
          END IF
  131     CONTINUE

*-------- CHECK IF GEOMETRICAL EVENT
  140     CONTINUE

*----------- ELIMINATE NUMERICAL INACCURACIES
DELX = ABS(X0-XG(IG))
IF(ITLIN(IMIN).EQ.6.AND.ITLIN(IMIN+1).EQ.6.AND.DELX.LE..05) THEN X0 = XG(IG)
          IMIN = 0
   END IF
   IF (IMIN.EQ.0) THEN
          IG = IG+1
  GO TO (151,152,153,151,152,154,155), ITGEO(IG-1) END IF
   GO TO 200

*------------ GEOMETRICAL EVENTS ------------*---------CHANGE OF WEDGE
1 OR UPPER PART OF STRUT
  151     CALL UPEXP (G,IMIN,NL,IG,*400,*330)

*-------- CHANGE OF WEDGE 2 OR LOWER PART OF STRUT 152          C A L L
LOWEXP (G,IMIN,NL,IG,*400,*330)

*-------- BEGINNING OF STRUT
  153     CALL BSTRUT (G,IMIN,NL,IG,*400)

*-------- END OF STRUT
  154     CALL ESTRUT (G,IMIN,NL,IG,*300,*350)

*-------- COMBUSTION CHAMBER ENTRANCE
  155     JLOOP = 1
  LOUT = .TRUE.
  DO 156 I = 1,NL
  YL(I) = TAN(TH(I))*(X0-X0OLD)+YLOLD(I) 156    CONTINUE
*  IF (LOUT) WRITE (19,920) X0
*  WRITE (19,911) NAREA
  GO TO 401

*----------------------------------------------*         ANALYSIS OF THE
FLOWFIELD EVENTS
*-----------------------------------------------200       CONTINUE
  GO TO (201,202,203,204,205,206) ITLIN(IMIN)
```

```
*———— UPPER LINE IS SHOCK, FAM. 1
  201    GO TO (212,211,215,214,220,221) ITLIN(IMIN+1)

*———— UPPER LINE IS SHOCK, FAM. 2
  202    GO TO (225,212,225,216,220,221) ITLIN(IMIN+1)

*———— UPPER LINE IS EXPANSION WAVE, FAM. 1 203    GO    TO
(215,213,218,217,223,224) ITLIN(IMIN+1)

*———— UPPER LINE IS EXPANSION WAVE, FAM. 2 204    GO    TO
(225,216,225,218,223,224) ITLIN(IMIN+1)

*———— UPPER LINE IS SLIP LINE
  205    GO TO (219,219,222,222,225,225) ITLIN(IMIN+1)

*———— UPPER LINE IS SOLID WALL
  206    GO TO (221,221,224,224,225,225) ITLIN(IMIN+1)

*———— SHOCK/SHOCK, DIFFERENT FAMILIES
  211    CALL SSDF (G,IMIN,NL,*300,*350,IPRINT)

*———— SHOCK/SHOCK, SAME FAMILIES
  212    CALL SSSF (G,IMIN,NL,*300,*350)

*———— SHOCK/EXP. WAVE, DIFF. FAMILIES, CASE A 213    CALL    SEDFA
(G,IMIN,NL,*300,*350,IPRINT,JUSE)

*———— SHOCK/EXP. WAVE, DIFF. FAMILIES, CASE B 214    CALL    SEDFB
(G,IMIN,NL,*300,*350,IPRINT,JUSE)

*———— SHOCK/EXP. WAVE, SAME FAMILIES, CASE A 215    CALL    SESF
(G,IMIN,NL,*350)

*———— SHOCK/EXP. WAVE, SAME FAMILIES, CASE B 216    CALL    SESF
(G,IMIN,NL,*350)

*———— EXP. WAVE/EXP. WAVE, DIFF. FAMILIES 217   CALL    EEDF
(G,IMIN,NL,*300,*350,IPRINT)

*———— EXP. WAVE/EXP. WAVE, SAME FAMILIES
  218    CALL EESF (G,IMIN,NL,*380)

*———— SHOCK REFLECTION AT SLIP LINE, CASE A 219    CALL    SRSPA
(G,IMIN,NL,*300,*350)

*———— SHOCK REFLECTION AT SLIP LINE, CASE B 220    CALL    SRSPB
(G,IMIN,NL,*300,*350)

*———— SHOCK REFLECTION AT SOLID WALL
  221    CALL SRW (G,IMIN,NL,*350)

*———— EXP. WAVE REFLECTION AT SLIP LINE, CASE A 222 CALL    ERSPA
(G,IMIN,NL,*300,*350)
```

```
*————— EXP. WAVE REFLECTION AT SLIP LINE, CASE B 223 CALL ERSPB
(G,IMIN,NL,*300,*350)

*————— EXP. WAVE REFLECTION AT SOLID WALL 224 CALL ERW
(G,IMIN,NL,*350)

*————— IMPOSSIBLE COMBINATION!
  225     WRITE (6,*) 'ERROR !'
*   WRITE (19,*) 'UPPER LINE IS NO.',IMIN
*   WRITE (19,*) 'TYPE OF UPPER LINE:',ITLIN(IMIN)
*   WRITE (19,*) 'TYPE OF LOWER LINE:',ITLIN(IMIN+1)
*   WRITE (19,*) 'PLEASE CHECK AGAIN!'
    GOTO 3333

*——————————————————————————————————* FINAL
SETUP OF THE NEW AREA - ONE LINE/FIELD IS ADDED
*——————————————————————————————————300 CONTINUE

*——— LINE-PROPERTIES
    DO 310 I=IMIN+3,NL
          TH(I)   = THOLD(I-1)
          ITLIN(I) = ITLOLD(I-1)
  310     CONTINUE

*——— FIELD-PROPERTIES
    DO 311 I=IMIN+2,NL-1
          AM(I)  = AMOLD(I-1)
          P(I)   = POLD(I-1)
          PT(I)  = PTOLD(I-1)
          ALF(I) = ALFOLD(I-1)
  311     CONTINUE

*——— VERTICAL LOCATIONS OF LINES AT NEW INTERSECTION
    DO 320 I = 1,IMIN-1
          YL(I) = TAN(TH(I))*(X0-X0OLD)+YLOLD(I)
  320     CONTINUE
YINT = TAN(THOLD(IMIN))*(X0-X0OLD)+YLOLD(IMIN)
    DO 321 I=IMIN,IMIN+2
          YL(I) = YINT
  321     CONTINUE
    DO 322 I=IMIN+3,NL
YL(I) = TAN(THOLD(I-1))*(X0-X0OLD)+YLOLD(I-1)
  322     CONTINUE

    GO TO 400

*——————————————————————————————————* FI-
NAL SETUP OF THE NEW AREA - TWO LINES/FIELDS ARE ADDED
*——————————————————————————————————330 CON-
TINUE

*——— LINE-PROPERTIES
    DO 331 I = IMIN+3,NL
          TH(I)   = THOLD(I-2)
```

```
          ITLIN(I) = ITLOLD(I-2)
   331    CONTINUE

*------- FIELD-PROPERTIES
    DO 332 I = IMIN+2,NL-1
          AM(I)  = AMOLD(I-2)
          P(I)   = POLD(I-2)
          PT(I)  = PTOLD(I-2)
          ALF(I) = ALFOLD(I-2)
   332    CONTINUE

*------- VERTICAL LOCATIONS OF LINES AT NEW INTERSECTION DO 333 I = 1,IMIN-
1
          YL(I) = TAN(TH(I))*(X0-X0OLD)+YLOLD(I)
   333    CONTINUE
YINT = TAN(THOLD(IMIN))*(X0-X0OLD)+YLOLD(IMIN)
    DO 334 I = IMIN,IMIN+2
          YL(I) = YINT
   334    CONTINUE
    DO 335 I = IMIN+3,NL
YL(I) = TAN(THOLD(I-2))*(X0-X0OLD)+YLOLD(I-2)
   335    CONTINUE

    GO TO 400

*------------------------------------------------------------
*    FINAL SETUP OF THE NEW AREA - NO LINE/FIELD IS ADDED
*------------------------------------------------------------

   350    CONTINUE

*------- VERTICAL LOCATIONS OF LINES AT NEW INTERSECTION
    DO 360 I = 1,IMIN-1
          YL(I) = TAN(TH(I))*(X0-X0OLD)+YLOLD(I)
   360    CONTINUE
YINT = TAN(THOLD(IMIN))*(X0-X0OLD)+YLOLD(IMIN)
    DO 361 I = IMIN,IMIN+1
          YL(I) = YINT
   361    CONTINUE
    DO 362 I = IMIN+2,NL
          YL(I) = TAN(THOLD(I))*(X0-X0OLD)+YLOLD(I)
   362    CONTINUE

    GO TO 400

*------------------------------------------------------------------*     FI-
NAL SETUP OF THE NEW AREA - ONE LINE/FIELD IS SUBTRACTED
*------------------------------------------------------------------380   CON-
TINUE

*------- LINE-PROPERTIES
    DO 385 I = IMIN+1,NL
          TH(I)    = THOLD(I+1)
          ITLIN(I) = ITLOLD(I+1)
```

```
 385     CONTINUE

*——— FIELD-PROPERTIES
   DO 386 I = IMIN,NL-1
        AM(I)  = AMOLD(I+1)
        P(I)   = POLD(I+1)
        PT(I)  = PTOLD(I+1)
        ALF(I) = ALFOLD(I+1)
 386    CONTINUE

*——— VERTICAL LOCATIONS OF LINES AT NEW INTERSECTION DO 390 I = 1,IMIN-
1
YL(I) = TAN(TH(I))*(X0-X0OLD)+YLOLD(I)
 390    CONTINUE
YINT = TAN(THOLD(IMIN))*(X0-X0OLD)+YLOLD(IMIN) YL(IMIN) = YINT
   DO 391 I = IMIN+1,NL
   YL(I) = TAN(THOLD(I+1))*(X0-X0OLD)+YLOLD(I+1) 391 CONTINUE

 400    CONTINUE
   LOUT = .TRUE.
*______________________________
*   PRINTOUT - STUFF
*______________________________
*        IF (LOUT) WRITE (19,913)
 401     IF (LOUT) THEN
         DO 450 I = 1,NL
             THDG = TH(I)*180./PI
             ALDG = ALF(I)*180./PI
*             WRITE (19,914) I,YL(I),THDG,ITLIN(I)
             IF (I.EQ.NL) GO TO 450
*             WRITE (19,915) I,AM(I),ALDG,P(I),PT(I)
 450         CONTINUE
        END IF

*——— WRITE TO BODY SURFACE DATA FILE IF FIRST STREAM TUBE IF(ISTR.EQ.1)
THEN
        ASTRUT =    0.
        FLX   = 0.
        FMM   =0.
        SQSTF  = SQRT(G/(RG*T0INF))

        DO 460 I = 1,NL-1
           IF (PT(I).NE.0.) THEN
              DY = YL(I) - YL(I+1)
              F1 = (1. + .5*GM*AM(I)**2)**(.5*GP/GM)
DFLX  = PT(I)*DY*H1*DZ*AM(I)*COS(ALF(I))*SQSTF/F1 FLX        = FLX + DFLX
              TR  = 1. + .5*GM*AM(I)**2
              RHO    = PT(I)/(RG*T0INF*TR**(1./GM))
              U2  = G*RG*T0INF*(AM(I)*COS(ALF(I)))**2/TR
DFMM  = (P(I) + RHO*U2)*DY*H1*DZ
              FMM    = FMM + DFMM
           ELSE
              ASTRUT = ASTRUT+(YL(I)-YL(I+1))
           END IF
```

```
 460        CONTINUE
          WIDTH = YL(1) - YL(NL) - ASTRUT
          AREAO = WIDTH*H1*DZ

*——— NOW GET AVG'D MASS & MOMENTUM FLUXES FROM CONT EQN, etc. R1  =
FLX/AREAO
          R2  = FMM/AREAO
          R3  = H0INF
          AA  = .5*GP/GM
          BB  = R2/GM
CC  = R3*R1**2 - .5*R2**2
DSC = BB**2 - 4.*AA*CC
          IF(DSC.LT.0.) THEN
*         WRITE(6,*),'  Uh-oh....DSC < 0...'
*         WRITE(6,*),'  AA = ',AA
*         WRITE(6,*),'  BB = ',BB
*         WRITE(6,*),'  CC = ',CC
*         WRITE(6,*),'  DSC = ',DSC
*         WRITE(6,*),'  R1 = ',R1
*         WRITE(6,*),'  R2 = ',R2
*         WRITE(6,*),'  R3 = ',R3
             GOTO 3333
          ENDIF
*——— Note: This PAV is for M>1 as is appropriate here.
*———         If M<1 at this point though, use PAV = .5*(BB + SQRT(DSC))/AA
          PAV = .5*(BB - SQRT(DSC))/AA
          UAV = (R2 - PAV)/R1
          RHAV = R1/UAV
          AMAV = UAV/SQRT(G*PAV/RHAV)
*          WRITE(55,*) X0,AMAV,PAV,RHAV

    ENDIF

*———————————
*    END OF LOOP
*———————————
IF (NAREA.EQ.500) THEN
WRITE (6,*) 'NUMBER OF ALLOWABLE AREAS EXCEEDED!' GOTO 3333
    END IF
*    WRITE (2,*) NL,X0
    DO 499 J = 1,NL
*          WRITE (2,*) YL(J),TH(J),ITLIN(J)
   499     CONTINUE
    IF (JLOOP) 100,500,500

   500     CONTINUE

*——————————————
*    FINAL PT-DISTRIBUTION
*——————————————
*    WRITE (19,930)
*    WRITE (19,931) YL(1),YL(NL)
    DO 550 j = 1,NL-1
*          WRITE (19,932) YL(j),PT(j)
```

```
 550     CONTINUE
*——— AVERAGE PT
   ASTRUT =   0.
   FLX   =   0.
   FMM   =  0.
   SQSTF  = SQRT(G/(RG*T0INF))

   DO 560 I = 1,NL-1
         IF (PT(I).NE.0.) THEN
             DY= YL(I) - YL(I+1)
             F1 = (1. + .5*GM*AM(I)**2)**(.5*GP/GM)
DFLX  = PT(I)*DY*H1*DZ*AM(I)*COS(ALF(I))*SQSTF/F1 FLX       = FLX + DFLX
             TR = 1. + .5*GM*AM(I)**2
             RHO   = PT(I)/(RG*T0INF*TR**(1./GM))
             U2 = G*RG*T0INF*(AM(I)*COS(ALF(I)))**2/TR
             DFMM  = (P(I) + RHO*U2)*DY*H1*DZ
             FMM   = FMM + DFMM
         ELSE
             ASTRUT = ASTRUT+(YL(I)-YL(I+1))
         END IF
  560    CONTINUE
   WIDTH = YL(1) - YL(NL) - ASTRUT
   AREAO = WIDTH*H1*DZ
*              WRITE(6,*), WIDTH,H1,DZ
*    WRITE(6,*),'  Streamtube area at end of inlet = ',AREAO
*    WRITE(6,*),'  Mass Flux at end of inlet = ',FLX
*    WRITE(6,*),'  Momentum flux at end of inlet = ',FMM
*    WRITE(6,*),'     P(3) = ',P(3)

*——— NOW GET AVG'D MASS & MOMENTUM FLUXES FROM CONT EQN, etc. R1 =
FLX/AREAO
   R2  = FMM/AREAO
   R3  = H0INF
   AA  = .5*GP/GM
   BB  = R2/GM
CC  = R3*R1**2 - .5*R2**2
DSC = BB**2 - 4.*AA*CC IF(DSC.LT.0.) THEN
*  WRITE(6,*),'  Uh-oh....DSC < 0...'
*  WRITE(6,*),'  AA = ',AA
*  WRITE(6,*),'  BB = ',BB
*  WRITE(6,*),'  CC = ',CC
*  WRITE(6,*),' DSC = ',DSC
*     WRITE(6,*),'  R1 = ',R1
*     WRITE(6,*),'  R2 = ',R2
*     WRITE(6,*),'  R3 = ',R3
          GOTO 3333
     ENDIF
*——— Note: This PAV is for M>1 as is appropriate here.
*———          If M<1 at this point though, use PAV = .5*(BB + SQRT(DSC))/AA
     PAV = .5*(BB - SQRT(DSC))/AA
     UAV = (R2 - PAV)/R1
     RHAV = R1/UAV
     AMAV = UAV/SQRT(G*PAV/RHAV)
```

```
        PTAV = PAV*(1. + .5*GM*AMAV**2)**(G/GM)
*          WRITE (19,933) PTAV,AMAV
*          WRITE (6,*) ,'check point 560',PTAV,AMAV

*------- FORCES
*          WRITE (19,*) 'FORCE ON UPPER WEDGE:  ',DFUW
*          WRITE (19,*) 'FORCE ON LOWER WEDGE:  ',DFLW
*          WRITE (19,*) 'FORCE ON UPPER STRUT (TOP):  ',FST1
*          WRITE (19,*) 'FORCE ON UPPER STRUT (BOTTOM):  ',FST2
*          WRITE (19,*) 'FORCE ON MIDDLE STRUT (TOP):  ',FST3
*          WRITE (19,*) 'FORCE ON MIDDLE STRUT (BOTTOM):  ',FST4
*          WRITE (19,*) 'FORCE ON LOWER STRUT (TOP):  ',FST5
*          WRITE (19,*) 'FORCE ON LOWER STRUT (BOTTOM):  ',FST6

*          CLOSE (2,STATUS='KEEP')
*          CLOSE (19,STATUS='KEEP')
 3333
        RETURN
        END
*
*
        SUBROUTINE UPEXP (G,IMIN,NL,IG,*,*) *------------------------------
------------------------------*         SUBROUTINE 'UPEXP' CALCULATES
THE FLOW CONDITIONS BEHIND AN
*   EXPANSION WHICH TAKES PLACE ABOVE A CONTOUR (LOWER WEDGE, UPPER
*   PART OF A STRUT)
*
*   REGION 1: 'UNDISTURBED' FLOW
*   REGION 2: BETWEEN THE FORWARD AND REARWARD CREATED EXPANSION
WAVE
*   REGION 3: BEHIND THE REARWARD EXPANSION WAVE
* *--------------------------------------------------------------------
PARAMETER (NGMAX=23,NLMAX=201,NFMAX=200,NST=5)

        DIMENSION JSTRUT(NGMAX)
DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
   1            ALF(NFMAX),ITLIN(NLMAX)
   DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX), 1
        PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

        LOGICAL LPM,LOUT

COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
   1            THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT
COMMON/GEO/  H,H1,XW1,XW2,D1,D1ST,D2,D2ST,XD(NST),YD(NST),
   1            DT(NST),DD(NST),XCC,DELST(NST),JSTRUT,ALST

*-------------------------------------------------------------------990
FORMAT (38HCHANGE OF LOWER WEDGE (WEDGE 1) AT X= ,F7.3)
   991 FORMAT (21HUPPER CHANGE OF STRUT,I2,7H AT X= ,F7.3)

        PI = 3.141592654
```

```
*________________________________
*    ANALYZE THE GEOMETRIC EVENT
*________________________________KST = JSTRUT(IG-1)
     IF (KST.EQ.0) THEN
*           IF (LOUT) WRITE (19,990) X0
            D    = D1-D1ST
            IMIN = NL
     ELSE
*           IF (LOUT) WRITE (19,991) KST,X0
D    = 2.*ATAN2(DD(KST),DT(KST))
SSTR = SQRT(DD(KST)**2+DT(KST)**2)
PHI = DELST(KST)+ATAN2(DD(KST),DT(KST)) YSTR = SSTR*SIN(PHI)+YD(KST)+.000001
DO 70 I = 1,NL
YCHECK = TAN(TH(I))*(X0-X0OLD)+YL(I) IF (YCHECK.GT.YSTR) THEN
                  IMIN = I+1
               ELSE
                  GO TO 75
               END IF
   70          CONTINUE
   75          CONTINUE
     END IF

*________________________________
*    CALCULATE THE NEW FIELDS
*________________________________
     AM1  = AM(IMIN-1)
     ALF1 = ALF(IMIN-1)
CALL PM (AM1,D,G,AM3,TH3FW,TH3RW,TH3,P3) D    = D/2.
     CALL PM (AM1,D,G,AM2,TH2FW,TH2RW,TH2,P2)

