NASA CONTRACTOR REPORT

NASA CR-2924

NASA CR-2924

COMPUTATIONAL TECHNIQUES FOR SOLAR WIND FLOWS PAST TERRESTRIAL PLANETS - THEORY AND COMPUTER PROGRAMS

Stephen S. Stahará, Denny S. Chaussee,
Barbara C. Trudinger, and John R. Spreiter

Prepared by
NIELSEN ENGINEERING & RESEARCH, INC.
Mountain View, Calif. 94043
for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION · WASHINGTON, D. C. · NOVEMBER 1977

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR- 2924		

4. Title and Subtitle	5. Report Date
Computational Techniques for Solar Wind Flows Past Terrestrial Planets - Theory and Computer Programs	November 1977
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Stephen S. Stahara, Denny S. Chaussee, Barbara C. Trudinger, and John R. Spreiter	NEAR TR 140
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
Nielsen Engineering & Research, Inc. 510 Clyde Avenue Mountain View, California 94043	NASW-2945
	13. Type of Report and Period Covered Contractor Report June 1976-July 1977

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administration Washington, D.C. 20546	

15. Supplementary Notes

16. Abstract Theoretical analysis and the development of user-oriented computer programs were carried out for the purpose of developing computational techniques for predicting the interaction of the solar wind with terrestrial planets. The procedures are based on a single-fluid, steady, dissipationless, magnetohydrodynamic model and are appropriate for the calculation of axisymmetric, supersonic, super-Alfvénic solar wind flow past both magnetic and nonmagnetic planets. The actual calculations are implemented by an assemblage of computer codes organized into one program. These include finite-difference codes which determine the gas-dynamic solution, together with a variety of special-purpose output codes for determining and automatically plotting both flow field and magnetic field results.

Theoretical results obtained using these procedures are reported for a variety of solar wind conditions and different obstacle shapes. Comparisons are made with previous results, and new results are presented for a number of solar wind flows. The computational programs developed under this work have been documented and are presented in a general user's manual included as part of this report.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Solar Wind Flows Finite-Difference Computations Steady Flow Magnetosphere Ionosphere	UNCLASSIFIED-UNLIMITED Cat. 92

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
UNCLASSIFIED	UNCLASSIFIED	133	$6.00

*For sale by the National Technical Information Service, Springfield, Virginia 22161

TABLE OF CONTENTS

Section	Page No.
LIST OF ILLUSTRATIONS	iv
LIST OF TABLES	v
SUMMARY	1
INTRODUCTION	1
LIST OF SYMBOLS	3
ANALYSIS	7
General Considerations	7
Mathematical Representation of Solar Wind-Magneto/Ionosphere Interaction	8
Magnetic Planet - Determination of the Magnetosphere Boundary	11
Nonmagnetic Planet - Determination of the Ionosphere Boundary	16
Calculation of the Gasdynamic Flow Properties	19
Calculation of the Magnetic Field	26
RESULTS AND DISCUSSION	29
CONCLUDING REMARKS	32
APPENDIX A - COMPUTER PROGRAM USER'S MANUAL	34
APPENDIX B - LISTING OF COMPUTER PROGRAM	73
REFERENCES	107
TABLE 1	110
FIGURES 1 THROUGH 15	113

LIST OF ILLUSTRATIONS

Figure

1 Comparison of the equatorial and principal meridian traces of the magnetosphere boundary as provided by the simplified theory of equation 16.

2 Illustration of ionopause shapes for various values of the ionosphere scale height to shock standoff distance ratio H/R_o.

3 Comparison of former and present computational procedures for determining the gasdynamic flow properties of solar wind-magneto/ionopause interactions.

4 Illustration of the components of the three-dimensional magnetic field.

5 Comparison of flow properties predicted by the present implicit method with other techniques and experiment for supersonic flow past a sphere; $M_\infty = 4.926$, $\gamma = 1.4$.

6 Comparison of implicit and inverse methods for shock shape and sonic line location, and density distribution along bow shock and magnetosphere boundary for $M_\infty = 8$, $\gamma = 5/3$ flow past the rotated equatorial trace of the magnetopause.

7 Bow wave and sonic line locations for various supersonic flows past the rotated equatorial trace of the magnetopause with $\gamma = 5/3$.

8 Bow shock location for solar wind flow with $M_\infty = 8$, $\gamma = 5/3$ past various ionopause shapes.

9 Bow shock and embedded shock locations for solar wind flow with $M_\infty = 5$, $\gamma = 5/3$ past the rotated principal meridian of the magnetosphere.

10 Magnetopause pressure coefficients for the principal meridian magnetopause shapes shown in figure 9.

11 Variation of shock stand-off distance with oncoming Mach number and ratio of specific heats for various magneto/ionopause traces as determined by the present implicit procedure.

12 Shock shapes for various supersonic flows past the rotated equatorial trace of the magnetopause; combined near (blunt body) and far (marching) solutions.

13 Shock shapes for $M_\infty = 8$, $\gamma = 5/3$ flow past an ionopause shape with $H/R_o = 0.1$; combined near (blunt body) and far (marching) field solutions.

Figure

14 Streamline, density, and velocity maps for $M_\infty = 8.0$, $\gamma = 5/3$ flow past the rotated equatorial trace of the magnetopause; combined blunt body and marching flow field.

15 Contours and field line locations of the in-plane magnetic field components $(B/B_\infty)_\perp$ and $(B/B_\infty)_{\parallel}$ for $M_\infty = 8$ and $\gamma = 5/3$ flow past the rotated equatorial trace of the magnetopause.

A.1 Card input for sample case.

A.2 Abbreviated output for sample case.

A.3 Plot output for sample case.

LIST OF TABLES

Table

1 Ordinates of various magneto/ionopause shapes.

COMPUTATIONAL TECHNIQUES FOR SOLAR WIND FLOWS PAST TERRESTRIAL PLANETS — THEORY AND COMPUTER PROGRAMS

by Stephen S. Stahara, Denny S. Chaussee, Barbara C. Trudinger, and John R. Spreiter*
Nielsen Engineering & Research, Inc.

SUMMARY

Theoretical analysis and the development of user-oriented computer programs were carried out for the purpose of developing computational techniques for predicting the interaction of the solar wind with terrestrial planets. The procedures are based on a single-fluid, steady, dissipationless, magnetohydrodynamic model and are appropriate for the calculation of axisymmetric, supersonic, super-Alfvénic solar wind flow past both magnetic and nonmagnetic planets. The actual calculations are implemented by an assemblage of computer codes organized into one program. These include finite-difference codes which determine the gasdynamic solution, together with a variety of special-purpose output codes for determining and automatically plotting both flow field and magnetic field results.

Theoretical results obtained using these procedures are reported for a variety of solar wind conditions and different obstacle shapes. Comparisons are made with previous results, and new results are presented for a number of solar wind flows. The computational programs developed under this work have been documented and are presented in a general user's manual included as part of this report.

INTRODUCTION

The gasdynamic and magnetohydrodynamic calculations of solar wind flow around magnetic and nonmagnetic planets that Spreiter et al. (refs. 1-8) carried out some years ago have been and continue to be widely used in the interpretation of data measured in space around the Earth and other planets. The objective at the time those calculations were made, was to

---

*Professor, Department of Applied Mechanics, Stanford University.
Consultant, Nielsen Engineering & Research, Inc.

provide some theoretical results based on a fluid rather than particle description of the flow that might be compared with measurements in space to determine the applicability of such a description. During the intervening years, the usefulness and accuracy of those results have become so well established that they are currently being used, for instance, as one of the ways in which the magnetic field of a planet such as Mercury or Mars is estimated from measurements of fly-by or orbiting spacecraft.

In such applications, the previous calculations leave much to be desired. They have been carried out for only a limited set of conditions such as obstacle shape, oncoming Mach number, angle between free-stream flow and magnetic vectors, etc., and are presented in archival publications only in the form of plots from which results for other conditions must be determined by scaling and interpolation. The usefulness of the present theoretical model, therefore, would be considerably enhanced if similar but more detailed results for the specific conditions measured by a spacecraft would be provided by the application of a documented, user-oriented, economical computer code.

That objective has been successfully accomplished in the work reported here. Based upon the identical theoretical model employed previously (refs. 1-8), and discussed in more detail below, a new more versatile means of solution far superior in efficiency and generality to the former method is now available. Current advances in computational methods have been incorporated in these programs and, in fact, an entirely new blunt body code based on a recently published algorithm was developed to determine the flow field in the region surrounding the nose of the obstacle. The new methods confirm previous results and are readily applicable to additional cases involving different geometries and flow conditions. The entire procedure is fully automated and provides detailed flow field and magnetic field properties in a convenient output format. An automatic plotting capability for generating report-quality plots is also included.

LIST OF SYMBOLS

a	speed of sound, $(\gamma p/\rho)^{1/2}$
a_e	planetary radius of earth, 6.37×10^8 cm
A	Alfvén speed, $(B^2/4\pi\rho)^{1/2}$
$\bar{A}$	Jacobian matrix associated with IMP code, equal to $\partial\hat{E}/\partial\hat{U}$
$\underset{\sim}{B}$	magnetic field vector
B_{eq}	strength of Earth's magnetic field at geomagnetic equator, equal to 0.312 gauss
$\bar{B}$	Jacobian matrix associated with IMP code, equal to $\partial\hat{F}/\partial\hat{U}$
C_p	specific heat at constant pressure
C_v	specific heat at constant volume
D	geometric distance to magnetosphere nose, equal to $a_e(f^2B_{eq}^2/2\pi K\rho_\infty V_\infty)^{1/6}$
e	internal energy, eq. (5)
E	column matrix defined by eq. (29)
$\hat{E}$	column matrix associated with IMP code, equal to $(\xi_T U + \xi_x E + \xi_R F)/J$
f	constant associated with tangential component of Earth's magnetic dipole field at magnetopause location
F	column matrix defined by eq. (29)
$\hat{F}$	column matrix associated with IMP code, equal to $(\eta_T U + \eta_x E + \eta_R F)/J$
$\underset{\sim}{F}$	column matrix defined by eq. (34)
g	acceleration due to gravity
G	column matrix defined by eq. (29)
$\underset{\sim}{G}$	column matrix defined by eq. (34)
h	enthalpy, eq. (6)
H	local scale height of atmosphere, $\bar{R}T/\mu g$
H_t	total enthalpy, eq. (35)
$\underset{\sim}{H}$	column matrix defined by eq. (34)
J	Jacobian matrix, eq. (30)