*________________________________
*    ASSIGN NEW LINES/FIELDS
*________________________________
     NL = NL+2
     IF (KST.NE.0) GO TO 200

*—— CHANGE OF LOWER WEDGE ———
*——— UPPER EXPANSION WAVE
     TH(NL-2)    = TH3FW+ALF1
     ITLIN(NL-2) = 4

*——— LOWER EXPANSION WAVE
     TH(NL-1) = TH3RW+ALF1
     ITLIN(NL-1) = 4

*——— SOLID WALL
     TH(NL) = D1ST
     ITLIN(NL) = 6

*——— REGION 2
     AM(NL-2) = AM2
     P(NL-2)  = P2*POLD(NL-3)
     PT(NL-2) = PTOLD(NL-3)
ALF(NL-2)= D1-((D1-D1ST)/2.)
```

```
*——— REGION 3
    AM(NL-1) = AM3
    P(NL-1)  = P3*POLD(NL-3)
    PT(NL-1) = PTOLD(NL-3)
    ALF(NL-1)= D1ST

*——— VERTICAL LOCATION OF LINES AT GEOMETRIC EVENT DO 100 I = 1,NL-3
      YL(I) = TAN(THOLD(I))*(X0-X0OLD)+YLOLD(I) 100   CONTINUE
YW1 = TAN(THOLD(NL-2))*(X0-X0OLD)+YLOLD(NL-2) YL(NL-2) = YW1
      YL(NL-1) = YW1
      YL(NL)   = YW1

      RETURN 1

*—— CHANGE OF STRUT ———
  200 CONTINUE

*——— UPPER EXPANSION WAVE
      TH(IMIN) = TH3FW+ALF1
      ITLIN(IMIN) = 4

*——— LOWER EXPANSION WAVE
      TH(IMIN+1) = TH3RW+ALF1
      ITLIN(IMIN+1) = 4

*——— SOLID WALL
TH(IMIN+2) = -ATAN2(DD(KST),DT(KST))+DELST(KST) ITLIN(IMIN+2) = 6

*——— REGION 2
      AM(IMIN) = AM2
      P(IMIN)  = P2*POLD(IMIN-1)
      PT(IMIN) = PTOLD(IMIN-1)
      ALF(IMIN)= ALF1-D

*——— REGION 3
      AM(IMIN+1) = AM3
      P(IMIN+1)  = P3*POLD(IMIN-1)
      PT(IMIN+1) = PTOLD(IMIN-1)
      ALF(IMIN+1)= -ATAN2(DD(KST),DT(KST))+DELST(KST)

      RETURN 2
      END
*
*
      SUBROUTINE LOWEXP (G,IMIN,NL,IG,*,*) *————————————————
————————————————————————* SUBROUTINE 'LOWEXP' CALCU-
LATES THE FLOW CONDITIONS BEHIND AN
*   EXPANSION WHICH TAKES PLACE BELOW A CONTOUR (UPPER WEDGE, LOWER
*   PART OF STRUT)
*———————————————————————————————————————*
*   REGION 1: 'UNDISTURBED' FLOW
*   REGION 2: BETWEEN THE FORWARD AND REARWARD CREATED EXPANSION
WAVE
*   REGION 3: BEHIND THE REARWARD EXPANSION WAVE
```

```
* *______________________________________________________________
PARAMETER (NGMAX=23,NLMAX=201,NFMAX=200,NST=5)

      DIMENSION JSTRUT(NGMAX)
DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
   1          ALF(NFMAX),ITLIN(NLMAX)
   DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX), 1
      PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

      LOGICAL LPM,LOUT

COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
   1            THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT
COMMON/GEO/  H,H1,XW1,XW2,D1,D1ST,D2,D2ST,XD(NST),YD(NST),
   1            DT(NST),DD(NST),XCC,DELST(NST),JSTRUT,ALST

*______________________________________________________________990
FORMAT (38HCHANGE OF UPPER WEDGE (WEDGE 2) AT X= ,F7.3)
   991     FORMAT (21HLOWER CHANGE OF STRUT,I2,7H AT X= ,F7.3)

     PI =   3.141592654

*______________________________
*    ANALYZE THE GEOMETRIC EVENT
*______________________________
     KST = JSTRUT(IG-1)
     IF (KST.EQ.0) THEN
*          IF (LOUT) WRITE (19,990) X0
           D    = -D2+D2ST
           IMIN = 1
     ELSE
*          IF (LOUT) WRITE (9,991) KST,X0
D    = 2.*ATAN2(DD(KST),DT(KST))
SSTR = SQRT(DD(KST)**2+DT(KST)**2)
PHI      =      -DELST(KST)+ATAN2(DD(KST),DT(KST))      YSTR     =    -
SSTR*SIN(PHI)+YD(KST)+.000001 DO 30 I = 1,NL
YCHECK = TAN(TH(I))*(X0-X0OLD)+YL(I) IF (YCHECK.GT.YSTR) THEN
                IMIN = I+1
             ELSE
                GO TO 50
             END IF
   30        CONTINUE
   50        CONTINUE
    END IF

*___________________________
*   CALCULATE THE NEW FIELDS
*___________________________
    AM1  = AM(IMIN)
    ALF1 = ALF(IMIN)
CALL PM (AM1,D,G,AM3,TH3FW,TH3RW,TH3,P3) TH3FW = -TH3FW
    TH3RW = -TH3RW
    D           = D/2.
    CALL PM (AM1,D,G,AM2,TH2FW,TH2RW,TH2,P2)
```

```
*________________________________
*   ASSIGN NEW LINES/FIELDS
*________________________________
      NL = NL+2
      IF (KST.NE.0) GO TO 200

*—— CHANGE OF UPPER WEDGE
*——— SOLID WALL
      TH(1) = D2ST
      ITLIN(1) = 6

*——— UPPER EXPANSION WAVE
      TH(2) = TH3RW+ALF1
      ITLIN(2) = 3

*——— LOWER EXPANSION WAVE
      TH(3) = TH3FW+ALF1
      ITLIN(3) = 3

*——— REGION 2
      AM(2) = AM2
      P(2)  = P2*POLD(1)
      PT(2) = PTOLD(1)
ALF(2)= D2+((-D2+D2ST)/2.)

*——— REGION 3
     AM(1) = AM3
     P(1)  = P3*POLD(1)
     PT(1) = PTOLD(1)
     ALF(1)= D2ST

*——— LINE-PROPERTIES
     DO 110 I=4,NL
TH(I)   = THOLD(I-2) ITLIN(I) = ITLOLD(I-2)
   110      CONTINUE

*——— FIELD-PROPERTIES
     DO 111 I=3,NL-1
            AM(I)  = AMOLD(I-2)
            P(I)   = POLD(I-2)
            PT(I)  = PTOLD(I-2)
            ALF(I) = ALFOLD(I-2)
   111      CONTINUE

*——— VERTICAL LOCATION OF LINES AT GEOMETRIC EVENT YW2 =
TAN(THOLD(1))*(X0-X0OLD)+YLOLD(1) YL(1) = YW2
     YL(2) = YW2
     YL(3) = YW2

     DO 120 I=4,NL
     YL(I) = TAN(THOLD(I-2))*(X0-X0OLD)+YLOLD(I-2) 120  CONTINUE

     RETURN 1
```

```
*—— CHANGE OF STRUT ——
  200     CONTINUE

*——— SOLID WALL
TH(IMIN) = DELST(KST)+ATAN2(DD(KST),DT(KST)) ITLIN(IMIN) = 6

*——— UPPER EXPANSION WAVE
    TH(IMIN+1) = TH3RW+ALF1
    ITLIN(IMIN+1) = 3

*——— LOWER EXPANSION WAVE
    TH(IMIN+2) = TH3FW+ALF1
    ITLIN(IMIN+2) = 3

*——— REGION 2
    AM(IMIN+1) = AM2
    P(IMIN+1)  = P2*POLD(IMIN)
    PT(IMIN+1) = PTOLD(IMIN)
    ALF(IMIN+1)= ALF1+D

*——— REGION 3
    AM(IMIN) = AM3
    P(IMIN)  = P3*POLD(IMIN)
    PT(IMIN) = PTOLD(IMIN)
    ALF(IMIN)= DELST(KST)+ATAN2(DD(KST),DT(KST))

    RETURN 2
    END
*
*
SUBROUTINE BSTRUT (G,IMIN,NL,IG,*)
*____________________________________________________________________*
 SUBROUTINE 'BSTRUT' CALCULATES THE FLOW CONDITIONS BEHIND THE
*   DISTURBANCES CAUSED BY THE BEGINNING OF THE STRUT
*____________________________________________________________________*
*   REGION 1: 'UNDISTURBED' FLOW
*   REGION 2: BEHIND UPPER CREATED WAVE
*   REGION 3: INTERIOR OF DIAMOND, DEAD ZONE
*   REGION 4: BEHIND LOWER CREATED WAVE
* *____________________________________________________________________
PARAMETER (NGMAX=23,NLMAX=201,NFMAX=200,NST=5)

      DIMENSION JSTRUT(NGMAX)
      DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
   1           ALF(NFMAX),ITLIN(NLMAX)
    DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX), 1
      PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

      LOGICAL LSHANG,LPM,LOUT

      COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
   1              THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT
      COMMON/GEO/  H,H1,XW1,XW2,D1,D1ST,D2,D2ST,XD(NST),YD(NST),
   1              DT(NST),DD(NST),XCC,DELST(NST),JSTRUT,ALST
```

```
*-----------------------------------------------------------------990
FORMAT (18HBEGINNING OF STRUT,I2,7H AT X= ,F7.3)

        PI = 3.141592654
        KST = JSTRUT(IG-1)

*       IF (LOUT) WRITE (19,990) KST,X0

*----------------------------------------------------*     DETERMINATION OF THE
AFFECTED FLOWFIELD
*----------------------------------------------------DO 100 I = 1,NL
YCHECK = TAN(TH(I))*(X0-X0OLD)+YL(I) IF (YCHECK.GT.YD(KST)) THEN
                IMIN = I
            ELSE
                GO TO 150
            END IF
   100 CONTINUE

*-------------------------------
*  CALCULATE THE NEW FIELD
*-------------------------------
   150 CONTINUE
        AM1 = AM(IMIN)
        ALF1 = ALF(IMIN)
DU = ATAN2(DD(KST),DT(KST))+DELST(KST) DL = ATAN2(DD(KST),DT(KST))-
DELST(KST) LSHANG = .TRUE.

*------- REGION 2
        D = DU-ALF1
        IF (D.GT.0) THEN
            KU = 0
*           WRITE(9,*) '  SHANG loc 3...D = ',180.*D/PI,' AM1 = ',AM1
            CALL SHANG (AM1,D,G,LSHANG,TH2,P2,PT2,AM2)
            ITLIN(IMIN+1) = 2
            ALF2 = DU
        ELSE
            KU = 1
            D = ABS(D)
CALL PM(AM1,D,G,AM5,TH5FW,TH5RW,TH5,P5) TH2 = TH5FW
            PT5 = 1.
            D = D/2.
CALL PM(AM1,D,G,AM2,TH2FW,TH2RW,TH2AV,P2) PT2 = 1.
            ITLIN(IMIN+1) = 4
            ITLIN(IMIN+1+KU) = 4
            ALF2 = ALF1-D
     END IF

*------- REGION 4
     D = DL+ALF1
     IF (D.GT.0) THEN
            KL = 0
*           WRITE(19,*) '  SHANG loc 4...D = ',180.*D/PI,' AM1 = ',AM1
CALL SHANG (AM1,D,G,LSHANG,TH4,P4,PT4,AM4) ITLIN(IMIN+4+KU) = 1
            ALF4 = -DL
```

```
    ELSE
          KL = 1
          D = ABS(D)
CALL PM(AM1,D,G,AM6,TH6FW,TH6RW,TH6,P6) TH4 = TH6FW
          PT6 = 1.
          D = D/2.
CALL PM(AM1,D,G,AM4,TH4FW,TH4RW,TH4AV,P4) PT4 = 1.
ITLIN(IMIN+3+KU+KL) = 3 ITLIN(IMIN+4+KU+KL) = 3
          ALF4 = ALF1+D
    END IF

*—— ASSIGNING THE NEW LINES AND FIELDS ——IF (KU.NE.0.OR.KL.NE.0) THEN
          NL = NL+5
    ELSE
          NL = NL+4
    END IF

*——— UPPER WAVE
    TH(IMIN+1) = TH2+ALF1
    IF (KU.EQ.0) GO TO 50

*——— SECOND UPPER WAVE (ONLY EXPANSION) TH(IMIN+1+KU) = ALF1+TH5RW
   50     CONTINUE

*——— UPPER BORDER OF DIAMOND TH(IMIN+2+KU) = DU ITLIN(IMIN+2+KU) = 6

*——— LOWER BORDER OF DIAMOND TH(IMIN+3+KU) = -DL ITLIN(IMIN+3+KU) = 6
    IF (KL.EQ.0) GO TO 60

*——— SECOND LOWER WAVE (ONLY EXPANSION) TH(IMIN+3+KU+KL) = -
TH6RW+ALF1
   60     CONTINUE

*——— LOWER WAVE
TH(IMIN+4+KU+KL) = -TH4+ALF1

*——— REGION 2
    AM(IMIN+1) = AM2
P(IMIN+1)  = P2*POLD(IMIN) PT(IMIN+1) = PT2*PTOLD(IMIN) ALF(IMIN+1)= ALF2
IF (KU.EQ.0) GO TO 70

*——— REGION 5 (ONLY EXPANSION) AM(IMIN+2) = AM5
P(IMIN+2)  = P5*POLD(IMIN) PT(IMIN+2) = PT5*PTOLD(IMIN) ALF(IMIN+2)= DU
   70     CONTINUE

*———REGION 3, DEAD ZONE AM(IMIN+2+KU)=0. P(IMIN+2+KU) =0. PT(IMIN+2+KU)
= 0. ALF(IMIN+2+KU)= 0.
IF (KL.EQ.0) GO TO 80

*——— REGION 6 (ONLY EXPANSION) AM(IMIN+3+KU) = AM6 P(IMIN+3+KU) =
P6*POLD(IMIN) PT(IMIN+3+KU) = PT6*PTOLD(IMIN) ALF(IMIN+3+KU)= -DL
   80     CONTINUE

*——— REGION 4
```

```
AM(IMIN+3+KU+KL) = AM4
P(IMIN+3+KU+KL)  = P4*POLD(IMIN) PT(IMIN+3+KU+KL) = PT4*PTOLD(IMIN)
ALF(IMIN+3+KU+KL)= ALF4

*________________________________________
*   FINAL SETUP OF THE NEW AREA
*______________________________*_____ LINE-PROPERTIES
    DO 310 I=IMIN+5+KU+KL,NL
TH(I)   = THOLD(I-4-KU-KL) ITLIN(I) = ITLOLD(I-4-KU-KL)
  310     CONTINUE

*_____ FIELD-PROPERTIES
    DO 311 I=IMIN+4+KU+KL,NL-1
          AM(I)  = AMOLD(I-4-KU-KL)
          P(I)   = POLD(I-4-KU-KL)
PT(I)  = PTOLD(I-4-KU-KL) ALF(I) = ALFOLD(I-4-KU-KL)
  311     CONTINUE

*_____ VERTICAL LOCATIONS OF LINES AT BEGINNING OF DIAMOND
    DO 320 I = 1,IMIN
          YL(I) = TAN(THOLD(I))*(X0-X0OLD)+YLOLD(I)
  320     CONTINUE
    DO 321 I=IMIN+1,IMIN+4+KU+KL
          YL(I) = YD(KST)
  321     CONTINUE
    DO 322 I=IMIN+5+KU+KL,NL
    YL(I) = TAN(THOLD(I-4-KU-KL))*(X0-X0OLD)+YLOLD(I-4-KU-KL) 322     CON-
TINUE

    RETURN 1
    END
*
*
    SUBROUTINE ESTRUT (G,IMIN,NL,IG,*,*)
*__________________________________________________________________________*
   SUBROUTINE 'ESTRUT' CALCULATES THE FLOW CONDITIONS BEHIND THE
*     END OF THE STRUTS
*
*     IN THIS CASE A FLOW PATTERNIS CREATED THAT IS EITHER SIMILAR TO
*     TO A SHOCK/SHOCK OR A SHOCK/EXP. WAVE INTERSECTION (DIFFERENT
*     FAMILIES) DEPENDING ON THE FLOW ANGLES BEHIND THE STRUT
* *__________________________________________________________________________
PARAMETER (NGMAX=23,NLMAX=201,NFMAX=200,NST=5)

      DIMENSION JSTRUT(NGMAX)
DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
   1          ALF(NFMAX),ITLIN(NLMAX)
DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),
   1          PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

      LOGICAL LOUT

COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
   1            THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT
```

```
COMMON/GEO/  H,H1,XW1,XW2,D1,D1ST,D2,D2ST,XD(NST),YD(NST),
   1            DT(NST),DD(NST),XCC,DELST(NST),JSTRUT,ALST

*--------------------------------------------------------------------990 FOR-
MAT (12HEND OF STRUT,I2,7H AT X= ,F7.3)

        PI = 3.141592654
        KST = JSTRUT(IG-1)

*         IF (LOUT) WRITE (19,990) KST,X0

*----------------------------------------------
*  DETERMINATION OF THE AFFECTED FLOWFIELD
*----------------------------------------------YSTE                    =
YD(KST)+2.*DT(KST)*SIN(DELST(KST))+.000001 DO 100 I=2,NL
YCHECK = TAN(TH(I))*(X0-X0OLD)+YL(I)
           IF (YCHECK.GT.YSTE) THEN
            IMIN = I+1
           ELSE
            GO TO 150
           END IF
   100 CONTINUE

*-------------------------------
*       CALCULATE THE NEW FIELDS
*-------------------------------
   150 CONTINUE
        IPRINT = -1
        CALL SSDF (G,IMIN,NL,*200,*210,IPRINT)

   200 CONTINUE
        IRESL = -1
        GO TO 250
   210 CONTINUE
        IRESL = 1

*------- CHECK IF NEW FLOW ANGLE IS GREATER THAN ADMISSIBLE *
              FOR A TWO SHOCK FLOW PATTERN
   250 CONTINUE
        DCRIT1 = ATAN2(DD(KST),DT(KST))+DELST(KST)
DCRIT2 = -ATAN2(DD(KST),DT(KST))+DELST(KST) IF (ALF(IMIN).GT.DCRIT1) THEN
          JUMP = -1
ELSE IF (ALF(IMIN).LT.(DCRIT2)) THEN
         JUMP = 1
    ELSE
         JUMP = 0
    END IF

    IF(JUMP) 300,600,300

*------- NO PURE SHOCK/SHOCK INTERSECTION, RESET DATA 300 CONTINUE
    IF (IRESL.NE.1) NL = NL-1
    DO 310 I = 1,NL
         TH(I) = THOLD(I)
```

```
        ITLIN(I) = ITLOLD(I)
  310 CONTINUE
   DO 311 I = 1,NL-1
        AM(I)  = AMOLD(I)
        P(I)   = POLD(I)
        PT(I)  = PTOLD(I)
        ALF(I) = ALFOLD(I)
  311 CONTINUE

   IF (JUMP) 400,600,500

*——— UPPER CREATED WAVE IS SHOCK BUT LOWER ONE IS EXPANSION 400
CONTINUE
   IPRINT = -1
   JUSE = 1
   CALL SEDFA (G,IMIN,NL,*700,*800,IPRINT,JUSE)

*——— LOWER CREATED WAVE IS SHOCK BUT UPPER ONE IS EXPANSION 500
CONTINUE
   IPRINT = -1
   JUSE = 1
   CALL SEDFB (G,IMIN,NL,*700,*800,IPRINT,JUSE)

*______________________________________________________________
*   FLOWFIELD IS DETERMINED, RETURN TO MAIN PROGRAM
*______________________________________________________________
  600 CONTINUE
   IF (IRESL) 700,700,800

  700 CONTINUE
   RETURN 1

  800 CONTINUE
   RETURN 2
   END
*
*
   SUBROUTINE SSDF (G,IMIN,NL,*,*,IPRINT) *______________________
_______________________________*     SUBROUTINE 'SSDF' CALCULATES THE
FLOW CONDITIONS BEHIND TWO
*  INTERSECTING ARBITRARY OBLIQUE SHOCKS OF DIFFERENT FAMILIES,
*  DERIVED FROM THE PREVIOUSLY DEVELOPPED SUBROUTINE 'INSECT'
*_____________________________________________________________*
*  REGION 1: BEHIND LOWER FIRST SHOCK
*  REGION 2: BEHIND UPPER FIRST SHOCK
*  REGION 3: BEHIND UPPER SECOND SHOCK
*  REGION 4: BEHIND LOWER SECOND SHOCK
*
*_____________________________________________________________

   PARAMETER (NLMAX=201,NFMAX=200)

DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
     1        ALF(NFMAX),ITLIN(NLMAX)
```

```
DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),
     1        PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

      LOGICAL LSHANG,LOUT

COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
     1          THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT

*------------------------------------------------------------------990FOR-
MAT (34HSHOCK/SHOCK, DIFF. FAMILIES AT X= ,F7.3)
  991 FORMAT (6HLINES ,I3,5H AND ,I3)
996 FORMAT (22HSLIP LINE IS NEGLECTED)

      IF (LOUT) THEN
*       IF (IPRINT) 2,1,1
    1
        IMINPL = IMIN+1

    2 IPRINT = 1
      END IF

      PI = 3.141592654

*----------------------------------
*    PROVIDE INITIAL VALUES
*----------------------------------
      AM1= AM(IMIN+1)
      AM2= AM(IMIN-1)
      ALF0 = ALF(IMIN)
      ALF1 = ALF(IMIN+1)
      ALF2 = ALF(IMIN-1)

*------- FIRST GUESS FOR THE NEW FLOW ANGLE OMEGA OMEGA3 = ALF2+(ALF1-
ALF0)
      OMEGA4 = ALF1+(ALF2-ALF0)
      OMEGA = (OMEGA3+OMEGA4)/2.