K	constant defined by eq. (14)
$\Delta\underset{\sim}{\ell}$	vector length of elemental magnetic flux tube
M	local Mach number, $\lvert\underset{\sim}{V}\rvert/a$
M_A	local Alfvén Mach number, $\lvert\underset{\sim}{V}\rvert/A$
p	pressure
q	magnitude of velocity vector, eq. (36)
Q	dummy parameter
r	spherical radial distance
R	cylindrical radial distance
$\bar{R}$	gas constant, 8.315×10^7 ergs/gm°K
R_i	spherical radius of ionopause, eq. (25)
R_m	spherical radius of magnetopause, eqs. (16-18)
R_o	spherical distance from center of planet to ionopause nose
S	entropy
ΔS	incremental distance along streamline
T	time
(u,v,w)	velocity components associated with the (X,R,ω) coordinate directions, respectively
U	column matrix defined by eq. (29)
$\hat{U}$	column matrix associated with IMP code, equal to U/J
$\tilde{U}$	column matrix defined by eq. (34)
$\underset{\sim}{V}$	velocity vector
x	axial distance from center of planet measured positive upstream
X	axial distance from center of planet measured positive downstream
β	spherical polar angle, measured with respect to planet center, from subsolar point with respect to undisturbed solar wind direction; varies from 0 in upstream direction to π in downstream direction; eq. (25)
γ	ratio of specific heats

δ_s	local angle of bow shock wave
$(\delta_\xi, \delta_\eta)$	second-order difference operators in (ξ, η) direction
η	transformation variable; eqs. (27), (37)
θ	polar angle measured with respect to north geomagnetic pole, eq. (15)
Λ	quantity defined by eq. (26)
μ	nondimensionalized mean molecular mass, equal to 1/2 for ionized atomic hydrogen
ξ	transformation variable, eqs. (27), (37)
ρ	density
τ	transformed time, eq. (27)
ϕ	geomagnetic longitude, eq. (16)
ψ	angle between outward normal to magneto/ionosphere boundary and oncoming undisturbed solar wind, eq. (12)
ω	azimuthal angle in axis-normal plane, eq. (33)

Subscripts

n	normal direction
P	arbitrary point on streamline
R	reference quantity
t	tangential direction
o	reference quantity at subsolar point
1	conditions upstream of a discontinuity
2	conditions downstream of a discontinuity
∞	upstream undisturbed quantity
($''$, $\perp$, n)	parallel, perpendicular, and normal magnetic field components as defined in eq. (42)

Page
Intentionally
Left Blank

ANALYSIS

General Considerations

An overall description of solar wind flow past terrestrial planets, including an account of the principal features of the interplanetary solar wind and a description of the physical basis of the continuum fluid model are provided in references 3 and 6. Although details of those accounts won't be repeated here, several specific points relevant to the present work will be reviewed.

The fundamental assumption underlying the theoretical analysis of large-scale features of the interaction of the solar wind with terrestrial planets is that the average bulk properties of the flow can be described adequately by the continuum equations of magnetohydrodynamics for a single-component, perfect, dissipationless (perfect electrically conducting, inviscid, nonheat-conducting) gas. Theoretical justification of this point has not yet been established, and proof remains essentially qualitative at present. The primary justification for use of the continuum fluid model is the outstanding agreement of the qualitative results predicted on this basis with those actually measured in space. It appears that the continuum model is capable of accounting both for many of the details as well as the broad features of the observations.

The primary emphasis on development of the continuum model has been on steady flow past the Earth (refs. 3, 4), a planet with a strong magnetic field. Initial applications to nonmagnetic planets have also been made (ref. 5). These particular applications are the ones toward which this study is directed.