*-------CALCULATE ANGLES OF SECOND SHOCKS WITH GUESSED OMEGA LSHANG
= .FALSE.
      D3 = -ALF2+OMEGA
*      WRITE(19,*) '  In SSDF...IMIN = ',IMIN
*      WRITE(19,*) '  SHANG loc 5...D3 = ',180.*D3/PI,' AM2 = ',AM2
      CALL SHANG (AM2,D3,G,LSHANG,TH3,PZZ,PTZZ,AMZZ)

      D4 = OMEGA-ALF1
      D4 = ABS(D4)
*      WRITE(19,*) '  SHANG loc 6...D4 = ',180.*D4/PI,' AM1 = ',AM1
      CALL SHANG (AM1,D4,G,LSHANG,TH4,PZZ,PTZZ,AMZZ)

*------- ITERATE THE NEW SHOCK WAVE ANGLES TH3 AND TH4
*            (NEWTON'S METHOD)
      NTH = 1
      P1  = P(IMIN+1)
      P2  = P(IMIN-1)
```

```
   50 CONTINUE
      TH3OLD = TH3
      TH4OLD = TH4
      S3 = (SIN(TH3))**2
      S4 = (SIN(TH4))**2
      C3 = (COS(TH3))**2
      C4 = (COS(TH4))**2
      AMS3 = AM2**2*S3
      AMS4 = AM1**2*S4
      GP = G+1.
      GM = G-1.
   CT3 = 1./TAN(TH3)
   CT4 = 1./TAN(TH4)

*——— EVALUATE THE F-FUNCTION FF
   FF = S3-P1/P2*(AM1/AM2)**2*S4-(1.-P1/P2)*GM/(2*G*AM2**2)

*——— EVALUATE THE G-FUNCTION GF
GF1 = ATAN2(2.*CT3*(AMS3-1.),(2.+AM2**2*(GP-2*S3))) GF2 = ATAN2(2.*CT4*(AMS4-
1.),(2.+AM1**2*(GP-2*S4)))
   GF = GF1+ALF2+GF2-ALF1

*——— EVALUATE THE DERIVATIVES
   FTH3 = 2.*SIN(TH3)*COS(TH3)

   FTH4 = -2.*P1/P2*(AM1/AM2)**2*SIN(TH4)*COS(TH4)

GTH31 = (2.+AM2**2*(GP-2.*S3))*(2.*AM2**2*(2.*C3-1.)+2./S3) GTH32 =
8.*AM2**2*C3*(AMS3-1.)
   GTH33 = (2.+AM2**2*(GP-2.*S3))**2
   GTH34 = (2.*CT3*(AMS3-1.))**2
   GTH3 = (GTH31+GTH32)/(GTH33+GTH34)

GTH41 = (2.+AM1**2*(GP-2.*S4))*(2.*AM1**2*(2.*C4-1.)+2./S4) GTH42 =
8.*AM1**2*C4*(AMS4-1.)
   GTH43 = (2.+AM1**2*(GP-2.*S4))**2
   GTH44 = (2.*CT4*(AMS4-1.))**2
   GTH4 = (GTH41+GTH42)/(GTH43+GTH44)

DETI = 1./(FTH3*GTH4-FTH4*GTH3)

*——— NEW VALUES FOR TH3, TH4
TH3=DETI*(-GTH4*FF+FTH4*GF)+TH3OLD TH4=DETI*(GTH3*FF-FTH3*GF)+TH4OLD
IF (ABS(TH3-TH3OLD).LT..00001.AND.ABS(TH4-TH4OLD).LT..00001) THEN GO TO 100
   ELSE
         NTH = NTH+1
         IF (NTH.GT.10) THEN
*              WRITE (19,*) ' TOO MANY ITERATIONS'
              GOTO 3333
         END IF
         GO TO 50
   END IF

*——— NEW FLOW ANGLE OMEGA
```

```
   100 CONTINUE
    OMEGA3 = GF1+ALF2
    OMEGA4 = ALF1-GF2

*——— FLOW CONDITIONS IN REGIONS 3 AND 4 OMEGA = (OMEGA3+OMEGA4)/2.
    D3 = -ALF2+OMEGA
    D4 = ALF1-OMEGA
    LSHANG = .TRUE.
*    WRITE(19,*) '  SHANG loc 6...D3 = ',180.*D3/PI,' AM2 = ',AM2
*    WRITE(19,*) '               D4 = ',180.*D4/PI,' AM1 = ',AM1
    CALL SHANG (AM2,D3,G,LSHANG,TH3,P3,PT3,AM3)
CALL SHANG (AM1,D4,G,LSHANG,TH4,P4,PT4,AM4) TH4 = -TH4

*——— ASSIGNING THE NEW LINES AND FIELDS *——— CHECK IF SLIP LINE IS
NEGLECTABLE
    PT3 = PT3*PTOLD(IMIN-1)
    PT4 = PT4*PTOLD(IMIN+1)
IF (ABS(AM3-AM4).LE..010.AND.ABS((PT3-PT4)/PT3).LE..010) THEN ISLIP = 1
    ELSE
         ISLIP = -1
    END IF
    IF (ISLIP) 200,200,300

*——— REGULAR PROCEDURE, SLIP LINE IS ADDED 200 CONTINUE
    NL = NL+1

*——— UPPER SHOCK
    TH(IMIN) = TH3+ALF2
    ITLIN(IMIN) = 2

*——— SLIP LINE
    TH(IMIN+1) = OMEGA
    ITLIN(IMIN+1) = 5

*——— LOWER SHOCK
    TH(IMIN+2) = TH4+ALF1
    ITLIN(IMIN+2) = 1

*——— REGION 3
    AM(IMIN) = AM3
    P(IMIN)  = P3*POLD(IMIN-1)
    PT(IMIN) = PT3
    ALF(IMIN)= OMEGA

*——— REGION 4
    AM(IMIN+1) = AM4
P(IMIN+1)  = P4*POLD(IMIN+1) PT(IMIN+1) = PT4
    ALF(IMIN+1)= OMEGA

    RETURN 1

*————————————————————————————————————————————————————————————————*
  SMALL DIFFERENCES BETWEEN REGION 3 AND 4, SLIP LINE NEGLECTED
*————————————————————————————————————————————————————————————————————300 CON-
```

```
TINUE
*    IF (LOUT) WRITE (19,996)

*——— UPPER SHOCK
     TH(IMIN) = TH3+ALF2
     ITLIN(IMIN) = 2

*——— LOWER SHOCK
     TH(IMIN+1) = TH4+ALF(IMIN+1)
     ITLIN(IMIN+1) = 1

*——— REGION 3 = REGION 4
     AM(IMIN) = (AM3+AM4)/2.
     P(IMIN)  = P3*POLD(IMIN-1)
     PT(IMIN) = AMIN1(PT3,PT4)
     ALF(IMIN)= OMEGA
 3333
     RETURN 2
     END
*
*
     SUBROUTINE SSSF (G,IMIN,NL,*,*) *————————————————————————
————————————————————*          SUBROUTINE 'SSSF' CALCULATES THE FLOW CON-
DITIONS BEHIND TWO
*    INTERSECTING OBLIQUE SHOCKS OF THE SAME FAMILY
*
*    REGION 0  : ABOVE BOTH INCOMING SHOCKS
*        REGION 1  : IN BETWEEN THE INCOMING SHOCKS
*        REGION 2  : BELOW BOTH INCOMING SHOCKS
*        REGION 3,4: BEHIND THE RESULTING WAVES
* *————————————————————————————————————————————————————

        PARAMETER (NLMAX=201,NFMAX=200)

DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
    1       ALF(NFMAX),ITLIN(NLMAX)
    DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX), 1
PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

        LOGICAL LOUT

COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
    1          THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT

*———————————————————————————————————————————————————————990 FORMAT
(33HSHOCK/SHOCK, SAME FAMILIES AT X= ,F7.3)
   991 FORMAT (6HLINES ,I3,5H AND ,I3)

        IF (LOUT) THEN
*          WRITE (19,990) X0
          IMINPL = IMIN+1
*          WRITE (19,991) IMIN,IMINPL
        END IF
```

```
      PI = 3.141592654

      JUSE = 4
      IPRINT = -1

      GO TO (100,200) ITLIN(IMIN)

*——— SHOCKS OF FAMILY 1
  100 CONTINUE
      CALL SEDFB (G,IMIN,NL,*500,*600,IPRINT,JUSE)

*——— SHOCKS OF FAMILY 2
  200 CONTINUE
      CALL SEDFA (G,IMIN,NL,*500,*600,IPRINT,JUSE)

  500 CONTINUE
      RETURN 1

  600 CONTINUE
      RETURN 2
      END
*
*
      SUBROUTINE SEDFA (G,IMIN,NL,*,*,IPRINT,JUSE) *————————————
————————————————————————————*          SUBROUTINE 'SEDFA' CAL-
CULATES THE FLOW CONDITIONS BEHIND TWO
*     INTERSECTING SHOCK AND EXPANSION WAVES OF DIFFERENT FAMILIES,
*     CASE A: EXPANSION WAVE OF FAMILY 1, SHOCK OF FAMILY 2
*
*     NOTE:
*     THIS SUBROUTINE IS ALSO USED FOR THE DETERMINATION OF THE FLOW-
*     FIELD BEHIND THE REFLECTION OF A WAVE AT A SLIP LINE SINCE THE
*     RESULTING WAVE PATTERN IS IN THIS CASE SIMILAR.
*     IN THE SAME WAY IT IS USED TO ITERATE THE WAVE PATTERN BEHIND
*     INTERSECTING WAVES OF THE SAME FAMILIES.
*
*     REGION 1: BEHIND LOWER FIRST SHOCK
*     REGION 2: BEHIND UPPER FIRST EXPANSION WAVE
*     REGION 3: BEHIND UPPER SECOND SHOCK
*     REGION 4: BEHIND LOWER SECOND EXPANSION WAVE
* *————————————————————————————————————————————————————

      PARAMETER (NLMAX=201,NFMAX=200)

DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
     1      ALF(NFMAX),ITLIN(NLMAX)
   DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX), 1
PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

      LOGICAL LSHANG,LPM,LOUT

COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
     1          THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT
```

```
*------------------------------------------------------------------------990FORMAT
(46HSHOCK/EXPANSION, DIFF. FAMILIES, CASE A AT X= ,F7.3) 991 FORMAT (6HLINES
,I3,5H AND ,I3)
995 FORMAT (17HWAVE IS NEGLECTED)
996 FORMAT (22HSLIP LINE IS NEGLECTED)

      IF (LOUT) THEN
       IF (IPRINT) 2,1,1
    1
       IMINPL = IMIN+1

    2 IPRINT = 1
      END IF

      PI = 3.141592654

*--------------------------------
*    PROVIDE INITIAL VALUES
*--------------------------------

      ALF0 = ALF(IMIN)
      ALF1 = ALF(IMIN+1)
      ALF2 = ALF(IMIN-1)
      AM1= AM(IMIN+1)
      AM2= AM(IMIN-1)
      P1  = P(IMIN+1)
      P2  = P(IMIN-1)

*------- FIRST GUESS FOR THE NEW FLOW ANGLE OMEGA
      GO TO (20,21,21,22,23) JUSE

*------- INITIAL GUESS FOR SHOCK/EXP. WAVE INTERSECTION 20 CONTINUE
      OMEGA3 = (ALF2+(ALF1-ALF0))
      OMEGA4 = (ALF1+(ALF2-ALF0))
      OMEGA = (OMEGA3+OMEGA4)/2.
      IF (OMEGA.LT.0.) THEN
       OMEGA = .98*OMEGA
      ELSE
       OMEGA = 1.02*(OMEGA+1.E-5)
      END IF
      GO TO 25

*------- INITIAL GUESS FOR REFLECTION AT SLIP LINE 21 CONTINUE
      OMEGA3 = (ALF2+(ALF1-ALF0))
      OMEGA4 = (ALF1+(ALF2-ALF0))
      OMEGA = (OMEGA3+OMEGA4)/2.
      IF (OMEGA.LT.0.) THEN
       OMEGA = .98*OMEGA
    ELSE
       OMEGA = 1.02*(OMEGA+1.E-5)
    END IF
    GO TO 25

*------- INITIAL GUESS FOR TWO SHOCKS OF SAME FAMILY 22 CONTINUE
```

```
      IF (ALF1.LT.0.) THEN
          OMEGA = 0.98*ALF1
      ELSE
          OMEGA = 1.02*(ALF1+1.E-5)
      END IF
      GO TO 25

*——— INITIAL GUESS FOR TWO EXP. WAVES OF SAME FAMILY 23 CONTINUE
      IF (ALF2.LT.0.) THEN
          OMEGA = 0.98*ALF2
      ELSE
          OMEGA = 1.02*(ALF2+1.E-5)
      END IF

     25 CONTINUE

*——— CALCULATE ANGLE OF SECOND SHOCK WITH GUESSED OMEGA LSHANG =
.FALSE.
      D3 = OMEGA-ALF2
      CALL SHANG (AM2,D3,G,LSHANG,TH3,PZZ,PTZZ,AMZZ)

*—— CALCULATE MACH NUMBER BEHIND SECOND EXPANSION WITH GUESSED
OMEGA D4  = OMEGA-ALF1
      CALL PM (AM1,D4,G,AM4,TH1ZZ,TH2ZZ,TH3ZZ,PZZ)

*————————————————————————————————————————*
  ITERATE THE NEW SHOCK WAVE ANGLE TH3 AND MACH NUMBER AM4
*               (NEWTON'S METHOD)
*————————————————————————————————————————NTH = 1
      GP = G+1.
      GM = G-1.
      GPM = GP/GM
     50 CONTINUE
      TH3OLD = TH3
      AM4OLD = AM4
      S3 = (SIN(TH3))**2
      C3 = (COS(TH3))**2
      CT3 = 1./TAN(TH3)
      AMS3 = AM2**2*S3
      AMQ1 = AM1**2-1.
      AMQ4 = AM4**2-1.
      AMG1 =      1.+GM/2.*AM1**2
      AMG4 =      1.+GM/2.*AM4**2

*——— EVALUATE THE F-FUNCTION FF
      FF1 = (AMG4/AMG1)**(-G/GM)
      FF = (2.*G*AMS3-GM)/GP-P1/P2*FF1

*——— EVALUATE THE G-FUNCTION GF
GF1 = ATAN2(2.*CT3*(AMS3-1.),(2.+AM2**2*(GP-2.*S3)))
GF2 = SQRT(GPM)*ATAN(SQRT(1./GPM*AMQ4))-ATAN(SQRT(AMQ4)) GF3 =
SQRT(GPM)*ATAN(SQRT(1./GPM*AMQ1))-ATAN(SQRT(AMQ1)) GF = GF1+ALF2-
GF2+GF3-ALF1
```

```
*——— EVALUATE THE DERIVATIVES
    FTH3 =        4.*G/GP*AM2**2*SIN(TH3)*COS(TH3)
FAM4 = P1/P2*G*AM4*AMG1**(G/GM)*AMG4**(-G/GM-1.)

GTH31 = (2.+AM2**2*(GP-2.*S3))*(2.*AM2**2*(2.*C3-1.)+2./S3) GTH32 =
8.*AM2**2*C3*(AMS3-1.)
    GTH33 = (2.+AM2**2*(GP-2.*S3))**2
    GTH34 = (2.*CT3*(AMS3-1.))**2
    GTH3 = (GTH31+GTH32)/(GTH33+GTH34)

    GAM4 = 1./SQRT(AMQ4)*(1./AM4-AM4/(1.+1./GPM*AMQ4))

    DETI = 1./(FTH3*GAM4-FAM4*GTH3)

*——— NEW VALUES FOR TH3, AM4
    TH3 = DETI*(-GAM4*FF+FAM4*GF)+TH3OLD
    AM4 = DETI*(GTH3*FF-FTH3*GF)+AM4OLD

IF (ABS(TH3-TH3OLD).LT..00001.AND.ABS(AM4-AM4OLD).LT..0001) THEN GO TO 100
    ELSE
            NTH = NTH+1
            IF (NTH.GT.10) THEN
                WRITE (6,*) ' TOO MANY ITERATIONS'
                GOTO 3333
            END IF
            GO TO 50
    END IF

*——— NEW FLOW ANGLE OMEGA
   100 CONTINUE
    OMEGA3 = GF1+ALF2
    OMEGA4 = GF2-GF3+ALF1

*——— FLOW CONDITIONS IN REGIONS 3 AND 4 OMEGA = (OMEGA3+OMEGA4)/2.
    D3 = OMEGA-ALF2
    LSHANG = .TRUE.
    CALL SHANG (AM2,D3,G,LSHANG,TH3,P3,PT3,AM3)
    D4 = OMEGA-ALF1
    CALL PM (AM1,D4,G,AM4,TH4FW,TH4RW,TH4,P4)
    TH4 = -TH4

*—————————————————————————————————
*   ASSIGNING THE NEW LINES AND FIELDS
*—————————————————————————————————
*——— CHECK IF SLIP LINE OR REFLECTED WAVE IS NEGLECTABLE PT3 =
PT3*PTOLD(IMIN-1)
    PT4 = PTOLD(IMIN+1)
    PT1 = PTOLD(IMIN+1)
    PT2 = PTOLD(IMIN-1)
    P3 = P3*POLD(IMIN-1)
    P4 = P4*POLD(IMIN+1)
    DFM23 = AM2-AM3
    DFA23 = OMEGA-ALF2
    DFP23 = (P3-P2)/P2
```

```
      DFPT23 = (PT2-PT3)/PT2
      DFM14 = AM4-AM1
      DFA14 = OMEGA-ALF1
      DFP14 = (P1-P4)/P1

      GO TO (151,152,153,152,153) JUSE

*------ CHECK IF SLIP LINE IS NEGLECTABLE
  151 CONTINUE
IF (ABS(AM3-AM4).LE.0.010.AND.ABS((PT3-PT4)/PT3).LE.0.010) THEN JUMP = 4
      ELSE
          JUMP = 1
        END IF
        GO TO 160

*------ CHECK IF EXP. WAVE IS NEGLECTABLE
  152 CONTINUE
IF (DFM14.LE..010.AND.DFA14.LE..0035.AND.DFP14.LE..010) THEN JUMP = 3
        ELSE
          JUMP = 1
        END IF
        GO TO 160

*------ CHECK IF SHOCK IS NEGLECTABLE
  153 CONTINUE
  IF  (DFM23.LE..010.AND.DFA23.LE..0035.AND.DFP23.LE..010.AND.DFPT23.  1
LE..010) THEN
          JUMP = 2
        ELSE
          JUMP = 1
        END IF

  160 CONTINUE
        GO TO (200,300,400,500) JUMP

*-----------------------------------------*        REGULAR PROCEDURE, SLIP
LINE IS ADDED
*-----------------------------------------200 CONTINUE
        NL = NL+1

*------ UPPER SHOCK
        TH(IMIN) = TH3+ALF2
        ITLIN(IMIN) = 2

*------ SLIP LINE
        TH(IMIN+1) = OMEGA
        ITLIN(IMIN+1) = 5

*------ LOWER EXPANSION WAVE
        TH(IMIN+2) = TH4+ALF1
        ITLIN(IMIN+2) = 3

*------ REGION 3
        AM(IMIN) = AM3
```

```
       P(IMIN)  = P3
       PT(IMIN) = PT3
       ALF(IMIN)= OMEGA

*------REGION 4
       AM(IMIN+1) = AM4
       P(IMIN+1)  = P4
       PT(IMIN+1) = PTOLD(IMIN+1)
       ALF(IMIN+1)= OMEGA
 3333
       RETURN 1

*------------------------------------------------------------------------*
SMALL DIFFERENCES IN REGION 2 AND 3, REFLECTED SHOCK WAVE IS
*          NEGLECTED
*--------------------------------------------------------------------------300  CON-
TINUE

*       IF (LOUT) WRITE (19,995)

       D = ABS(ALF2-ALF1)
CALL PM (AM1,D,G,AM4,TH4FW,TH4RW,TH4,P4) TH4 = -TH4

*------ SLIP LINE
    TH(IMIN) = ALFOLD(IMIN-1)
    ITLIN(IMIN) = 5

*------ LOWER EXPANSION WAVE
    TH(IMIN+1) = TH4+ALF1
    ITLIN(IMIN+1) = 3

*------ REGION 4 (REGION 3 REMAINS UNCHANGED) AM(IMIN) = AM4
    P(IMIN)  = POLD(IMIN-1)
    PT(IMIN) = PTOLD(IMIN+1)
    ALF(IMIN)= ALFOLD(IMIN-1)

    RETURN 2

*------------------------------------------------------------------------*
 SMALL DIFFERENCES IN REGION 1 AND 4, REFLECTED EXP. WAVE IS
*          NEGLECTED
*--------------------------------------------------------------------------400  CON-
TINUE

*   IF (LOUT) WRITE (19,995)

    D = ABS(ALF1-ALF2)
    LSHANG = .TRUE.
    CALL SHANG (AM2,D,G,LSHANG,TH3,P3,PT3,AM3)

*------ UPPER SHOCK
    TH(IMIN) = TH3+ALF2
    ITLIN(IMIN) = 2
```

```
*——— SLIP LINE
   TH(IMIN+1) = ALFOLD(IMIN+1)
   ITLIN(IMIN+1) = 5

*——— REGION 3
   AM(IMIN) = AM3
   P(IMIN)  = POLD(IMIN+1)
   PT(IMIN) = PT3*PTOLD(IMIN-1)
   ALF(IMIN)= ALFOLD(IMIN+1)

   RETURN 2

*————————————————————————————————————————————————————————————*
  SMALL DIFFERENCES BETWEEN REGION 3 AND 4, SLIP LINE NEGLECTED
*————————————————————————————————————————————————————————————500 CON-
TINUE

*   IF (LOUT) WRITE (19,996)

*——— UPPER SHOCK
   TH(IMIN) = TH3+ALF2
   ITLIN(IMIN) = 2

*——— LOWER EXPANSION WAVE
   TH(IMIN+1) = TH4+ALF1
   ITLIN(IMIN+1) = 3

*——— REGION 3 = REGION 4
   AM(IMIN) = (AM3+AM4)/2.
   P(IMIN)  = P3
   PT(IMIN) = PTOLD(IMIN+1)
      ALF(IMIN)= OMEGA

      RETURN 2
      END
*
*
      SUBROUTINE SEDFB (G,IMIN,NL,*,*,IPRINT,JUSE) *————————————
————————————————————————*                          SUBROUTINE
'SEDFB' CALCULATES THE FLOW CONDITIONS BEHIND TWO
*     INTERSECTING SHOCK AND EXPANSION WAVES OF DIFFERENT FAMILIES,
*     CASE B: SHOCK OF FAMILY 1, EXPANSION WAVE OF FAMILY 2
*
*     NOTE:
*     THIS SUBROUTINE IS ALSO USED FOR THE DETERMINATION OF THE FLOW-
*     FIELD BEHIND THE REFLECTION OF A WAVE AT A SLIP LINE SINCE THE
*     RESULTING WAVE PATTERN IS IN THIS CASE SIMILAR
*     IN THE SAME WAY IT IS USED TO ITERATE THE WAVE PATTERN BEHIND
*     INTERSECTING WAVES OF THE SAME FAMILIES.
*
*     REGION 1: BEHIND LOWER FIRST EXPANSION WAVE
*     REGION 2: BEHIND UPPER FIRST SHOCK
*     REGION 3: BEHIND UPPER SECOND EXPANSION WAVE
*     REGION 4: BEHIND LOWER SECOND SHOCK
```

```
*
*___________________________________________________________________________

      PARAMETER (NLMAX=201,NFMAX=200)

DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
     1        ALF(NFMAX),ITLIN(NLMAX)
DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),
     1        PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

      LOGICAL LSHANG,LPM,LOUT

COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
     1          THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT

*___________________________________________________________________________990 FOR-
MAT (46HSHOCK/EXPANSION, DIFF. FAMILIES, CASE B AT X= ,F7.3) 991 FORMAT
(6HLINES ,I3,5H AND ,I3)
995 FORMAT (17HWAVE IS NEGLECTED)
996 FORMAT (22HSLIP LINE IS NEGLECTED)

      IF (LOUT) THEN
       IF (IPRINT) 2,1,1
  1
       IMINPL = IMIN+1

  2   IPRINT = 1
      END IF

      PI = 3.141592654

*_____________________________
*   PROVIDE INITIAL VALUES
*_____________________________

      ALF0 = ALF(IMIN)
      ALF1 = ALF(IMIN+1)
      ALF2 = ALF(IMIN-1)
      AM1= AM(IMIN+1)
      AM2= AM(IMIN-1)
      P1  = P(IMIN+1)
      P2  = P(IMIN-1)

*------ FIRST GUESS FOR THE NEW FLOW ANGLE OMEGA
    GO TO (20,21,21,22,23) JUSE

*------ INITIAL GUESS FOR SHOCK/EXP. WAVE INTERSECTION 20 CONTINUE
    OMEGA3 = (ALF2+(ALF1-ALF0))
    OMEGA4 = (ALF1+(ALF2-ALF0))
    OMEGA = (OMEGA3+OMEGA4)/2.
    IF (OMEGA.LT.0.) THEN
       OMEGA =   1.02*OMEGA
    ELSE
       OMEGA =   0.98*(OMEGA-1.E-5)
    END IF
```

```
      GO TO 25

*——— INITIAL GUESS FOR REFLECTION AT SLIP LINE 21 CONTINUE
      OMEGA3 = (ALF2+(ALF1-ALF0))
      OMEGA4 = (ALF1+(ALF2-ALF0))
      OMEGA = (OMEGA3+OMEGA4)/2.
      IF (OMEGA.LT.0.) THEN
          OMEGA =   1.02*OMEGA
      ELSE
          OMEGA =   0.98*(OMEGA-1.E-5)
      END IF
      GO TO 25

*——— INITIAL GUESS FOR TWO SHOCKS OF SAME FAMILY
   22 CONTINUE
      IF (ALF2.LE.0.) THEN
          OMEGA =   1.02*(ALF2-1.E-5)
      ELSE
          OMEGA =   0.98*ALF2
      END IF
      GO TO 25

*——— INITIAL GUESS FOR TWO EXP. WAVES OF SAME FAMILY
   23 CONTINUE
      IF (ALF1.LE.0.) THEN
          OMEGA =   1.02*(ALF1-1.E-5)
      ELSE
          OMEGA =   0.98*ALF1
      END IF

   25 CONTINUE

*—— CALCULATE MACH NUMBER BEHIND SECOND EXPANSION WITH GUESSED
OMEGA D3  = ABS(-OMEGA+ALF2)
      CALL PM (AM2,D3,G,AM3,TH1ZZ,TH2ZZ,TH4ZZ,PZZ)

*——— CALCULATE ANGLE OF SECOND SHOCK WITH GUESSED OMEGA
      LSHANG = .FALSE.
      D4 = ABS(-OMEGA+ALF1)
      CALL SHANG (AM1,D4,G,LSHANG,TH4,PZZ,PTZZ,AMZZ)


*————————————————————————————————————————
*     ITERATE THE NEW SHOCK WAVE ANGLE TH4 AND MACH NUMBER AM3
*           (NEWTON'S METHOD)
*————————————————————————————————————————NTH = 1
      GP = G+1.
      GM = G-1.
      GPM = GP/GM
   50 CONTINUE
      TH4OLD = TH4
      AM3OLD = AM3
      S4 = (SIN(TH4))**2
```

```
      C4 = (COS(TH4))**2
      CT4 = 1./TAN(TH4)
      AMS4 = AM1**2*S4
      AMQ2 = AM2**2-1.
      AMQ3 = AM3**2-1.
      AMG2 = 1.+GM/2.*AM2**2
      AMG3 = 1.+GM/2.*AM3**2

*——— EVALUATE THE F-FUNCTION FF FF1 = (AMG3/AMG2)**(-G/GM)
      FF = P1/P2*(2.*G*AMS4-GM)/GP-FF1

*——— EVALUATE THE G-FUNCTION GF
    GF1 = ATAN2(2.*CT4*(AMS4-1.),(2.+AM1**2*(GP-2*S4)))
    GF2 = SQRT(GPM)*ATAN(SQRT(1./GPM*AMQ3))-ATAN(SQRT(AMQ3)) GF3 =
    SQRT(GPM)*ATAN(SQRT(1./GPM*AMQ2))-ATAN(SQRT(AMQ2)) GF = -GF1+ALF1+GF2-
    GF3-ALF2

*——— EVALUATE THE DERIVATIVES
      FTH4 = P1/P2*4.*G/GP*AM1**2*SIN(TH4)*COS(TH4)
      FAM3 = G*AM3*AMG2**(G/GM)*AMG3**(-G/GM-1.)

    GTH41 = (2.+AM1**2*(GP-2.*S4))*(2.*AM1**2*(2.*C4-1.)+2./S4) GTH42 =
    8.*AM1**2*C4*(AMS4-1.)
      GTH43 = (2.+AM1**2*(GP-2.*S4))**2
      GTH44 = (2.*CT4*(AMS4-1.))**2
      GTH4 = -(GTH41+GTH42)/(GTH43+GTH44)

      GAM3 = 1./SQRT(AMQ3)*(-1./AM3+AM3/(1.+1./GPM*AMQ3))

      DETI = 1./(FAM3*GTH4-FTH4*GAM3)

*——— NEW VALUES FOR TH3, AM4
      TH4 = DETI*(GAM3*FF-FAM3*GF)+TH4OLD
      AM3 = DETI*(-GTH4*FF+FTH4*GF)+AM3OLD

    IF (ABS(TH4-TH4OLD).LT..00001.AND.ABS(AM3-AM3OLD).LT..0001) THEN GO TO 100
      ELSE
          NTH = NTH+1
          IF (NTH.GT.10) THEN
*            WRITE (19,*) ' TOO MANY ITERATIONS'
            GOTO 3333
          END IF
          GO TO 50
      END IF

*——— NEW FLOW ANGLE OMEGA
    100 CONTINUE
      OMEGA4 = -GF1+ALF1
      OMEGA3 = -GF2+GF3+ALF2

*——— FLOW CONDITIONS IN REGIONS 3 AND 4 OMEGA = (OMEGA3+OMEGA4)/2.
      D3 = -OMEGA+ALF2
    CALL PM (AM2,D3,G,AM3,TH3FW,TH3RW,TH3,P3) D4 = -OMEGA+ALF1
      LSHANG = .TRUE.
```

```
CALL SHANG (AM1,D4,G,LSHANG,TH4,P4,PT4,AM4) TH4 = -TH4

*--------------------------------------------------
*   ASSIGNING THE NEW LINES AND FIELDS
*--------------------------------------------------
*--------- CHECK IF SLIP LINE OR REFLECTED WAVE IS NEGLECTABLE PT3 =
PTOLD(IMIN-1)
      PT4 = PT4*PTOLD(IMIN+1)
      PT1 = PTOLD(IMIN+1)
      PT2 = PTOLD(IMIN-1)
      P3 = P3*POLD(IMIN-1)
      P4 = P4*POLD(IMIN+1)
      DFM23 = AM3-AM2
      DFA23 = ALF2-OMEGA
      DFP23 = (P2-P3)/P2
      DFM14 = AM1-AM4
      DFA14 = ALF1-OMEGA
      DFP14 = (P4-P1)/P1
      DFPT14 = (PT1-PT4)/PT1

      GO TO (151,152,153,152,153) JUSE

*-------- CHECK IF SLIP LINE IS NEGLECTABLE
  151 CONTINUE
IF (ABS(AM3-AM4).LE.0.010.AND.ABS((PT3-PT4)/PT3).LE.0.010) THEN JUMP = 4
      ELSE
        JUMP = 1
      END IF
      GO TO 160

*-------- CHECK IF EXP. WAVE IS NEGLECTABLE
  152 CONTINUE
IF (DFM23.LE..010.AND.DFA23.LE..0035.AND.DFP23.LE..015) THEN JUMP = 2
      ELSE
        JUMP = 1
      END IF
      GO TO 160

*-------- CHECK IF SHOCK IS NEGLECTABLE
  153 CONTINUE
  IF(DFM14.LE..010.AND.DFA14.LE..0035.AND.DFP14.LE..015.AND.DFPT14.1   LE..010)
THEN
        JUMP = 3
      ELSE
        JUMP = 1
      END IF

  160 CONTINUE
      GO TO (200,300,400,500) JUMP

*--------------------------------------------------*        REGULAR PROCEDURE, SLIP
LINE IS ADDED
*--------------------------------------------------200 CONTINUE
      NL = NL+1
```

```
*——— UPPER EXPANSION WAVE
      TH(IMIN) = TH3+ALF2
      ITLIN(IMIN) = 4

*——— SLIP LINE
      TH(IMIN+1) = OMEGA
      ITLIN(IMIN+1) = 5

*——— LOWER SHOCK
      TH(IMIN+2) = TH4+ALF1
      ITLIN(IMIN+2) = 1

*——— REGION 3
   AM(IMIN) = AM3
   P(IMIN)  = P3
PT(IMIN) = PTOLD(IMIN-1) ALF(IMIN)= OMEGA

*——— REGION 4
   AM(IMIN+1) = AM4
   P(IMIN+1)  = P4
   PT(IMIN+1) = PT4
   ALF(IMIN+1)= OMEGA
3333
   RETURN 1

*————————————————————————————————————————*
 SMALL DIFFERENCES IN REGION 2 AND 3, REFLECTED EXP. WAVE IS
*      NEGLECTED
*————————————————————————————————————————300 CON-
TINUE

*   IF (LOUT) WRITE (19,995)

   D = ABS(ALF2-ALF1)
   LSHANG = .TRUE.
   CALL SHANG (AM1,D,G,LSHANG,TH4,P4,PT4,AM4)
   TH4 = -TH4

*——— SLIP LINE
   TH(IMIN) = ALFOLD(IMIN-1)
   ITLIN(IMIN) = 5

*——— LOWER SHOCK
   TH(IMIN+1) = TH4+ALF1
   ITLIN(IMIN+1) = 1

*——— REGION 4 (REGION 3 REMAINS UNCHANGED) AM(IMIN) = AM4
   P(IMIN)  = POLD(IMIN-1)
   PT(IMIN) = PT4*PTOLD(IMIN+1)
   ALF(IMIN)= ALFOLD(IMIN-1)

   RETURN 2

*————————————————————————————————————————*
```

```
  SMALL DIFFERENCES IN REGION 1 AND 4, REFLECTED SHOCK WAVE IS
*       NEGLECTED
*-----------------------------------------------------------------400 CON-
TINUE

*   IF (LOUT) WRITE (19,995)

    D = ABS(ALF1-ALF2)
    CALL PM (AM2,D,G,AM3,TH3FW,TH3RW,TH3,P3)

*------ UPPER EXP. WAVE
    TH(IMIN) = TH3+ALF2
    ITLIN(IMIN) = 4

*------ SLIP LINE
    TH(IMIN+1) = ALFOLD(IMIN+1)
    ITLIN(IMIN+1) = 5

*------ REGION 3
    AM(IMIN) = AM3
    P(IMIN)  = POLD(IMIN+1)
    PT(IMIN) = PTOLD(IMIN-1)
    ALF(IMIN)= ALFOLD(IMIN+1)

       RETURN 2

*---------------------------------------------------------------*
SMALL DIFFERENCES BETWEEN REGION 3 AND 4, SLIP LINE NEGLECTED
*---------------------------------------------------------------500 CON-
TINUE

*       IF (LOUT) WRITE (19,996)

*------ UPPER EXPANSION WAVE
       TH(IMIN) = TH3+ALF2
       ITLIN(IMIN) = 4

*------ LOWER SHOCK
       TH(IMIN+1) = TH4+ALF1
       ITLIN(IMIN+1) = 1

*------ REGION 3 = REGION 4
       AM(IMIN) = (AM3+AM4)/2.
       P(IMIN)  = P3
       PT(IMIN) = PTOLD(IMIN-1)
       ALF(IMIN)= OMEGA

       RETURN 2
       END
*
*
       SUBROUTINE SESF (G,IMIN,NL,*) *-----------------------------------
---------------------------------*       SUBROUTINE 'SESF' CALCULATES THE FLOW
CONDITIONS BEHIND TWO
```

```
*      INTERSECTING OBLIQUE SHOCK- AND EXPANSION WAVES OF THE SAME
FAMILY
*
*      REGION 0  : ABOVE BOTH INCOMING WAVES
*      REGION 1  : IN BETWEEN THE INCOMING WAVES
*      REGION 2  : BELOW THE INCOMING WAVES
*      REGION 3,4: BEHIND THE RESULTING WAVES
* *------------------------------------------------------------------

       PARAMETER (NLMAX=201,NFMAX=200)

DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
   1       ALF(NFMAX),ITLIN(NLMAX)
DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),
   1       PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

       LOGICAL LSHANG,LPM,LOUT

COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
   1         THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT

*------------------------------------------------------------------990 FOR-
MAT (37HSHOCK/EXPANSION, SAME FAMILIES AT X= ,F7.3)
  991 FORMAT (6HLINES ,I3,5H AND ,I3)

       IF (LOUT) THEN
*        WRITE (19,990) X0
        IMINPL = IMIN+1
*        WRITE (19,991) IMIN,IMINPL
       END IF

       PI = 3.141592654

       GO TO (100,200,100,200) ITLIN(IMIN)

*--------------------------
*    WAVES OF FAMILY I
*--------------------------
  100 CONTINUE

   AM0 = AM(IMIN+1)
   ALF0 = ALF(IMIN+1)
   ALF2 = ALF(IMIN-1)

*------ CALCULATE NEW FIELD
   IF ((ALF2-ALF0).LT.0.) THEN
        D = ABS(ALF2-ALF0)
        LSHANG = .TRUE.
  CALL SHANG (AM0,D,G,LSHANG,TH3,P3,PT3,AM3) ELSE
        D = ALF2-ALF0
CALL PM (AM0,D,G,AM3,TH3FW,TH3RW,TH3,P3) PT3 = 1.
   END IF

*---------------------------------------------*    ASSIGNING THE NEW LINES
```

```
AND FIELDS
*--------------------------------------------------*------ SLIP LINE
    TH(IMIN) = ALF2
    ITLIN(IMIN) = 5

*------ RESULTING WAVE
    IF ((ALF2-ALF0).LT.0.) THEN
        TH(IMIN+1) = -TH3+ALF0
        ITLIN(IMIN+1) = 1
    ELSE
        TH(IMIN+1) = -TH3+ALF0
        ITLIN(IMIN+1) = 3
    END IF