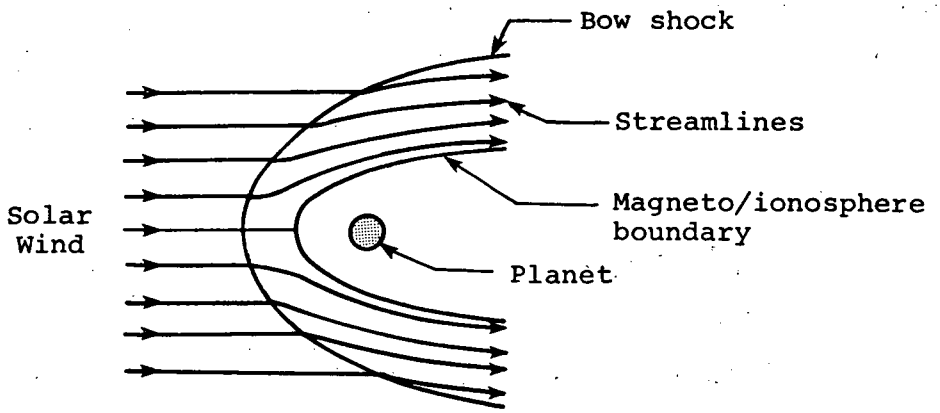
The following subsection provides the mathematical basis of the interaction of the solar wind with planetary magneto/ionospheres. The governing differential equations as well as relations for discontinuities present in the flow are discussed. The next two subsections provide the determination of the magnetosphere and ionosphere boundary shapes. The following two subsections discuss the calculation of the gasdynamic and magnetic field properties.

Mathematical Representation of Solar Wind-Magneto/Ionosphere Interaction

The differential equations governing the motion of the solar wind are the magnetohydrodynamic equations for steady flow of a dissipationless perfect gas and can be written as follows

$$\left.\begin{aligned} &\nabla\cdot\rho\underset{\sim}{V} = 0 \\ &\rho(\underset{\sim}{V}\cdot\nabla)\underset{\sim}{V} + \nabla p = -(1/4\pi)\underset{\sim}{B} \times \underset{\sim}{\text{Curl}}\ \underset{\sim}{B} \\ &\underset{\sim}{\text{Curl}}(\underset{\sim}{B} \times \underset{\sim}{V}) = 0 \\ &\text{div}\ \underset{\sim}{B} = 0 \\ &(\underset{\sim}{V}\cdot\nabla)S = 0 \\ &S - S_o = C_v \ell n \frac{p/p_o}{(\rho/\rho_o)^{\gamma}} \end{aligned}\right\} \qquad (1)$$

and apply in the region exterior to the magneto/ionosphere boundary, as shown in the sketch below.



In these equations, ρ, p, S, and $\underset{\sim}{V}$ refer to the density, pressure, entropy, and velocity of the gas, $\underset{\sim}{B}$ denotes the magnetic field, and $\gamma = C_p/C_v$ is the ratio of specific heats, equal to 5/3 for a monatomic gas. The temperature T is related to the pressure and density by the equation of state for a perfect gas

$$p = \rho\, \bar{R}\, T/\mu \qquad (2)$$

in which $\overline{R} = (C_p - C_v)\mu = 8.315\times10^7$ ergs/gm°K is the universal gas constant, and μ is the mean molecular mass nondimensionalized so that $\mu = 16$ for atomic oxygen. For fully ionized hydrogen, μ is thus 1/2. The magnetic field $\underset{\sim}{B}$ is expressed in terms of gaussian units.

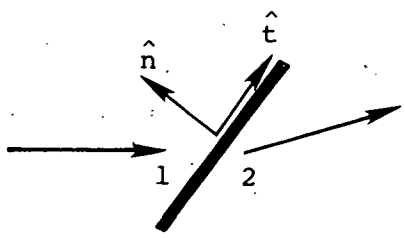
We note that it is fully equivalent and convenient for some purposes to replace the entropy equation $(\underset{\sim}{V}\cdot\nabla)S = 0$ in equation (1) by the following energy equation:

$$\mathrm{div}[\rho\underset{\sim}{V}(\tfrac{1}{2}V^2 + h) + \underset{\sim}{V}B^2/4\pi - (\underset{\sim}{B}\cdot\underset{\sim}{V})\underset{\sim}{B}/4\pi] = 0 \qquad (3)$$

Because of the omission of dissipative terms in equation (1), surfaces of discontinuity, such as the bow shock and the magneto/ionosphere boundary indicated in the sketch above, may develop in the flow. Across these surfaces, continuous solutions of the dissipationless differential equations cease to exist. The flow is no longer governed solely by the differential equations (1), but must be supplemented by additional considerations. Mass, momentum, magnetic flux, and energy must be conserved across these surfaces, and these conditions lead to the following relations which relate quantities on the two sides of any such discontinuity:

$$\left.\begin{aligned}
&[\rho V_n] = 0\\
&[\rho V_n \underset{\sim}{V} + (p + B^2/8\pi)\hat{n} - B_n\underset{\sim}{B}/4\pi] = 0\\
&[B_n\underset{\sim}{V} - \underset{\sim}{B}V_n] = 0\\
&[B_n] = 0\\
&[\rho V_n(h + V^2/2) + V_nB^2/4\pi - (\underset{\sim}{B}\cdot\underset{\sim}{V})B_n/4\pi] = 0
\end{aligned}\right\} \qquad (4)$$