*------ REGION 3
    AM(IMIN) = AM3
    P(IMIN)  = POLD(IMIN-1)
    PT(IMIN) = PT3*PTOLD(IMIN+1)
    ALF(IMIN)= ALFOLD(IMIN-1)

    RETURN 1

*------------------------------
*   WAVES OF FAMILY II
*------------------------------
   200 CONTINUE
    AM0 = AM(IMIN-1)
    ALF0 = ALF(IMIN-1)
    ALF2 = ALF(IMIN+1)

*------ CALCULATE NEW FIELD
    IF ((ALF2-ALF0).GT.0.) THEN
        D = ALF2-ALF0
        LSHANG = .TRUE.
   CALL SHANG (AM0,D,G,LSHANG,TH3,P3,PT3,AM3) ELSE
        D = ABS(ALF2-ALF0)
 CALL PM (AM0,D,G,AM3,TH3FW,TH3RW,TH3,P3) PT3 = 1.
    END IF

*------ ASSIGNING THE NEW LINES AND FIELDS *------ RESULTING WAVE
    IF ((ALF2-ALF0).GT.0.) THEN
 TH(IMIN) = TH3+ALF0 ITLIN(IMIN) = 2
        ELSE
 TH(IMIN) = TH3+ALF0 ITLIN(IMIN) = 4
        END IF

*------ SLIP LINE
        TH(IMIN+1) = ALF2
        ITLIN(IMIN+1) = 5

*------ REGION 3
        AM(IMIN) = AM3
        P(IMIN)  = POLD(IMIN+1)
 PT(IMIN) = PT3*PTOLD(IMIN-1) ALF(IMIN)= ALFOLD(IMIN+1)
```

```
      RETURN 1
      END
*
*
      SUBROUTINE EEDF (G,IMIN,NL,*,*,IPRINT) *----------------------------
---------------------------------------------*        SUBROUTINE 'EEDF' CALCU-
LATES THE FLOW CONDITIONS BEHIND TWO
*     INTERSECTING EXPANSION WAVES OF DIFFERENT FAMILIES,
*-------------------------------------------------------------------------*
*     REGION 1: BEHIND LOWER FIRST EXPANSION WAVE
*     REGION 2: BEHIND UPPER FIRST EXPANSION WAVE
*     REGION 3: BEHIND UPPER SECOND EXPANSION WAVE
*     REGION 4: BEHIND LOWER SECOND EXPANSION WAVE
*
*-------------------------------------------------------------------------

      PARAMETER (NLMAX=201,NFMAX=200)

DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
     1      ALF(NFMAX),ITLIN(NLMAX)
DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),
     1      PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

      LOGICAL LPM,LOUT

COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
     1        THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT

*-------------------------------------------------------------------990 FOR-
MAT (42HEXPANSION/EXPANSION, DIFF. FAMILIES AT X= ,F7.3)
  991 FORMAT (6HLINES ,I3,5H AND ,I3)
996 FORMAT (22HSLIP LINE IS NEGLECTED)

      IF (LOUT) THEN
       IF (IPRINT) 2,1,1
  1
       IMINPL = IMIN+1

  2   IPRINT = 1
      END IF

      PI = 3.141592654

*---------------------------
*   PROVIDE INITIAL VALUES
*---------------------------
      ALF1 = ALF(IMIN+1)
      ALF2 = ALF(IMIN-1)
   ALF0 = ALF(IMIN)
   AM1= AM(IMIN+1)
   AM2= AM(IMIN-1)

*------ FIRST GUESS FOR THE NEW FLOW ANGLE OMEGA
   OMEGA = ALF1+(ALF2-ALF0)
```

```
*—— CALCULATE MACH NUMBERS BEHIND SECOND EXPANSION WITH GUESSED
OMEGA D3 = ABS(OMEGA-ALF2)
    CALL PM (AM2,D3,G,AM3,THZZ,TH2ZZ,TH4ZZ,PZZ)

    LPM = .FALSE.
    D4 = OMEGA-ALF1
    CALL PM (AM1,D4,G,AM4,THZZ,TH2ZZ,TH4ZZ,PZZ)

*    ITERATE THE NEW MACH NUMBERS AM3 AND AM4
*            (NEWTON'S METHOD)
*______________________________________________
    NTH = 1
    P1 = P(IMIN+1)
    P2 = P(IMIN-1)
    GP = G+1.
    GM = G-1.
    GPM = GP/GM
   50 CONTINUE
    AM3OLD = AM3
    AM4OLD = AM4
    AMQ2 = AM2**2-1.
    AMQ3 = AM3**2-1.
    AMG2 = 1.+GM/2.*AM2**2
    AMG3 = 1.+GM/2.*AM3**2
    AMQ1 = AM1**2-1.
    AMQ4 = AM4**2-1.
    AMG1 = 1.+GM/2.*AM1**2
    AMG4 = 1.+GM/2.*AM4**2

*——— EVALUATE THE F-FUNCTION FF
    FF1 = (AMG3/AMG2)**(-G/GM)
    FF2 = (AMG4/AMG1)**(-G/GM)
    FF = FF1-P1/P2*FF2

*——— EVALUATE THE G-FUNCTION GF
GF1 = SQRT(GPM)*ATAN(SQRT(1./GPM*AMQ3))-ATAN(SQRT(AMQ3)) GF2 =
SQRT(GPM)*ATAN(SQRT(1./GPM*AMQ2))-ATAN(SQRT(AMQ2))    GF3    =
SQRT(GPM)*ATAN(SQRT(1./GPM*AMQ4))-ATAN(SQRT(AMQ4))    GF4    =
SQRT(GPM)*ATAN(SQRT(1./GPM*AMQ1))-ATAN(SQRT(AMQ1)) GF = ALF2-GF1+GF2-
ALF1-GF3+GF4

*——— EVALUATE THE DERIVATIVES
    FAM3 = -G*AM3*AMG2**(G/GM)*AMG3**(-G/GM-1.)
FAM4 = P1/P2*G*AM4*AMG1**(G/GM)*AMG4**(-G/GM-1.)

    GAM3 = 1./SQRT(AMQ3)*(1./AM3-AM3/(1.+1./GPM*AMQ3))
    GAM4 = 1./SQRT(AMQ4)*(1./AM4-AM4/(1.+1./GPM*AMQ4))

    DETI =   1./(FAM3*GAM4-FAM4*GAM3)

*——— NEW VALUES FOR TH3, AM4
    AM3 = DETI*(-GAM4*FF+FAM4*GF)+AM3OLD
    AM4 = DETI*(GAM3*FF-FAM3*GF)+AM4OLD
```

```
IF (ABS(AM3-AM3OLD).LT..0001.AND.ABS(AM4-AM4OLD).LT..0001) THEN GO TO 100
    ELSE
        NTH = NTH+1
        IF (NTH.GT.10) THEN
WRITE (6,*) ' TOO MANY ITERATIONS' GOTO 3333
            END IF
            GO TO 50
    END IF

*——— NEW FLOW ANGLE OMEGA
   100 CONTINUE
    OMEGA3 = ALF2-GF1+GF2
    OMEGA4 = GF3-GF4+ALF1

*——— FLOW CONDITIONS IN REGIONS 3 AND 4 OMEGA = (OMEGA3+OMEGA4)/2.
    D3 = ABS(OMEGA-ALF2)
CALL PM (AM2,D3,G,AM3,TH3FW,TH3RW,TH3,P3) D4 = OMEGA-ALF1
CALL PM (AM1,D4,G,AM4,TH4FW,TH4RW,TH4,P4) TH4 = -TH4

*————————————————————————————————* ASSIGNING THE NEW LINES AND
FIELDS
*————————————————————————————————*——— CHECK IF SLIP LINE IS
NEGLECTABLE PT3 = PTOLD(IMIN-1)
    PT4 = PTOLD(IMIN+1)
IF (ABS(AM3-AM4).LE.0.010.AND.ABS((PT3-PT4)/PT3).LE.0.010) THEN ISLIP = 1
    ELSE
        ISLIP = -1
    END IF
    IF (ISLIP) 200,200,300

*————————————————————————————————
*   REGULAR PROCEDURE, SLIP LINE IS ADDED
*————————————————————————————————
   200 CONTINUE
    NL = NL+1

*——— UPPER EXPANSION WAVE
    TH(IMIN) = TH3+ALF2
    ITLIN(IMIN) = 4

*——— SLIP LINE
    TH(IMIN+1) = OMEGA
    ITLIN(IMIN+1) = 5

*——— LOWER EXPANSION WAVE
    TH(IMIN+2) = TH4+ALF1
    ITLIN(IMIN+2) = 3

*——— REGION 3
    AM(IMIN) = AM3
    P(IMIN)  = P3*POLD(IMIN-1)
    PT(IMIN) = PTOLD(IMIN-1)
    ALF(IMIN)= OMEGA
```

```
*——— REGION 4
    AM(IMIN+1) = AM4
    P(IMIN+1)  = P4*POLD(IMIN+1)
    PT(IMIN+1) = PTOLD(IMIN+1)
    ALF(IMIN+1)= OMEGA
3333
    RETURN 1

*———————————————————————————————————————————————————*
* SMALL DIFFERENCES BETWEEN REGION 3 AND 4, SLIP LINE NEGLECTED
*———————————————————————————————————————————————————300 CON-
TINUE

*       IF (LOUT) WRITE (19,996)

*———— UPPER EXPANSION WAVE
        TH(IMIN) = TH3+ALF2
        ITLIN(IMIN) = 4

*——— LOWER EXPANSION WAVE
        TH(IMIN+1) = TH4+ALF1
        ITLIN(IMIN+1) = 3

*——— REGION 3 = REGION 4
        AM(IMIN) = (AM3+AM4)/2.
        P(IMIN)  = P3*POLD(IMIN-1)
        PT(IMIN) = (PT3+PT4)/2.
        ALF(IMIN)= OMEGA

        RETURN 2
        END
*
*
        SUBROUTINE EESF (G,IMIN,NL,*) *————————————————————
————————————————————————*          SUBROUTINE 'EESF' CALCULATES THE FLOW
CONDITIONS BEHIND TWO
*       INTERSECTING EXPANSION WAVES OF THE SAME FAMILY
*
*       REGION 0  : 'UNDISTURBED' FLOW
*       REGION 1  : BEHIND FIRST INCOMING EXPANSION
*       REGION 2  : BEHIND SECOND INCOMING EXPANSION
*
*       ISENTROPIC FLOW OVER EXPANSION: ONLY ONE RESULTING EXPANSION
*       WAVE IS ASSUMED
*———————————————————————————————————————————————————

        PARAMETER (NLMAX=201,NFMAX=200)

DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
   1      ALF(NFMAX),ITLIN(NLMAX)
DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),
   1      PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

        LOGICAL LPM,LOUT
```

```
      COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
     1             THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT

*-----------------------------------------------------------------------990 FOR-
MAT (41HEXPANSION/EXPANSION, SAME FAMILIES AT X= ,F7.3)
  991 FORMAT (6HLINES ,I3,5H AND ,I3)

      IF (LOUT) THEN
*       WRITE (19,990) X0
        IMINPL = IMIN+1
*       WRITE (19,991) IMIN,IMINPL
      END IF

      PI = 3.141592654

      GO TO (100,200,100,200) ITLIN(IMIN)

*-------------------------
*  WAVES OF FAMILY 1
*-------------------------
  100 CONTINUE

*-----------------------------------*        CALCULATE THE RESULTING WAVE
*-----------------------------------AM0 = AM(IMIN+1)
      ALF0 = ALF(IMIN+1)
      ALF2 = ALF(IMIN-1)
      D = ALF2-ALF0
      CALL PM (AM0,D,G,AM3,TH3FW,TH3RW,TH3,P3)

*------ RESULTING EXPANSION WAVE
      NL = NL-1
      TH(IMIN) = -TH3+ALF0
      ITLIN(IMIN) = 3

      RETURN 1

*-------------------------
*  WAVES OF FAMILY 2
*-------------------------
  200 CONTINUE

*------ CALCULATE THE RESULTING WAVE
      AM0 = AM(IMIN-1)
      ALF0 = ALF(IMIN-1)
      ALF2 = ALF(IMIN+1)
      D = ABS(ALF2-ALF0)
      CALL PM (AM0,D,G,AM3,TH3FW,TH3RW,TH3,P3)

*------ RESULTING EXPANSION WAVE
      NL = NL-1
      TH(IMIN) = TH3+ALF0
      ITLIN(IMIN) = 4

      RETURN 1
```

```
      END
*
*
      SUBROUTINE SRSPA (G,IMIN,NL,*,*) *--------------------------------
-----------------------------------*      SUBROUTINE 'SRSPA' CALCULATES
THE FLOW CONDITIONS BEHIND THE
*     REFLECTION OF A SHOCK OF FAMILY 2 AT A SLIP LINE
*     RESULTING FLOWFIELD IS SIMILAR TO THE INTERSECTION OF SHOCK AND
*     EXPANSION WAVES OF DIFFERENT FAMILIES SO THAT THE SUBROUTINE
*     'SEDFA' CAN BE USED
*     CASE A: SHOCK OF FAMILY 2
*
*     REGION 1: BEHIND INCOMING SHOCK
*     REGION 2: BEHIND SLIP LINE
*     REGION 3: BEHIND CREATED SECOND SHOCK
*     REGION 4: BEHIND REFLECTED EXPANSION WAVE
**-----------------------------------------------------------------------
PARAMETER (NLMAX=201,NFMAX=200)

DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
   1      ALF(NFMAX),ITLIN(NLMAX)
DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),
   1      PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

      LOGICAL LOUT

COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
   1          THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT

*-----------------------------------------------------------------------
990 FORMAT (44HSHOCK REFLECTION AT SLIP LINE, CASE A AT X= ,F7.3)
  991 FORMAT (6HLINES ,I3,5H AND ,I3)

      IF (LOUT) THEN
*       WRITE (19,990) X0
       IMINPL = IMIN+1
*       WRITE (19,991) IMIN,IMINPL
      END IF

      IPRINT = -1
      JUSE = 2
      CALL SEDFA (G,IMIN,NL,*100,*200,IPRINT,JUSE)

  100 CONTINUE
      RETURN 1

  200 CONTINUE
      RETURN 2
      END
*
*
      SUBROUTINE SRSPB (G,IMIN,NL,*,*) *--------------------------------
-----------------------------------*      SUBROUTINE 'SRSPB' CALCULATES
THE FLOW CONDITIONS BEHIND THE
```

```
*       REFLECTION OF A SHOCK OF FAMILY 1 AT A SLIP LINE
*       RESULTING FLOWFIELD IS SIMILAR TO THE INTERSECTION OF SHOCK AND
*       EXPANSION WAVES OF DIFFERENT FAMILIES SO THAT THE SUBROUTINE
*       'SEDFB' CAN BE USED
*       CASE B: SHOCK OF FAMILY 1
*
*       REGION 1: BEHIND SLIP LINE
*       REGION 2: BEHIND INCOMING SHOCK
*       REGION 3: BEHIND REFLECTED EXPANSION WAVE
*       REGION 4: BEHIND CREATED SECOND SHOCK
* *-----------------------------------------------------------------
      PARAMETER (NLMAX=201,NFMAX=200)

      DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
     1          ALF(NFMAX),ITLIN(NLMAX)
      DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),
     1          PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

            LOGICAL LOUT

      COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
     1              THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT

*----------------------------------------------------------------990 FOR-
MAT (44HSHOCK REFLECTION AT SLIP LINE, CASE B AT X= ,F7.3) 991 FORMAT
(6HLINES ,I3,5H AND ,I3)

            IF (LOUT) THEN
*             WRITE (19,990) X0
             IMINPL = IMIN+1
*             WRITE (19,991) IMIN,IMINPL
            END IF

            IPRINT = -1
            JUSE = 2
            CALL SEDFB (G,IMIN,NL,*100,*200,IPRINT,JUSE)

        100 CONTINUE
            RETURN 1

        200 CONTINUE
            RETURN 2
            END
*
*
            SUBROUTINE SRW (G,IMIN,NL,*) *-----------------------------
-----------------------------*        SUBROUTINE 'SRW' CALCULATES THE FLOW
CONDITIONS BEHIND THE
*       REFLECTION OF A SHOCK AT A SOLID WALL
*
*       REGION 1: IN FRONT OF INCOMING SHOCK
*       REGION 2: BEHIND INCOMING SHOCK/IN FRONT OF REFLECTED SHOCK
*       REGION 3: BEHIND REFLECTED SHOCK
* *-----------------------------------------------------------------
```

```
      PARAMETER (NLMAX=201,NFMAX=200)

      DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
     1      ALF(NFMAX),ITLIN(NLMAX)
      DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),
     1      PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

      LOGICAL LSHANG,LOUT

      COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
     1          THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT

*------------------------------------------------------------------990
      FORMAT(33HSHOCK/SHOCK, SAME FAMILIES AT X= ,F7.3)
  991 FORMAT (6HLINES ,I3,5H AND ,I3)

      IF (LOUT) THEN
*       WRITE (19,990) X0
        IMINPL = IMIN+1
*       WRITE (19,991) IMIN,IMINPL
      END IF

      PI = 3.141592654

      GO TO (100,100,50,50,50,200) ITLIN(IMIN)

   50 CONTINUE
*      WRITE (19,*) 'ERROR! SHOCK REFLECTION AT WALL WITH'
*      WRITE (19,*) 'TYPE(1)= ',ITLIN(IMIN),' TYPE(2)= ',ITLIN(IMIN+1)
      GOTO 3333

*-----------------------------
*   SHOCK IS ABOVE WALL
*-----------------------------
  100 CONTINUE
      AM2 = AM(IMIN-1)
      ALF1 = ALF(IMIN)
      ALF2 = ALF(IMIN-1)

*------- CALCULATE THE NEW FIELD
      D = ALF1-ALF2
      LSHANG = .TRUE.
      CALL SHANG (AM2,D,G,LSHANG,TH3,P3,PT3,AM3)

*----- ASSIGN NEW LINES AND FIELDS -----*------- REFLECTED SHOCK
      TH(IMIN) = TH3+ALF2
      ITLIN(IMIN) = 2

*------- REGION 3
      AM(IMIN) = AM3
      P(IMIN)  = P3*POLD(IMIN-1)
      PT(IMIN)  = PT3*PTOLD(IMIN-1) ALF(IMIN) = ALFOLD(IMIN)

      RETURN 1
```

```
*________________________________
*  SHOCK IS BELOW WALL
*________________________________
   200 CONTINUE
       AM2 = AM(IMIN+1)
       ALF1 = ALF(IMIN)
       ALF2 = ALF(IMIN+1)

*—— CALCULATE THE NEW FIELD ——
       D = ABS(ALF1-ALF2)
       LSHANG = .TRUE.
CALL SHANG (AM2,D,G,LSHANG,TH3,P3,PT3,AM3) TH3 = -TH3

*—— ASSIGN NEW LINES AND FIELDS ——*——— REFLECTED SHOCK
       TH(IMIN+1) = TH3+ALF2
       ITLIN(IMIN+1) = 1

*——— REGION 3
       AM(IMIN)  = AM3
       P(IMIN)   = P3*POLD(IMIN+1)
       PT(IMIN)  = PT3*PTOLD(IMIN+1)
       ALF(IMIN) = ALFOLD(IMIN)
   3333
       RETURN 1
       END
*
*
       SUBROUTINE ERSPA (G,IMIN,NL,*,*) *———————————————
———————————————————*        SUBROUTINE 'ERSPA' CALCULATES
THE FLOW CONDITIONS BEHIND THE
*      REFLECTION OF AN EXPANSION WAVE OF FAMILY 2 AT A SLIP LINE
*      RESULTING FLOWFIELD IS SIMILAR TO THE INTERSECTION OF SHOCK AND
*      EXPANSION WAVES OF DIFFERENT FAMILIES SO THAT THE SUBROUTINE
*      'SEDFB' CAN BE USED
*      CASE A: EXPANSION WAVE OF FAMILY 2
*
*      REGION 1: BEHIND INCOMING EXPANSION WAVE
*      REGION 2: BEHIND SLIP LINE
*      REGION 3: BEHIND CREARTED EXPANSION WAVE
*      REGION 4: BEHIND REFLECTED SHOCK
* *——————————————————————————————————————————
PARAMETER (NLMAX=201,NFMAX=200)

DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
   1      ALF(NFMAX),ITLIN(NLMAX)
DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),
   1      PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

       LOGICAL LOUT
COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
   1         THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT

*——————————————————————————————————————————————————990 FOR-
MAT (48HEXP. WAVE REFLECTION AT SLIP LINE, CASE A AT X= ,F7.3) 991 FORMAT
```

```
(6HLINES ,I3,5H AND ,I3)

      IF (LOUT) THEN
*       WRITE (19,990) X0
        IMINPL = IMIN+1
*       WRITE (19,991) IMIN,IMINPL
      END IF

      IPRINT = -1
      JUSE = 3
      CALL SEDFB (G,IMIN,NL,*100,*200,IPRINT,JUSE)

  100 CONTINUE
      RETURN 1

  200 CONTINUE
      RETURN 2
      END
*
*
      SUBROUTINE ERSPB (G,IMIN,NL,*,*) *-----------------------------
----------------------------------*     SUBROUTINE 'ERSPB' CALCULATES
THE FLOW CONDITIONS BEHIND THE
*     REFLECTION OF AA EXPANSION WAVE OF FAMILY 1 AT A SLIP LINE
*     RESULTING FLOWFIELD IS SIMILAR TO THE INTERSECTION OF SHOCK AND
*     EXPANSION WAVES OF DIFFERENT FAMILIES SO THAT THE SUBROUTINE
*     'SEDFA' CAN BE USED
*     CASE B: EXPANSION WAVE OF FAMILY 1
*
*     REGION 1: BEHIND SLIP LINE
*     REGION 2: BEHIND INCOMING EXPANSION WAVE
*     REGION 3: BEHIND REFLECTED SHOCK
*     REGION 4: BEHIND CREATED SECOND EXPANSION WAVE
* *----------------------------------------------------------------
PARAMETER (NLMAX=201,NFMAX=200)

DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
   1       ALF(NFMAX),ITLIN(NLMAX)
DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),
   1       PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

      LOGICAL LOUT

COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
   1          THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT

*---------------------------------------------------------------990FOR-
MAT (48HEXP. WAVE REFLECTION AT SLIP LINE, CASE B AT X= ,F7.3) 991 FORMAT
(6HLINES ,I3,5H AND ,I3)

      IF (LOUT) THEN
*       WRITE (19,990) X0
        IMINPL = IMIN+1
*       WRITE (19,991) IMIN,IMINPL
```

```
      END IF

      IPRINT = -1
      JUSE = 3
      CALL SEDFA (G,IMIN,NL,*100,*200,IPRINT,JUSE)

  100 CONTINUE
      RETURN 1

  200 CONTINUE
      RETURN 2
      END
*
*
      SUBROUTINE ERW (G,IMIN,NL,*)
*---------------------------------------------------------------------*
    SUBROUTINE 'ERW' CALCULATES THE FLOW CONDITIONS BEHIND THE
*     REFLECTION OF AN EXPANSION WAVE AT A SOLID WALL
*
*     REGION 1: IN FRONT OF INCOMING EXPANSION WAVE
*     REGION 2: BEHIND INCOMING EXPANSION WAVE/IN FRONT OF REFLECTED
*          EXPANSION WAVE
*     REGION 3: BEHIND REFLECTED EXPANSION WAVE
* *-------------------------------------------------------------------
PARAMETER (NLMAX=201,NFMAX=200)

DIMENSION YL(NLMAX),TH(NLMAX),AM(NFMAX),P(NFMAX),PT(NFMAX),
    1     ALF(NFMAX),ITLIN(NLMAX)
DIMENSION YLOLD(NLMAX),THOLD(NLMAX),AMOLD(NFMAX),POLD(NFMAX),
    1     PTOLD(NFMAX),ALFOLD(NFMAX),ITLOLD(NLMAX)

      LOGICAL LPM,LOUT

COMMON/INLT0/ AM,P,PT,ALF,TH,ITLIN,AMOLD,POLD,PTOLD,ALFOLD,
    1       THOLD,ITLOLD,YL,YLOLD,X0,X0OLD,LOUT

*---------------------------------------------------------------------990 FOR-
MAT (35HEXP. WAVE REFLECTION AT WALL AT X= ,F7.3)
  991 FORMAT (6HLINES ,I3,5H AND ,I3)

      IF (LOUT) THEN
*       WRITE (19,990) X0
        IMINPL = IMIN+1
*       WRITE (19,991) IMIN,IMINPL
      END IF

      PI = 3.141592654

      GO TO (50,50,100,100,50,200) ITLIN(IMIN)

   50 CONTINUE
*     WRITE (19,*) 'ERROR! EXP. WAVE REFLECTION AT WALL WITH'
*     WRITE (19,*) 'TYPE(1)= ',ITLIN(IMIN),' TYPE(2)= ',ITLIN(IMIN+1)
      GOTO 3333
```

```
*________________________________________
*   EXPANSION WAVE IS ABOVE WALL
*________________________________________
   100 CONTINUE
       AM2 = AM(IMIN-1)
       ALF1 = ALF(IMIN)
       ALF2 = ALF(IMIN-1)

*——— CALCULATE THE NEW FIELD
       D = ABS (ALF1-ALF2)
       CALL PM (AM2,D,G,AM3,TH3FW,TH3RW,TH3,P3)

*—— ASSIGN NEW LINES AND FIELDS ——*——— REFLECTED EXPANSION WAVE
       TH(IMIN) = TH3+ALF2
       ITLIN(IMIN) = 4

*——— REGION 3
       AM(IMIN)  = AM3
       P(IMIN)   = P3*POLD(IMIN-1)
       PT(IMIN)  = PTOLD(IMIN-1)
       ALF(IMIN) = ALFOLD(IMIN)

       RETURN 1

*————————————————————————————————*  EXPANSION WAVE IS BELOW WALL
*————————————————————————————————200 CONTINUE
   AM2 = AM(IMIN+1)
   ALF1 = ALF(IMIN)
   ALF2 = ALF(IMIN+1)

*——— CALCULATE THE NEW FIELD
   D = ABS(ALF1-ALF2)
CALL PM (AM2,D,G,AM3,TH3FW,TH3RW,TH3,P3) TH3 = -TH3

*—— ASSIGN NEW LINES AND FIELDS ——*——— REFLECTED EXPANSION WAVE
   TH(IMIN+1) = TH3+ALF2
   ITLIN(IMIN+1) = 3

*——— REGION 3
   AM(IMIN)  = AM3
   P(IMIN)   = P3*POLD(IMIN+1)
   PT(IMIN)  = PTOLD(IMIN+1)
   ALF(IMIN) = ALFOLD(IMIN)
 3333
   RETURN 1
   END
*
*
   SUBROUTINE SHANG (AM1,D,G,LSHANG,TH,PR,PTR,AMY) *————————————
———————————————————————————————————*   SUBROUTINE   SHANG
FINDS THE SHOCK ANGLE FOR A GIVEN FREESTREAM
*  MACH NUMBER, WEDGE ANGLE AND GIVEN RATIO OF SPECIFIC HEATS
*  THE FLOW CONDITIONS BEHIND THE SHOCK ARE ALSO COMPUTED (OPTION)
*  SUBROUTINE BY BOB ROACH, GEORGIA INSTITUTE OF TECHNOLOGY
```

```
*
*  INPUT
*  AM1    = UPTREAM MACH NUMBER
*  D        = WEDGE ANGLE (RAD)
*  G        = RATIO OF SPECIFIC HEATS
*  LSHANG = TRUE: ADDITIONAL COMPUTATION OF POST SHOCK VALUES
*
*  OUTPUT
*  TH      = SHOCK WAVE ANGLE (RAD)
*  PR      = PRESSURE      (NON-DIMENSIONAL)
*  R       = DENSITY        (     "     )
*  T       = TEMPERATURE   (     "     )
*  PTR  = TOTAL PRESSURE (     "     )
*  AMY  = MACH NUMBER BEHIND SHOCK
*
*------------------------------------------------------------------------LOGICAL
LSHANG
              CALL InitBeachBall(512)
              CALL ShowBeachBall

*------ CHECK MACH NUMBER
   IF (AM1.LT.1.) THEN
*          WRITE (19,*) ' INPUT MACH NUMBER TO SHANG SUBSONIC !'
*          WRITE (19,*) ' PLEASE CHECK AGAIN !'
*          WRITE (9,*) ' RETURNING TO MAIN PROGRAM ......'
        RETURN
          GOTO 3333
   END IF

   PI = 3.141592654
   GP = G + 1.
   GM = G - 1.
   AM2 = AM1**2
   CD = 1./TAN(D)

*------ CHECK TO SEE IF DELTA > DELTAMAX
   GSTF      = GP*AM2
   CC        = (GSTF-4.+SQRT(GP*(AM2*(GSTF+8.*GM)+16.)))/(4.*G*AM2)
   THMAX = ASIN(SQRT(CC))
   STM2      = (SIN(THMAX))**2
   DMAX      = ATAN2(1.,(TAN(THMAX)*(.5*GSTF/(AM2*STM2-1.)-1.)))
   DMAXD = 180.*DMAX/PI
   IF (D.GT.DMAX) THEN
          WRITE (6,*) ' IN SHANG...'
WRITE (6,*) '   WEDGE ANGLE TOO LARGE FOR MACH NUMBER' WRITE (6,*) '
DELTAMAX= ',DMAXD,' FOR M= ',AM1 WRITE (6,*) ' FOR THE GIVEN DELTA, THETA
= 90 DEG.' WRITE (6,*) ' FLOW BECOMES SUBSONIC !'
          GOTO 3333
   END IF

*------ FIRST GUESS: PARABOLIC CURVE FIT TO FUNCTION
   AMI = 1./AM1
   SIG = ATAN2(AMI,SQRT(1.-AMI**2))
   DL     = THMAX-SIG
```

```
      DL2  = DL*(THMAX+SIG)
      AA     = DMAX/(DL2-2.*DL*THMAX)
      BB     = 2.*DMAX*THMAX/(2.*DL*THMAX-DL2)
      CC     = -(AA*SIG+BB)*SIG
      TH  =  .5*(-BB+SQRT(BB**2-4.*AA*(CC-D)))/AA

*——— NEWTON ITERATE FOR THETA
      NTH  = 1
     100 CONTINUE
      TH1  = TH
      ST      = SIN(TH)
      ST2  = ST**2
      CT      = COS(TH)
      TT      = ST/CT
      SC2  = 1./CT**2
      STF  = AM2*ST2-1.
      STF2 = .5*GSTF/STF-1.
      F             = TT*STF2-CD
      FP      = SC2*STF2-ST2*(AM2*GSTF/STF**2)
      TH      = TH1-F/FP
IF (ABS(TH-TH1).LT..001) GO TO 200
      NTH  = NTH + 1
      IF (NTHETA.GE.20) THEN
WRITE (6,*) ' TOO MANY ITERATIONS IN SHANG' GOTO 3333
      ELSE
          GO TO 100
      END IF
CALL SpinBeachBall(1)

     200 CONTINUE

      IF (LSHANG) GO TO 300
      RETURN

*——— COMPUTE POST-SHOCK VALUES
     300 CONTINUE
      AMN2      = (AM1*SIN(TH))**2
      PR        = (2.*G*AMN2 - GM)/GP
      PT1       = (GP*AMN2/(GM*AMN2 + 2.))**(G/GM)
      PT2       = (GP/(2.*G*AMN2 - GM))**(1./GM)
      PTR       = PT1*PT2
CALL SpinBeachBall(1)

      AMNUM = (GP*AM1)**2*AMN2 - 4.*(AMN2 - 1.)*(G*AMN2 + 1.)
      AMDEN = (2.*G*AMN2 - GM)*(GM*AMN2 + 2.)
      AMY   = SQRT(AMNUM/AMDEN)
            CALL SpinBeachBall(1)

*————— ALL DONE —————
 3333
      RETURN
            CALL RelBeachBall

      END
```

```
*
*
    SUBROUTINE PM (AM1,D,G,AMNY,THFW,THRW,TH,PN) *----------------
-------------------------------------------* SUBROUTINE PM COM-
PUTES THE MACH NUMBER DOWNSTREAM OF AN ISENTROPIC
*  TURN OF DELTA USING THE PRANDTL-MEYER FUNCTION
*  PRESSURE BEHIND THE EXPANSION AND FORWARD AND REARWARD MACHLINE
*  OF THE FAN ARE ALSO COMPUTED (OPTION)
*  SUBROUTINE BY BOB ROACH, GEORGIA INSTITUTE OF TECHNOLOGY
*
*  INPUT
*  AM1  = MACH NUMBER OF INCOMING AIRFLOW
*  D    = TURNING ANGLE (RAD)
*  G    = RATIO OF SPECIFIC HEATS
*  LPM  = TRUE: COMPUTATION OF POST EXPANSION VALUES
*
*  OUTPUT:
*  AMNY  = MACH NUMBER BEHIND EXPANSION
*  THFW  = ANGLE OF FORWARD MACHLINE (RAD)
*  THRW  = ANGLE OF REARWARD MACHLINE (RAD)
*  TH    = ANGLE OF AVERAGE MACHLINE (RAD)
*  PN    = STATIC PRESSURE (NON-DIMENSIONAL)
**-------------------------------------------------------------------
LOGICAL LPM
          CALL InitBeachBall(512)
          CALL ShowBeachBall


   PI = 3.141592654
   GP = G+1.
   GM = G-1.
   GR = GP/GM

*------COMPUTE PM-FUNCTION FOR INCOMING AIRFLOW S1 = SQRT(1./GR*(AM1**2-
1.))
   S2 = SQRT(AM1**2-1.)
ANU = SQRT(GR)*ATAN(S1)-ATAN(S2)

*------ NEW TURNING ANGLE
   DST = D+ANU

*------ FIRST GUESS - RATIONAL FUNCTION FIT AAA =  .5*PI*(SQRT(GR)-1.)
AMN1 = 1./(1.-(DST/AAA)**(1./GP))

*------ NEWTON ITERATION
   NAMN = 1
  100 CONTINUE
   NAMN = NAMN+1
   AMNL = AMN1
          CALL SpinBeachBall(1)

   PGF  = SQRT(AMN1**2-1.)
F   = SQRT(GR)*ATAN2(PGF,SQRT(GR))-ATAN(PGF)-DST
   FP   = (AMN1/(1.+PGF**2/GR)-1./AMN1)/PGF
```

```
      AMN1 = AMN1-F/FP
                  CALL SpinBeachBall(1)

      IF (ABS(AMNL-AMN1).LT..001) GO TO 200
      IF (NAMN.GE.20) THEN
   WRITE (6,*) 'TOO MANY ITERATIONS IN PM'
         GOTO 3333
      ELSE
         GO TO 100
      END IF
                  CALL SpinBeachBall(1)

     200 CONTINUE
      AMNY = AMN1

*——— COMPUTE POST EXPANSION VALUES
PN = ((1.+GM/2.*AMNY**2)/(1.+GM/2.*AM1**2))**(-G/GM) THFW = ASIN(1./AM1)
      THRW = ASIN(1./AMNY)-D
      TH = (THFW+THRW)/2.

*——— ALL DONE ———
 3333
                    CALL RelBeachBall

      RETURN

      END
*
*
        SUBROUTINE MSUP(AR,G,AM) *————————————————————
—————————————*
*       This subroutine is used to compute the supersonic
*       Mach number corresponding to a given input area
*       ratio, A/A* (AR), and ratio of specific heats (G).
```

```
*
      SUBROUTINE RAYLEIGH(N,AM3,P03,T03,F,ER,AM4,P04,T04,T4,P4)
*----------------------------------------------------------------------
*
*    This subroutine computes the combustion zone heat addition in the
*         scramjet attitude sensitivity program.  Flow properties from the
*         exit of the diffuser are passed from the INLET subroutine
*         as well as the minimum allowed combustor outflow Mach number.
*         If the desired Mach number results in a combustor temperature that
*         is higher than that allowable,the exit Mach number and fuel flow rate
*         are adjusted to limit the system temperature to a specified maximum value.
*         No limit is placed on equivalence ratio. Approximate corrections are made
*         to the gas properties to take account of fuel rich operation when the
*         minimum specified equivalence ratio is higher than that computed in the
*         algorithm.
*
*    Variables:
*         AM3     combustor inflow Mach number
*         T03    combustor inflow stagnation temperature
*         P03    combustor inflow stagnation pressure
*         AM4     combustor outflow Mach number
*         T04    combustor outflow stagnation temperature
*         P04    combustor outflow stagnation pressure
*         F      fuel/air mass flow rate ratio
*         FCRIT  critical (stoichiometric) fuel/air ratio
*         ETAB   combustor thermodynamic efficiency
*                    ER      equivalence ratio specified in input
*         QR     fuel heating value
*                (for H2 = 51600 BTU/#m = 1.2927E+9 ft-#f/slug)
*         G1     ratio of specific heats for air
*         G2     ratio of specific heats for the fuel/air mixture
*         RG     gas constant for air
*
*----------------------------------------------------------------------
      COMMON/CMBST/ G2,RG2,CP2,G2M,G2P,QR,ETAB,FCRIT
      COMMON/GASPR/ G,RG,GM,GP,CP
            KOUNTER = 0
*------------------- DATA -------------------

            TMAX = 5500.
            FGIVEN = ER*FCRIT
            AM4 = 1.15
            G1  = G
      RG1 = RG
      G1M = G1 - 1.
      G1P = G1 + 1.
      CP1 = G1*RG1 / G1M

  100 CONTINUE
        KOUNTER = KOUNTER +1
      G2M = G2 - 1.
      G2P = G2 + 1.
      CP2 = G2*RG2 / G2M
```

```
            T3 = T03/(1.+ .5*G2M*AM3**2)
            P3 = P03/(1.+ .5*G2M*AM3**2)**(G2/G2M)

*——— RAYLEIGH LINE HEAT ADDITION:

      RS1=(1.+G2*AM3**2)/(1+G2*AM4**2)
            RS2=(AM4/AM3)**2
            RS3=(1.+ 0.5*G2M*AM4**2)/(1.+ 0.5*G2M*AM3**2)
            T4 = T3*(RS1**2)*RS2
            P4 = P3*RS1
            P04 = P03*RS1*RS3**(G2/(G2-1.))
            T04 = T03*(RS1**2)*RS2*Rs3
              FMAX = (T04 - T03)/(ETAB*QR/CP2 - T04)
*           WRITE(6,*) 'PASS ',KOUNTER
*           WRITE(6,*) 'M4 =',AM4,'T4 =',T4,'T04 =',T04
*           WRITE(6,*) 'Emax =',FMAX/FCRIT
*           WRITE(6,*) ' '
            IF (FMAX.GT.FCRIT) THEN
*                 WRITE(6,*),'    Adjust the Exit Mach Number'
*                 WRITE(6,*), '    for allowable energy input'
                  FMAX = FCRIT
                  T04 = (T03 + FMAX*ETAB*QR/CP2)/(1. + FMAX)

101   CONTINUE
                  RS1=(1.+G2*AM3**2)/(1+G2*AM4**2)
                  RS2=(AM4/AM3)**2
                  RS3=(1.+ 0.5*G2M*AM4**2)/(1.+ 0.5*G2M*AM3**2)

                  S1 = RS1*RS1*RS2*RS3 - T04/T03
                  S2 = G2M*AM4/(1.+ 0.5*G2M*AM4**2)
                  S2 = S2 + 2./AM4 - 4*G2*AM4/(1.+ G2*AM4**2)
                  S2 = RS1*RS1*RS2*RS3*S2
                        DAM = - S1/S2
*                       write(6,*) AM4,DAM
                        AM4 = AM4 + DAM
            IF (ABS(DAM).GT..000001) GO TO 101
            T4 = T3*(RS1**2)*RS2
            P4 = P3*RS1
            P04 = P03*RS1*RS3**(G2/(G2-1.))
*           WRITE(6,*) 'M4 =',AM4,'T4 =',T4,'T04 =',T04
*           WRITE(6,*) 'Emax =',FMAX/FCRIT
*           WRITE(6,*) ' '

            END IF

*———— Check Exit Temperature:
 102  CONTINUE

            IF(T4.LT.TMAX) GOTO 120
            IF(T4.GT.TMAX) THEN
*                 WRITE(6,*) 'Exceeded Max Temp :Adjusting Mach Number'
                  GOTO 103
            END IF
 103  CONTINUE
```