Here, $(\hat{n}, \hat{t})$ denote unit vectors normal and tangential to the discontinuity surface, as sketched below,



The basis for important simplifying approximations to the governing tangential discontinuity conditions

$$V_n = B_n = [p + B^2/8\pi] = 0$$
$$[\underset{\sim}{V}_t] \neq 0; \quad [\underset{\sim}{B}_t] \neq 0; \quad [\rho] \neq 0 \tag{9}$$

which apply at the Earth's magnetosphere boundary are that the gas pressure p is much less than the magnetic pressure $B^2/8\pi$ in the confined region interior to the magnetopause, and that p is much greater than $B^2/8\pi$ in the exterior flow region around the forward part of the magnetopause. Consequently, the boundary condition to be satisfied by the exterior flow at the magnetopause can be satisfactorily approximated by the limiting case of the tangential discontinuity conditions in equation (9) in which there is a vacuum ($p = \rho = 0$) on one side, and no magnetic field on the other. Thus, the condition $[p + B^2/8\pi] = 0$ becomes

$$(B^2/8\pi)_{mag} = (p)_{flow} \tag{10}$$

while the equations governing the magnetic field interior to the magnetopause are

$$\text{div } \underset{\sim}{B} = 0, \quad \text{Curl } \underset{\sim}{B} = 0 \tag{11}$$

Furthermore, it has been shown (refs. 3, 6) that the gas pressure of the flow on the forward portion of the magnetosphere boundary — $(p)_{flow}$ in equation (10) — can be approximated adequately by the Newtonian formula

$$p = p_{st} \cos^2 \psi \tag{12}$$

Here, ψ is the angle between the outward normal to the magnetosphere boundary and the flow direction of the oncoming undisturbed solar wind, and p_{st} is the stagnation or ram pressure exerted on the nose of the magnetosphere and is given by

$$p_{st} = K \rho_\infty V_\infty^2 \tag{13}$$

In this relation, K is a constant usually taken as one, but whose actual value is

$$K = \frac{1}{\gamma}\left[\frac{(\gamma+1)^{(\gamma+1)}}{\gamma-(\gamma-1)/2M_\infty^2}\right]^{\frac{1}{\gamma-1}} \tag{14}$$

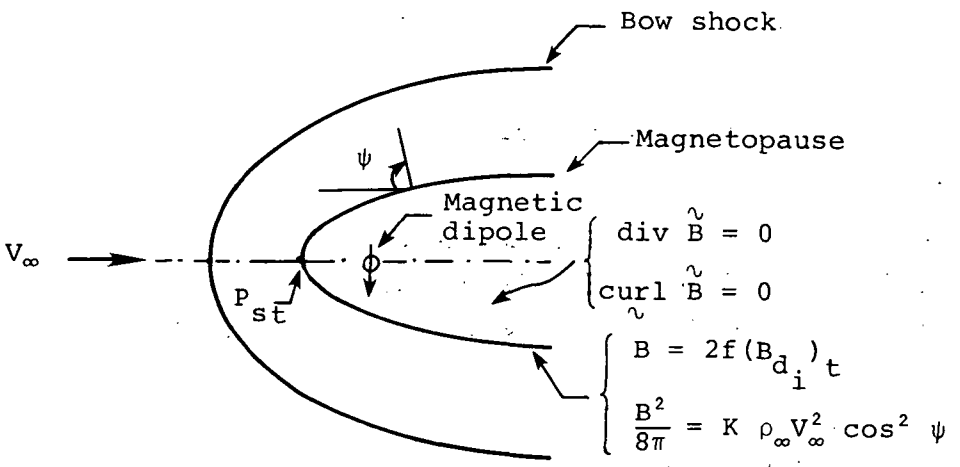
For the high Mach number flows typical of solar wind conditions, K approaches 0.844 for $\gamma = 2$ and 0.881 for $\gamma = 5/3$. The important implication associated with the introduction of the Newtonian approximation is that the calculation of the shape of the magnetosphere boundary decouples from the calculation of the external flow.

Implementation of the above analysis requires a representation of the Earth's magnetic field. This is provided with sufficient accuracy for the present application by a magnetic dipole located at the center of the Earth and given by

$$\underset{\sim}{B} = -B_{eq}\,(a_e/r)^3(\sin\theta\,\hat{e}_\theta + 2\cos\theta\,\hat{e}_r) \tag{15}$$