```
                    RS1=(1.+G2*AM3**2)/(1+G2*AM4**2)
                    RS2=(AM4/AM3)**2
                    RS3=(1.+ 0.5*(G2-1.)*AM4**2)/(1.+ 0.5*(G2-1.)*AM3**2)
                    T04 = T03*(RS1**2)*RS2*RS3
                    T4 = T3*(RS1**2)*RS2
                    FX1 = T4-TMAX
                    FX2 = T4*2.*(1./AM4-2.*G2*AM4/(1.+G2*AM4**2))
                    DAM = - FX1/FX2
*                   write(6,*) T4,AM4,DAM
                    AM4=AM4+DAM
              IF (ABS(DAM).GT..000001) GO TO 103

 110  CONTINUE

      RS1=(1.+G2*AM3**2)/(1+G2*AM4**2)
              RS2=(AM4/AM3)**2
              RS3=(1.+ 0.5*(G2-1.)*AM4**2)/(1.+ 0.5*(G2-1.)*AM3**2)
              T4 = T3*(RS1**2)*RS2
              P4 = P3*RS1
              P04 = P03*RS1*RS3**(G2/(G2-1.))
              T04 = T03*(RS1**2)*RS2*Rs3
   FMAX = (T04 - T03)/(ETAB*QR/CP2 - T04)

*             WRITE(6,*) 'M4 =',AM4
*             WRITE(6,*) 'T4 =',T4,'T04 =',T04
*             WRITE(6,*) 'Emax =',FMAX/FCRIT
*             WRITE(6,*) ' '


 120  CONTINUE
              F = FMAX
              IF (F.LT.FGIVEN) THEN
*             WRITE(6,*) 'Excess Hydrogen, adjust mixture properties:'
                    MWEIGHT=1./((1.- FGIVEN)/28.97 + FGIVEN/2.016)
                    RG2 = 1545.43*32.17/MWEIGHT
                    IF (KOUNTER.EQ.1) GOTO 100
                    GOTO 122
              END IF
              IF (F.GT.FGIVEN) THEN
*                   WRITE(6,*) 'Less fuel flow than can be accommodated,'
*                   WRITE(6,*) 'adjust exit Mach number:'
                    F = FGIVEN
                    FMAX = FCRIT
                    T04 = (T03 + F*ETAB*QR/CP2)/(1. + F)
 121  CONTINUE
                    RS1=(1.+G2*AM3**2)/(1+G2*AM4**2)
                    RS2=(AM4/AM3)**2
                    RS3=(1.+ 0.5*G2M*AM4**2)/(1.+ 0.5*G2M*AM3**2)
                    S1 = RS1*RS1*RS2*RS3 - T04/T03
                    S2 = G2M*AM4/(1.+ 0.5*G2M*AM4**2)
                    S2 = S2 + 2./AM4 - 4*G2*AM4/(1.+ G2*AM4**2)
                    S2 = RS1*RS1*RS2*RS3*S2
                          DAM = - S1/S2
*                         write(6,*) AM4,DAM
```

```
                         AM4 = AM4 + DAM
                 IF (ABS(DAM).GT..000001) GO TO 121
122     CONTINUE
                     RS1=(1.+G2*AM3**2)/(1+G2*AM4**2)
                     RS2=(AM4/AM3)**2
                     RS3=(1.+ 0.5*G2M*AM4**2)/(1.+ 0.5*G2M*AM3**2)
                     T04 = T03*(RS1**2)*RS2*Rs3
      FMAX = (T04 - T03)/(ETAB*QR/CP2 - T04)
                 T4 = T3*(RS1**2)*RS2
                 P4 = P3*RS1
                 P04 = P03*RS1*RS3**(G2/(G2-1.))
                 END IF
*                WRITE(6,*) 'Final Output:'
*                WRITE(6,*) 'M4 = ',AM4,'T4 = ',T4,'ER = ',FMAX/FCRIT


*______________________________________
      RETURN
      END
*
*
```

```
* *-----------------------------------------------------------------
      GM = G - 1.
      GP = G + 1.

*----------- INITIAL GUESS FOR AM
      AM = SQRT((GP/GM)**GP * AR**GM)

*----------- FIXED POINT ITERATION FOR AM
      KI  = 0
  100 KI  = KI + 1
      AML = AM
   AM = SQRT((GP*(AR*AM)**(2.*GM/GP) - 2.)/GM) IF(ABS(AM-AML).GT.1.E-6) GO TO
100

*----------- CONVERGED
      RETURN
      END
*
*
      SUBROUTINE MOCA(Kicker) *------------------------------------------
---------------------
*
*   This subroutine computes the flow between an axisymmetric
*   or 2D solid body and a shock.  Initial conditions are those
*   of the exhaust of a SCRAMjet with Mach number near unity.
*   Freestream conditions are those downstream of the expansion
*   from the cowl lip of the engine.
*
*-----------------------------------------------------------------

PARAMETER    (IM=501,JM=15)

   COMMON/CBSPA/ AAX(IM),BAX(IM),CAX(IM),AAY(IM),BAY(IM),CAY(IM), 1
      AAC(IM),BAC(IM),CAC(IM),NPA
COMMON/DEPVR/ RHO(IM,JM),VT(IM,JM),T(IM,JM),P(IM,JM),
   1          TH(IM,JM),AM(IM,JM),WS(IM),H0(IM,JM),P0(IM,JM)
      COMMON/GASPR/ G,RG,GM,GP,CP
      COMMON/GRIDS/ X(IM),Y(IM,JM)
      COMMON/GRIDF/ XF(IM),YF(IM),CF(IM),ILF
      COMMON/GRIDA/ XA(IM),YA(IM),CA(IM),ILA
      COMMON/INFTY/ AMINF,PINF,TINF,ALPHA

      DIMENSION   W(JM)

      CHARACTER*1  ZZZ

*--------------- USEFUL STUFF ---------------
      PI  = 3.141592654
      CAL = COS(ALPHA)
         Kicker = 0
      JSL = JM - 2
      JSS = JSL + 1
      T0INF = TINF*(1. + .5*GM*AMINF**2)
      H0INF = CP*T0INF
```

```
      RHINF = PINF/(RG*TINF)
      VINF  = AMINF*SQRT(G*RG*TINF)

*____________________________________________________________
*                MAIN LOOP
*____________________________________________________________*
 OPEN(2,FILE='moca.dat',STATUS='UNKNOWN')
   I = ILF + 1
*   WRITE(2,*) X(I),Y(I,1),Y(I,JM),Y(I,JM)
   DXF = .015
   EXF = 1.1
   100     I = I + 1

*---------- DETERMINE THE STEP SIZE, DX ----------DXF = DXF*EXF
   IF(DXF.GT..5) DXF = .25
*-------- LIMITATION AT GROUND PLANE
DY = (Y(I-1,JSL) - Y(I-1,1))/(JSL - 1.) EM = ATAN2(1.,SQRT(AM(I-1,1)**2 - 1.))
DXW = DXF*DY/(TAN(EM + TH(I-1,2)) - TAN(TH(I-1,2)))

*-------- LIMITATION AT SHOCK
*    EM = ATAN2(1.,SQRT(AM(I-1,JM)**2 - 1.))
*    DXS = DXF*DY/(TAN(EM - TH(I-1,JM)) + TAN(WS(I-1)))

*    DX = DXS
*    IF(DXW.LT.DXS) DX = DXW
   DX = DXW
*  WRITE(6,*),' DX = ',DXW

*---------- SET Y-STATIONS AT THIS I ----------
   XX   = X(I-1) + DX
   IF(XX.GE.XA(NPA)) THEN
          XX= XA(NPA)
          DX= XA(NPA) - X(I-1)
          ILA    = I
          Y(I,1) = YA(NPA)
          CAW   = CA(NPA)
   ELSE
          CALL INTRPC(XA,XX,AAX,BAX,CAX,NPA,IZ,TT)
Y(I,1)  = YA(IZ) + ((AAY(IZ)*TT + BAY(IZ))*TT + CAY(IZ))*TT CAW          =  CA(IZ)  +
((AAC(IZ)*TT + BAC(IZ))*TT + CAC(IZ))*TT
   ENDIF
   X(I)    = XX
   Y(I,JSL) = Y(I-1,JSL) + DX*TAN(TH(I-1,JSL))
   Y(I,JSS) = Y(I,JSL)
   Y(I,JM)  = Y(I-1,JM) + DX*TAN(WS(I-1))

   DY = (Y(I,JSL) - Y(I,1))/(JSL - 1.)
   DO 120 J = 2,JSL-1
          X(I)    = X(I)
          Y(I,J) = Y(I,J-1) + DY
  120     CONTINUE

*____________________________________________________________
*              CONDITIONS AT THE WALL
```

```
*------------------------------------------------------------------TH(I,1)        =
ATAN2((Y(I,1)-Y(I-1,1)),DX)

*-------- INITIAL GUESS,VALUES AT LAST WALL STATION
   P(I,1)  = P(I-1,1)
   AM(I,1) = AM(I-1,1)
   RHO(I,1) = RHO(I-1,1)
   H0(I,1) = H0(I-1,1)
   VT(I,1) = VT(I-1,1)
   PW     = P(I-1,1)

   AM1   = AM(I-1,1)
   TH1   = TH(I-1,1)

*------------ ITERATION LOOP FOR THE WALL --------
*      WRITE(6,*),' Iteration loop for wall...AM1 = ',AM1
       KW = 0
*-------- LOCATE YC1
  130 CONTINUE
       KW = KW + 1
       PWL = P(I,1)
EMW = ATAN2(1.,SQRT(AM(I,1)**2 - 1.))
EMU = .5*(EMW + ATAN2(1.,(SQRT(AM1**2 - 1.)))) PHI = EMU - .5*(TH(I,1) + TH1)
       YC1 = Y(I,1) + DX*TAN(PHI)
DC1 = SQRT(DX**2 + (Y(I,1) - YC1)**2)

       DO 140 J = 2,JSL
          J1 = J - 1
IF(YC1.LT.Y(I-1,J)) GO TO 145
  140 CONTINUE

*-------- COMPUTE PRESSURE & OTHER STUFF AT YC1 *                Use  4  pt
Lagrange interpolation
  145 CONTINUE
       JPM = 0
       IF(J1.EQ.1) JPM = 1
       IF(J1.EQ.JM-1) JPM = -1

       JB = J1 - 1 + JPM
       JC = J1 + JPM
       JD = J1 + 1 + JPM
       JE = J1 + 2 + JPM

       YB = Y(I-1,JB)
       YC = Y(I-1,JC)
       YD = Y(I-1,JD)
       YE = Y(I-1,JE)

YF1=(YC1-YC)*(YC1-YD)*(YC1-YE)/((YB-YC)*(YB-YD)*(YB-YE)) YF2=(YC1-YB)*(YC1-
YD)*(YC1-YE) / ((YC-YB)*(YC-YD)*(YC-YE)) YF3 = (YC1-YB)*(YC1-YC)*(YC1-YE) /
((YD-YB)*(YD-YC)*(YD-YE)) YF4 = (YC1-YB)*(YC1-YC)*(YC1-YD) / ((YE-YB)*(YE-
YC)*(YE-YD))

       TH1 = TH(I-1,JB)*YF1 + TH(I-1,JC)*YF2 +
```

```
     1                 TH(I-1,JD)*YF3 +  TH(I-1,JE)*YF4
       AM1 =  AM(I-1,JB)*YF1 +  AM(I-1,JC)*YF2 +
     1                 AM(I-1,JD)*YF3 +  AM(I-1,JE)*YF4
       VT1 =  VT(I-1,JB)*YF1 +  VT(I-1,JC)*YF2 +
     1                 VT(I-1,JD)*YF3 +  VT(I-1,JE)*YF4
       RH1 = RHO(I-1,JB)*YF1 + RHO(I-1,JC)*YF2 +
     1                RHO(I-1,JD)*YF3 + RHO(I-1,JE)*YF4
       P1  =   P(I-1,JB)*YF1 +   P(I-1,JC)*YF2 +
     1                  P(I-1,JD)*YF3 +   P(I-1,JE)*YF4

       BT1 = SQRT(AM1**2 - 1.)
       BRV1 = BT1/(RH1*VT1**2)
       RHS1 = SIN(TH1)/(AM1*YC1)

*---------- AVERAGES BETWEEN YC1 & Y(I,1)
BRV = .5*(BRV1 + SQRT(AM(I,1)**2 - 1.)/(RHO(I,1)*VT(I,1)**2)) RHS = -.5*DC1*(RHS1 +
SIN(TH(I,1))/(AM(I,1)*Y(I,1)))
       RHS = 0.

*---------- GET PRESSURE
       P(I,1) = P1 + (RHS + (TH(I,1) - TH1))/BRV

*---------- GET DENSITY FROM SOUND SPEED EQN
AA = .5*G*(P(I,1)/RHO(I,1) + P(I-1,1)/RHO(I-1,1))
       RHO(I,1) = RHO(I-1,1) + (P(I,1) - P(I-1,1))/AA

*---------- GET VT FROM STAGNATION ENTHALPY
    VT(I,1) = SQRT(2.*(H0(I,1) - G*P(I,1)/(GM*RHO(I,1))))

*---------- MACH NUMBER
AM(I,1) = VT(I,1)/SQRT(G*P(I,1)/RHO(I,1))
    AMW    = AM(I,1)

*-------------- CHECK TO SEE IF CONVERGED
    DP = ABS(PWL - P(I,1))
*   WRITE(6,*),'  dp = ',DP
    IF(DP.GT..01) GO TO 130
*   WRITE(6,*),'  Done with wall...'

*------------------------------------------------------------------------*
              DO INTERIOR POINTS
*------------------------------------------------------------------------DO 280 J =
2,JSL-1

*-------------- INITIAL GUESS FOR THIS J
          P(I,J)  = P(I-1,J)
          AM(I,J)  = AM(I-1,J)
          RHO(I,J) = RHO(I-1,J)
          H0(I,J)  = H0(I-1,J)
          VT(I,J)  = VT(I-1,J)
          TH(I,J)  = TH(I-1,J)

          AM1  = AM(I,J)
          TH1  = TH(I,J)
```

```
          AM2  = AM(I,J)
          TH2  = TH(I,J)
          AMS  = AM(I,J)
          THS  = TH(I,J)

          KI = 0

*---------------- ITERATION LOOP AT THIS POINT ------------210            CON-
TINUE
          KI = KI + 1
          PL = P(I,J)

*-------- LOCATE YC1
EM  = ATAN2(1.,SQRT(AM(I,J)**2 - 1.))
EMU = .5*(EM + ATAN2(1.,(SQRT(AM1**2 - 1.)))) PHI = EMU - .5*(TH(I,J) + TH1)
          YC1 = Y(I,J) + DX*TAN(PHI)
DC1 = SQRT(DX**2 + (Y(I,J) - YC1)**2)

*---------- FIND NEAREST J-INDEX BELOW YC1
          DO 220 JZ = 2,JSL
             J1 = JZ - 1
          IF(YC1.LT.Y(I-1,JZ)) GO TO 225 220      CONTINUE

*-------- COMPUTE PRESSURE & OTHER STUFF AT YC1 *          Using 4 pt Lagrange
interpolation
   225       CONTINUE
          JPM = 0
          IF(J1.EQ.1) JPM = 1
IF(J1.EQ.JSL-1) JPM = -1

          JB = J1 - 1 + JPM
          JC = J1 + JPM
          JD = J1 + 1 + JPM
          JE = J1 + 2 + JPM

          YB = Y(I-1,JB)
          YC = Y(I-1,JC)
          YD = Y(I-1,JD)
          YE = Y(I-1,JE)

YF1=(YC1-YC)*(YC1-YD)*(YC1-YE)/((YB-YC)*(YB-YD)*(YB-YE)) YF2=(YC1-YB)*(YC1-
YD)*(YC1-YE) / ((YC-YB)*(YC-YD)*(YC-YE)) YF3 = (YC1-YB)*(YC1-YC)*(YC1-YE) /
((YD-YB)*(YD-YC)*(YD-YE)) YF4 = (YC1-YB)*(YC1-YC)*(YC1-YD) / ((YE-YB)*(YE-
YC)*(YE-YD))

       TH1 = TH(I-1,JB)*YF1 + TH(I-1,JC)*YF2 +
    1          TH(I-1,JD)*YF3 + TH(I-1,JE)*YF4
       AM1 = AM(I-1,JB)*YF1 + AM(I-1,JC)*YF2 +
    1          AM(I-1,JD)*YF3 + AM(I-1,JE)*YF4
       VT1 = VT(I-1,JB)*YF1 + VT(I-1,JC)*YF2 +
    1          VT(I-1,JD)*YF3 + VT(I-1,JE)*YF4
       RH1 = RHO(I-1,JB)*YF1 + RHO(I-1,JC)*YF2 +
    1                RHO(I-1,JD)*YF3 + RHO(I-1,JE)*YF4
       P1  = P(I-1,JB)*YF1 + P(I-1,JC)*YF2 +
```

```
1              P(I-1,JD)*YF3 +   P(I-1,JE)*YF4

          BT1  = SQRT(AM1**2 - 1.)
          BRV1 = BT1/(RH1*VT1**2)
*          RHS1 = SIN(TH1)/(AM1*YC1)
          RHS1 = 0.

*—————— LOCATE YC2
EMU = .5*(EM + ATAN2(1.,(SQRT(AM2**2 - 1.)))) PHI = EMU + .5*(TH(I,J) + TH2)
          YC2 = Y(I,J) - DX*TAN(PHI)
DC2 = SQRT(DX**2 + (Y(I,J) - YC2)**2)

          DO 230 JZ = 2,JSL
            J2 = JZ - 1
          IF(YC2.LT.Y(I-1,JZ)) GO TO 235 230       CONTINUE

*———— COMPUTE PRESSURE & OTHER STUFF AT YC2 *                    Use   4   pt
Lagrange interpolation
  235     CONTINUE
          JPM = 0
          IF(J2.EQ.1) JPM = 1
IF(J2.EQ.JSL-1) JPM = -1

          JB = J2 - 1 + JPM
          JC = J2 + JPM
          JD = J2 + 1 + JPM
          JE = J2 + 2 + JPM

          YB = Y(I-1,JB)
          YC = Y(I-1,JC)
          YD = Y(I-1,JD)
          YE = Y(I-1,JE)

YF1=(YC2-YC)*(YC2-YD)*(YC2-YE)/((YB-YC)*(YB-YD)*(YB-YE)) YF2=(YC2-YB)*(YC2-
YD)*(YC2-YE) / ((YC-YB)*(YC-YD)*(YC-YE)) YF3 = (YC2-YB)*(YC2-YC)*(YC2-YE) /
((YD-YB)*(YD-YC)*(YD-YE)) YF4 = (YC2-YB)*(YC2-YC)*(YC2-YD) / ((YE-YB)*(YE-
YC)*(YE-YD))

       TH2 = TH(I-1,JB)*YF1 + TH(I-1,JC)*YF2 +
  1              TH(I-1,JD)*YF3 + TH(I-1,JE)*YF4
       AM2 = AM(I-1,JB)*YF1 + AM(I-1,JC)*YF2 +
  1              AM(I-1,JD)*YF3 + AM(I-1,JE)*YF4
       VT2 = VT(I-1,JB)*YF1 + VT(I-1,JC)*YF2 +
  1              VT(I-1,JD)*YF3 + VT(I-1,JE)*YF4
       RH2 = RHO(I-1,JB)*YF1 + RHO(I-1,JC)*YF2 +
  1                    RHO(I-1,JD)*YF3 + RHO(I-1,JE)*YF4
       P2  =  P(I-1,JB)*YF1 +  P(I-1,JC)*YF2 +
  1                     P(I-1,JD)*YF3 +  P(I-1,JE)*YF4

          BT2 = SQRT(AM2**2 - 1.)
          BRV2 = BT2/(RH2*VT2**2)
*          RHS2 = SIN(TH2)/(AM2*YC2)
          RHS2 =   0.
```

```
*————————— LOCATE YS
EMU = .5*(EM + ATAN2(1.,(SQRT(AMS**2 - 1.)))) PHI = .5*(TH(I,J) + THS)
YS = Y(I,J) - DX*TAN(PHI)
          DS = SQRT(DX**2 + (Y(I,J) - YS)**2)

          DO 240 JZ = 2,JSL
             JS = JZ - 1
          IF(YS.LT.Y(I-1,JZ)) GO TO 245 240        CONTINUE

*———— COMPUTE PRESSURE & OTHER STUFF AT YS *          Use 4 pt Lagrange in-
terpolation
  245     CONTINUE
          JPM = 0
IF(JS.EQ.1) JPM = 1 IF(JS.EQ.JSL-1) JPM = -1

          JB = JS - 1 + JPM
          JC = JS + JPM
          JD = JS + 1 + JPM
          JE = JS + 2 + JPM

          YB = Y(I-1,JB)
          YC = Y(I-1,JC)
          YD = Y(I-1,JD)
          YE = Y(I-1,JE)

YF1 = (YS-YC)*(YS-YD)*(YS-YE)/((YB-YC)*(YB-YD)*(YB-YE)) YF2 = (YS-YB)*(YS-
YD)*(YS-YE)/((YC-YB)*(YC-YD)*(YC-YE)) YF3 = (YS-YB)*(YS-YC)*(YS-YE)/((YD-
YB)*(YD-YC)*(YD-YE)) YF4 = (YS-YB)*(YS-YC)*(YS-YD)/((YE-YB)*(YE-YC)*(YE-YD))

THS = TH(I-1,JB)*YF1 + TH(I-1,JC)*YF2 +
   1                 TH(I-1,JD)*YF3 + TH(I-1,JE)*YF4
AMS = AM(I-1,JB)*YF1 + AM(I-1,JC)*YF2 +
   1                 AM(I-1,JD)*YF3 + AM(I-1,JE)*YF4
VTS = VT(I-1,JB)*YF1 + VT(I-1,JC)*YF2 +
   1                 VT(I-1,JD)*YF3 + VT(I-1,JE)*YF4
RHOS = RHO(I-1,JB)*YF1 + RHO(I-1,JC)*YF2 +
   1                 RHO(I-1,JD)*YF3 + RHO(I-1,JE)*YF4
H0S = H0(I-1,JB)*YF1 + H0(I-1,JC)*YF2 +
   1                 H0(I-1,JD)*YF3 + H0(I-1,JE)*YF4
          PS = P(I-1,JB)*YF1 + P(I-1,JC)*YF2 +
   1                  P(I-1,JD)*YF3 + P(I-1,JE)*YF4

*————————— SOLVE FOR P(I,J) ————————
*   Note: Eq. 2 & 3
BRV = SQRT(AM(I,J)**2 - 1.)/(RHO(I,J)*VT(I,J)**2)
*         RHS = SIN(TH(I,J))/(AM(I,J)*Y(I,J))
          RHS = 0.

          BRV1 = .5*(BRV + BRV1)
*         RHS1 = .5*DC1*(RHS + RHS1)
          RHS1 = 0.

          BRV2 = .5*(BRV + BRV2)
*         RHS2 = .5*DC2*(RHS + RHS2)
```

```
      RHS2 = 0.

      DTH = TH1 - TH2
      P(I,J) = (BRV1*P1 + BRV2*P2 - RHS1 - RHS2 - DTH)/(BRV1 + BRV2)

*-------------- SOLVE FOR TH(I,J) --------------
*   Note: Eq. 2
      TH(I,J) = TH1 + BRV1*(P(I,J) - P1) + RHS1

*-------------- SOLVE FOR RHO(I,J) --------------
*   Note: Eq. 1
      AA = .5*G*(P(I,J)/RHO(I,J) + PS/RHOS)
      RHO(I,J) = RHOS + (P(I,J) - PS)/AA

*-------------- SOLVE FOR H0(I,J) --------------
*   Note: On same streamline as point f.
      H0(I,J) = H0S

*-------------- SOLVE FOR VT(I,J) --------------
*   Note: Eq. 4
      VT(I,J) = SQRT(2.*(H0(I,J) - G*P(I,J)/(GM*RHO(I,J))))

*-------------- SOLVE FOR OTHER STUFF --------------AM(I,J) = VT(I,J)/
SQRT(G*P(I,J)/RHO(I,J))

*---------- CHECK CONVERGENCE AT THIS POINT ----------
      DP = ABS(P(I,J) - PL)
      IF(DP.GT..001) GO TO 210

*--------------------------------------------------
*        END OF MAIN LOOP
*--------------------------------------------------280   CONTINUE
*  WRITE(6,*),'  finished interior points at this station...'

*--------------------------------------------------
*              POINTS AT SLIPLINE
*--------------------------------------------------*-------------- INI-
TIAL GUESS FOR SLIPLINE
   TH(I,JSL)  = TH(I-1,JSL)
   AM(I,JSL)  = AM(I-1,JSL)
   VT(I,JSL)  = VT(I-1,JSL)
   RHO(I,JSL) = RHO(I-1,JSL)
   H0(I,JSL)  = H0(I-1,JSL)
   P(I,JSL)   = P(I-1,JSL)

   AM2 = AM(I-1,JSL)
   TH2 = TH(I-1,JSL)

   TH(I,JSS)  = TH(I-1,JSS)
   AM(I,JSS)  = AM(I-1,JSS)
   VT(I,JSS)  = VT(I-1,JSS)
   RHO(I,JSS) = RHO(I-1,JSS)
   H0(I,JSS)  = H0(I-1,JSS)
   P(I,JSS)   = P(I-1,JSS)
```

```
AM1 = .5*(AM(I-1,JSS) + AM(I-1,JM)) TH1 = .5*(TH(I-1,JSS) + TH(I-1,JM))

*————— GET STUFF AT J = JSL
    KI = 0

count = 0
   300     CONTINUE
             count = count + 1
             IF (count.GT.30.) THEN
WRITE(6,*)'slipline did not converge' Kicker = 5
GOTO 3331
             END IF

       KI = KI + 1
*      WRITE(6,*),' KI       = ',KI
*      WRITE(6,*),' AM(I,JSL) = ',AM(I,JSL)
*      WRITE(6,*),' AM2      = ',AM2

*————— LOCATE YC2
EML = ATAN2(1.,SQRT(AM(I,JSL)**2 - 1.))
EMU = .5*(EML + ATAN2(1.,(SQRT(AM2**2 - 1.)))) PHI = EMU + .5*(TH(I,JSL) + TH2)
       YC2 = Y(I,JSL) - DX*TAN(PHI)
DC2 = SQRT(DX**2 + (Y(I,JSL) - YC2)**2)

*————— FIND J-LOCATION OF YC2
       DO 310 JZ = 2,JSL
J2 = JZ - 1 IF(YC2.LT.Y(I-1,JZ)) GO TO 315
   310 CONTINUE

*————— GET VALUES AT YC2
*   Use 4 pt Lagrange interpolation
   315 CONTINUE
       JPM = 0
IF(J2.EQ.JSL-1) JPM = -1

       JB = J2 - 1 + JPM
       JC = J2 + JPM
       JD = J2 + 1 + JPM
       JE = J2 + 2 + JPM

       YB = Y(I-1,JB)
       YC = Y(I-1,JC)
       YD = Y(I-1,JD)
       YE = Y(I-1,JE)

YF1=(YC2-YC)*(YC2-YD)*(YC2-YE)/((YB-YC)*(YB-YD)*(YB-YE)) YF2=(YC2-YB)*(YC2-
YD)*(YC2-YE)/((YC-YB)*(YC-YD)*(YC-YE)) YF3=(YC2-YB)*(YC2-YC)*(YC2-YE)/((YD-
YB)*(YD-YC)*(YD-YE)) YF4 = (YC2-YB)*(YC2-YC)*(YC2-YD)/((YE-YB)*(YE-YC)*(YE-
YD))

       TH2 = TH(I-1,JB)*YF1 + TH(I-1,JC)*YF2 +
    1                TH(I-1,JD)*YF3 + TH(I-1,JE)*YF4
       AM2 = AM(I-1,JB)*YF1 + AM(I-1,JC)*YF2 +
    1                AM(I-1,JD)*YF3 + AM(I-1,JE)*YF4
```

```
      VT2 =  VT(I-1,JB)*YF1 +  VT(I-1,JC)*YF2 +
     1                 VT(I-1,JD)*YF3 +  VT(I-1,JE)*YF4
      RH2 = RHO(I-1,JB)*YF1 + RHO(I-1,JC)*YF2 +
     1                RHO(I-1,JD)*YF3 + RHO(I-1,JE)*YF4
      P2  =   P(I-1,JB)*YF1 +   P(I-1,JC)*YF2 +
     1                 P(I-1,JD)*YF3 +   P(I-1,JE)*YF4

*     WRITE(6,*),'    AM2 = ',AM2
      BT2  = SQRT(AM2**2 - 1.)
      BRV2 = BT2/(RH2*VT2**2)
*      RHS2 = SIN(TH2)/(AM2*YC2)
      RHS2 =    0.

*-------- GET AVERAGED VALUES BETWEEN YC2 AND SLIPLINE AT I BRV =
SQRT(AM(I,JSL)**2 - 1.)/(RHO(I,JSL)*VT(I,JSL)**2) BRV2 = .5*(BRV2 + BRV)
*       RHS  = SIN(TH(I,JSL))/(AM(I,JSL)*Y(I,JSL))
*   RHS2 = .5*DC2*(RHS + RHS2)
   RHS2 = 0.

*------------ COMPUTE CORRESPONDING VALUE OF P(I,JSL) P(I,JSL) = P2 + (RHS2 -
(TH(I,JSL) - TH2))/BRV2

*------------ GET OTHER VARIABLES AT (I,JSL)
AA = .5*G*(P(I,JSL)/RHO(I,JSL) + P(I-1,JSL)/RHO(I-1,JSL)) RHO(I,JSL) = RHO(I-1,JSL) +
(P(I,JSL) - P(I-1,JSL))/AA VT(I,JSL) = SQRT(2.*(H0(I,JSL)-G*P(I,JSL)/(GM*RHO(I,JSL))))
AM(I,JSL) = VT(I,JSL)/SQRT(G*P(I,JSL)/RHO(I,JSL))

*--------------- NOW GET STUFF AT J = JSS
*  WRITE(6,*),'   AM(I,jss) = ',AM(I,JSS)
*-------- LOCATE YC1
   EM1 = ATAN2(1.,SQRT(AM1**2 - 1.))
EMU = .5*(ATAN2(1.,SQRT(AM(I,JSS)**2 - 1.)) + EM1)
   PHI = EMU - .5*(TH(I,JSS) + TH1)
   YC1 = Y(I,JSS) + DX*TAN(PHI)
   DC1 = SQRT(DX**2 + (Y(I,JSS) - YC1)**2)

*-------- COMPUTE PRESSURE & OTHER STUFF AT YC1
*  Using linear interpolation
   IF((I-1).EQ.(ILF+1)) THEN
         YR = 1.
   ELSEIF(YC1.GT.Y(I-1,JM)) THEN
         YR = 1.
   ELSE
   YR = (YC1 - Y(I-1,JSS))/(Y(I-1,JM) - Y(I-1,JSS)) ENDIF

TH1 = TH(I-1,JSS) + YR*(TH(I-1,JM) - TH(I-1,JSS)) AM1 = AM(I-1,JSS) + YR*(AM(I-1,JM) -
AM(I-1,JSS)) VT1 = VT(I-1,JSS) + YR*(VT(I-1,JM) - VT(I-1,JSS)) RH1 = RHO(I-1,JSS) +
YR*(RHO(I-1,JM) - RHO(I-1,JSS)) P1 = P(I-1,JSS) + YR*(P(I-1,JM) - P(I-1,JSS))

*  WRITE(6,*),'    AM1 = ',AM1
   BT1 = SQRT(AM1**2 - 1.)
   BRV1 = BT1/(RH1*VT1**2)
*  RHS1 = SIN(TH1)/(AM1*YC1)
   RHS1 = 0.
```

```
*———— GET AVERAGED VALUES BETWEEN YC1 AND SLIPLINE AT I BRV =
SQRT(AM(I,JSS)**2 - 1.)/(RHO(I,JSS)*VT(I,JSS)**2) BRV1 = .5*(BRV1 + BRV)
*   RHS = SIN(TH(I,JSS))/(AM(I,JSS)*Y(I,JSS))
*   RHS1 = -.5*DC1*(RHS + RHS1)
    RHS1 = 0.

*———————— COMPUTE CORRESPONDING VALUE OF P(I,JSS) P(I,JSS) = P1 + (RHS1 +
(TH(I,JSS) - TH1))/BRV1

*———— GET OTHER STUFF AT J = JSS
AA = .5*G*(P(I,JSS)/RHO(I,JSS) + P(I-1,JSS)/RHO(I-1,JSS)) RHO(I,JSS) = RHO(I-1,JSS) +
(P(I,JSS) - P(I-1,JSS))/AA ARG = H0(I,JSS) - G*P(I,JSS)/(GM*RHO(I,JSS))
*   WRITE(6,*),'  ARG = ',ARG
    VT(I,JSS) = SQRT(2.*(H0(I,JSS)-G*P(I,JSS)/(GM*RHO(I,JSS)))) *  WRITE(6,*),'
P(I,JSS) = ',P(I,JSS),'  P1 = ',P1
*   WRITE(6,*),' RHO(I,JSS) = ',RHO(I,JSS),'  RH1 = ',RH1
AM(I,JSS) = VT(I,JSS)/SQRT(G*P(I,JSS)/RHO(I,JSS))
*   WRITE(6,*),'   VT(I,JSS) = ',VT(I,JSS),'  VT1 = ',VT1
*   WRITE(6,*),'   AM(I,JSS) = ',AM(I,JSS),'  AM1 = ',AM1

*——————— NOW COMPARE PRESSURES BETWEEN JSL & JSS
    DP = ABS(P(I,JSL) - P(I,JSS))
*   WRITE(6,*),' DP  = ',DP,'  P(I,JSL) = ',P(I,JSL)
*   WRITE(6,*),'  Hit any key when ready...'
*   READ(5,'(A1)') ZZZZ
    IF(DP.LT..05) GO TO 330
    IF(KI.GT.40) THEN
*   WRITE(6,*),'  KI = ',KI,'...GOTO 3333ping...'
*   WRITE(6,*),' DP = ',DP,'  P(JSL) = ',P(I,JSL)
    ENDIF

*——————— NOT CONVERGED, SET UP FOR NEXT ITERATION AT SLIP LINE PA =
.6*P(I,JSL) + .4*P(I,JSS)
THL = TH2 - BRV2*(PA - P2) + RHS2 THS = TH1 + BRV1*(PA - P1) - RHS1 THA = .5*(THL
+ THS)
TH(I,JSL) = .3*TH(I,JSS) + .7*THA TH(I,JSS) = .3*TH(I,JSL) + .7*THA GO TO 300

*————————————— SLIPLINE CONVERGED ————————————330      CONTINUE
*   WRITE(6,*),' slipline converged....'

*————————————————————————————————————————————————————*
     POINT AT THE SHOCK
*————————————————————————————————————————————————————*———————— INI-
TIAL SHOCK ANGLE = LAST STATION
    WS(I) = WS(I - 1)
    AM1  = AMINF
    AM2  = AM(I-1,JM)
    TH2  = TH(I-1,JM)
    KS    = 0
              COUNTER = 0

*—— GET QUANTITIES FROM OBLIQUE SHOCK RELATIONS 410 CONTINUE
              COUNTER = COUNTER + 1
              IF (COUNTER.GT.40) THEN
```

```
WRITE(6,*) 'Nozzle Did Not Converge' Kicker = 3
                    GOTO 3331
                    END IF

    KS          = KS + 1
    Y(I,JM) = Y(I-1,JM) + DX*TAN(WS(I))
    AMN         = AMINF*SIN(WS(I))
    PR          = (2.*G*AMN**2 - GM)/GP
    DEN         = .5*(GP*AMN)**2/GM
    TR          = (1. + .5*GM*AMN**2)*(2.*G*AMN**2/GM - 1.)/DEN
    RR          = PR/TR

    PL     = PINF*PR
    RHOL = RHINF*RR
    F1     = AMN**2 - 1.
    F2     = G*AMN**2 + 1.
    F3     = (GP*AMINF*AMN)**2
    VTL  = VINF*SQRT(1. - 4.*F1*F2/F3)
    SW     = SIN(WS(I))
THL  = ATAN2(2.*F1,(TAN(WS(I))*(2.+(GP-2.*SW**2)*AMINF**2))) SWT  = SIN(WS(I) -
THL)
AML  = SQRT((GM*AMN**2 + 2.)/((2.*G*AMN**2 - GM)*SWT**2)) VT(I,JM)  = VTL
    RHO(I,JM) = RHOL
    P(I,JM)  = PL
    AM(I,JM)  = AML
    TH(I,JM)  = THL

*------------- NOW GET P(I,JM) FROM MOC -------------*------------- LOCATE
YC2
    EML = ATAN2(1.,SQRT(AML**2 - 1.))
EMU = .5*(EML + ATAN2(1.,(SQRT(AM2**2 - 1.)))) PHI = EMU + .5*(TH(I,JM) + TH2)
    YC2 = Y(I,JM) - DX*TAN(PHI)
DC2 = SQRT(DX**2 + (Y(I,JM) - YC2)**2)

*------------- GET VALUES AT YC2
*     Use Linear interpolation
    IF((I-1).EQ.(ILF+1)) THEN
            YR = 1.
    ELSE
   YR  = (YC2 - Y(I-1,JM-1))/(Y(I-1,JM)-Y(I-1,JM-1)) ENDIF

TH2 = TH(I-1,JM-1) + YR*(TH(I-1,JM)-TH(I-1,JM-1)) AM2 = AM(I-1,JM-1) + YR*(AM(I-
1,JM)-AM(I-1,JM-1)) VT2 = VT(I-1,JM-1) + YR*(VT(I-1,JM)-VT(I-1,JM-1)) RH2 = RHO(I-
1,JM-1)+ YR*(RHO(I-1,JM)-RHO(I-1,JM-1)) P2 = P(I-1,JM-1) + YR*(P(I-1,JM) - P(I-1,JM-1))

    BT2 = SQRT(AM2**2 - 1.)
    BRV2 = BT2/(RH2*VT2**2)
*    RHS2 = SIN(TH2)/(AM2*YC2)
    RHS2 = 0.

*--------- GET AVERAGED VALUES BETWEEN YC2 AND SHOCK AT I

BRV = SQRT(AM(I,JM)**2 - 1.)/(RHO(I,JM)*VT(I,JM)**2) BRV2 = .5*(BRV2 + BRV)
*    RHS = SIN(TH(I,JM))/(AM(I,JM)*Y(I,JM))
```

```
*   RHS2 = .5*DC2*(RHS + RHS2)
    RHS2 = 0.

*———————— COMPUTE CORRESPONDING VALUE OF P(I,JM) P(I,JM) = P2 + (RHS2 -
(TH(I,JM) - TH2))/BRV2

*———————— COMPARE WITH SHOCK VALUE
    DP = P(I,JM) - PL
*   WRITE(6,*),'  DP = ',DP,'   P(I,JM) = ',P(I,JM)
    IF(ABS(DP).LT..1) GO TO 450
*   WRITE(6,*),'  Hit any key when ready...'
*   READ(5,'(A1)') ZZZZ
    IF(ABS(DP).LT..01) GO TO 450

*——————— PRESSURE NOT CONVERGED,GET NEW SHOCK ANGLE & Y-LOCATION *
   Note: use is made of the shock angle,M,& pressure ratio
    PA = P(I,JM)/PINF
    F1 = .5*(GP*PA + GM)/(G*AMINF**2)
    WS(I) = .5*(WS(I) + ATAN(SQRT(F1/(1. - F1))))
*              WRITE(6,*),'pressure not converged'
    GOTO 410

*——————— PRESSURE CONVERGED,SHOCK LOCATION SET
  450      CONTINUE
*   WRITE(6,*),' X = ',X(I),'  AM(i,jm) = ',AM(I,JM),
*    1            '  AM(I,1) = ',AM(I,1)
*   WRITE(2,*) X(I),Y(I,1),Y(I,JSL),Y(I,JM)

*———————————— WRITE TO BODY SURFACE DATA FILE ————————*
    WRITE(55,*) X(I),AM(I,JM),P(I,JM),RHO(I,JM)

*———————————— GO TO NEXT I-STATION ———————————IF(X(I).LT.XA(NPA)) GO TO
100
*          CLOSE(2,STATUS='KEEP')
 3331      CONTINUE
*————————————————————————RETURN
    END
```