where $B_{eq} = 0.312$ gauss is the strength of the field at the geomagnetic equator, $a_e = 6.37\times10^8$ cm is the radius of the Earth, r is the geocentric distance, and θ is the polar angle measured with respect to the north geomagnetic pole which is at 78.6° north latitude and 70.1° west longitude. The problem for determining the magnetopause and the distortion of the confined geomagnetic field is then reduced to finding a solution of equations (11) which has the dipole singularity equation (15) at the origin, and which satisfies the tangential discontinuity conditions $B_n = 0$ and $B^2/8\pi = K\rho_\infty V_\infty^2\cos^2\psi$ at the unknown location of the magnetopause. This problem is identical to the classical steady-state Chapman-Ferraro problem (ref. 10) specified over four decades ago, and based on particle rather than continuum concepts. Even at this stage, however, the problem still remains sufficiently complex that only a few numerical solutions have been carried out (refs. 11-13) at the present time. An early approximation which has proven quite successful in providing accurate magnetopause shapes (refs. 9, 12, 14) is to replace the $B_n = 0$ condition with the condition that the magnetic-field intensity at the magnetosphere boundary is equal to 2f times the tangential component of the Earth's dipole field, where f is a constant usually taken as unity (refs. 15-17).

The final mathematical statement for the free-boundary problem for the shape of the magnetosphere boundary, then, is summarized in the sketch below:



This formulation leads to the following partial differential equation for the geocentric distance r to the magnetopause for the case in which the dipole axis is perpendicular to the solar-wind flow (ref. 18)

$$\left[\frac{1 + 3\cos^2\theta}{R_m^6}\left(\frac{1}{R_m \sin\theta}\frac{\partial R_m}{\partial \phi}\right)^2 + \frac{1}{R_m^6}\left(\sin\theta + \frac{2\cos\theta}{R}\frac{\partial R_m}{\partial \phi}\right)^2\right] =$$

$$\left[\sin\phi\sin\theta - \frac{\sin\phi\cos\theta}{R_m}\frac{\partial R_m}{\partial \theta} - \frac{\cos\phi}{R_m \sin\theta}\frac{\partial R_m}{\partial \phi}\right]^2 \qquad (16)$$

where $R_m = r/D$, $D = a_e(f^2 B_{eq}^2/2\pi K\rho_\infty V_\infty^2)^{1/6}$ is the geocentric distance to the magnetosphere nose, ϕ is the geomagnetic longitude measured with respect to a line through the origin that is normal to both the dipole axis and the free-stream direction, and is equal to $\pi/2$ when directed in the upstream direction, and $3\pi/2$ when directed downstream.

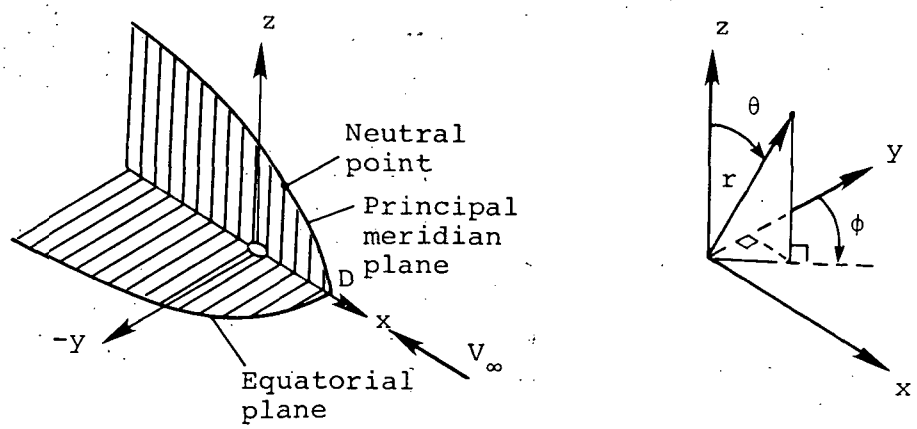
A considerable simplification occurs if attention is confined to either the geomagnetic equatorial plane or the principal meridian plane defined by the dipole axis and the direction of the incident solar wind. These equations are; in the equatorial plane ($\theta = \pi/2$, $\partial R_m/\partial\theta = 0$)

$$\frac{dR_m}{d\phi} = R_m \left(\frac{R_m^6 \sin\phi \cos\phi + \sqrt{R_m^6 - 1}}{R_m^6 \cos^2\phi - 1} \right) \qquad \pi/2 \leq \phi \leq \frac{3\pi}{2} \tag{17}$$

and in the principal meridian plane ($\phi = \pm \pi/2$, $\frac{\partial R}{\partial \phi} = 0$)

$$\frac{dR_m}{d\theta} = (R_m \tan\theta) \frac{R_m^3 \mp 1}{R_m^3 \pm 2} \quad , \qquad -\pi/2 \leq \theta \leq \frac{\pi}{2} \tag{18}$$

A sketch of the traces of the magnetosphere boundary in the equatorial and principal meridian planes is provided below together with the spherical (r, θ, ϕ) coordinate system. We note in particular that the



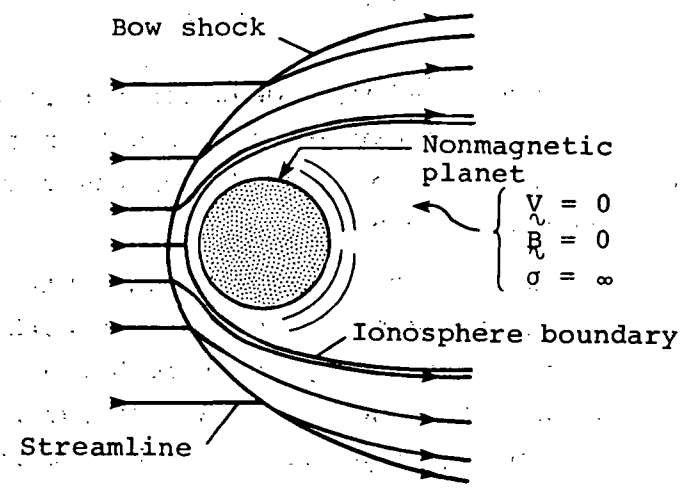
shape of the equatorial trace is a smooth curve, while that of the principal meridian contains a pronounced dent at the location of the neutral point where the magnetic field vanishes in the ideal theory. In all of the reported work to date (refs. 1-4, 6, 8) concerning flow fields past the magnetosphere, the magnetopause has been approximated by an axisymmetric shape obtained by rotating the equatorial trace of the boundary about the longitudinal axis. That this provides a very good approximation to the actual three-dimensional shape is shown in figure 1 which displays in more detail the comparison of the equatorial and principal meridian traces of the magnetopause as calculated by equations (17) and (18), respectively. Also shown on the principal meridian trace is a dashed line which forms a tangent surface across the cusped neutral point region, and which represents the free-streamline surface which will

most likely appear in nature (ref. 19). We note that all three of these shapes are quite close to one another so that the rotated equatorial trace provides a satisfactory approximation to the three-dimensional shape. Furthermore, it has been shown (ref. 3, 13) that higher-order approximations to the Chapman-Ferraro problem for the magnetosphere shape are in reasonable agreement with the approximate shape given by the solution to equation (16).

Consequently, the boundary shape of the magnetosphere for which the majority of results are presented here is the rotated equatorial trace given by equation (17). That shape has been incorporated in the associated computer codes developed herein as the default shape for magnetosphere calculations. Tabulated ordinates (Y_m/D - vs. - x/D) for the equatorial trace are provided in Table 1 where Y_m is the cylindrical radial coordinate of the profile.

Nonmagnetic Planet — Determination of the Ionosphere Boundary

The determination of the ionosphere boundary initiates from the assumptions that the ionosphere, or at least the outer part of it that participates in the interaction with the solar wind, is idealized as a spherically symmetric and hydrostatically supported plasma having infinite electrical conductivity, effectively bound to the planet and incapable of mixing with the solar wind, as indicated in the sketch below:



This interior plasma is separated from the flowing solar plasma by a tangential discontinuity across which the same relations, given previously by equations (9) for the analogous problem for a magnetic planet, must hold. The assumption of hydrostatic support implies a quiescent ionosphere in which the bulk motions of the gas with respect to the planet are sufficiently small ($\underset{\sim}{V} = 0$) that equilibrium exists between the pressure gradient and gravity, viz.

$$\frac{dp}{dr} = -\rho g \tag{19}$$

where p and ρ are the gas pressure and density, r is the radial distance measured from the center of the planet, and g is the acceleration due to gravity. The variation of g is inversely proportional to r^2, so that $g = g_s(r_s/r)^2$ where the subscript s denotes values at the surface of the planet. Since the density ρ is related to the pressure according to the perfect gas law equation (2), equation (19) can be integrated to yield

$$p = p_R \exp\left(-\int_{R_R}^{r} \frac{dr}{H}\right) \tag{20}$$

where p_R is the pressure at some reference radius R_R and H is the local scale height of the atmosphere given by $H = RT/g$. If H is regarded as constant; that is, if variations of g and T with r are neglected, equation (20) can be integrated directly to yield

$$p = p_R \exp\left(-\frac{r - R_R}{H}\right) \tag{21}$$

In view of uncertainties associated with measurements of the atmospheric properties of Venus and Mars, the variation of p with r as given by equation (21) was adopted in previous solar wind/ionosphere applications (ref. 5) and has also been used herein.

In the present application, it is also evident that the gas pressure p is much larger than the magnetic pressure $B^2/8\pi$ on both sides of the ionopause. Therefore, the discontinuity pressure balance relation $[p + B^2/8\pi] = 0$ of equation (9) reduces to a simple equality between the ionospheric pressure given by equation (21) and the static pressure of the flowing solar plasma adjacent to the ionopause. Introducing as in the magnetosphere application the Newtonian approximation equation (12) for

the static pressure of the flowing plasma on the exterior boundary of the ionosphere, we arrive at the following equation for the pressure balance at the ionopause R_i:

$$p_{st} \cos^2 \psi = K\rho_\infty V_\infty^2 \cos^2 \psi = p_R \exp\left[-\frac{R_i - R_R}{H}\right] \tag{22}$$

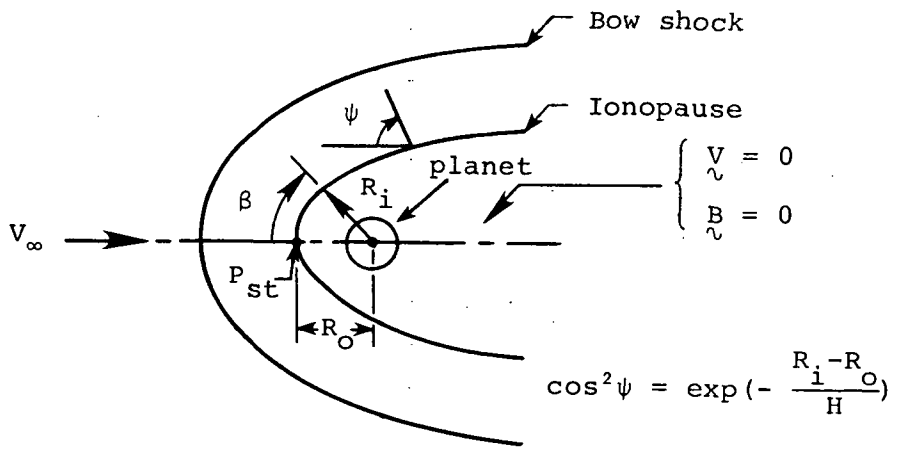
It is convenient to choose as the reference radius and location the stagnation point on the ionopause; that is, $R_R = R_o$ where R_o is the distance from the center of the planet to the nose of the ionopause. This implies that

$$p_R = p_o = K\rho_\infty V_\infty^2 \tag{23}$$

and that at all points along the ionosphere boundary

$$\cos^2 \psi = \exp\left[-\frac{R_i - R_o}{H}\right] \tag{24}$$

The final mathematical statement of the free-boundary problem for determining the shape of the ionosphere boundary then is summarized in the sketch below:



By relating the local angle ψ to the slope of the ionopause, the following ordinary differential equation results for the ordinates of the ionosphere boundary

$$\frac{dR_i}{d\beta} = R_i\left[\frac{\sin 2\beta - 2\sqrt{\Lambda - \Lambda^2}}{2(\Lambda - \sin^2\beta)}\right] \qquad 0 \leq \beta \leq \pi \tag{25}$$

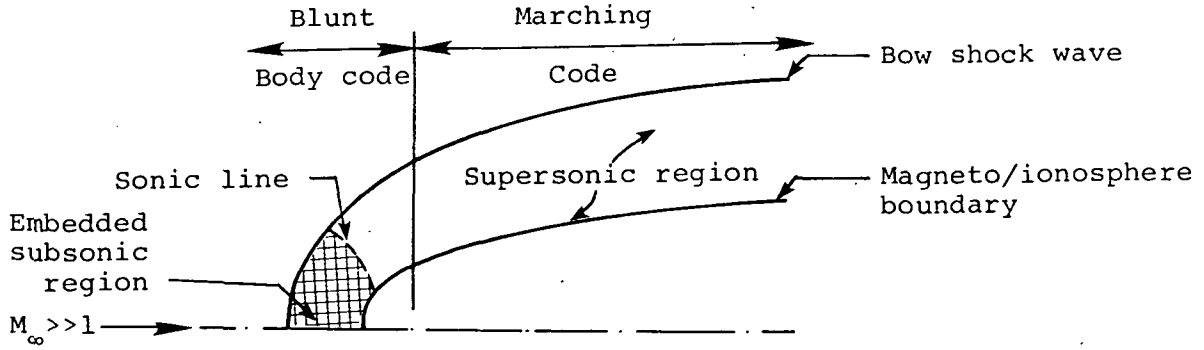
where

$$\Lambda = \exp[-(R_i - R_o)/H] \tag{26}$$

and β is the angle measured from the subsolar point as indicated above. Results for various ionopause shapes obtained by integrating equation (25) for different values H/R_o in the range $0.001 \leq H/R_o \leq 1.0$ are provided in figure 2. We note that the range of interest for planetary applications to Venus and Mars appears to be $0.01 \lesssim H/R_o \lesssim 0.30$ (ref. 5). Tabulated ordinates $(Y_i/R_o$ - vs. - X/R_o are provided in Table 1 for $H/R_o = 0.01$, 0.1, 0.2, and 0.5, where $Y_i = R_i \sin\beta$ is the cylindrical radial coordinate of the ionopause profile.

Calculation of the Gasdynamic Flow Properties

Computation of the gasdynamic flow properties consists of determining solutions to the differential equations and discontinuity conservation equations given by equations (7). Because both the downstream tail region (far field) as well as the region in the vicinity of the obstacle nose (near field) are generally of interest in solar wind-magneto/ionopause interactions, the computational methods selected must be capable of determining this entire flow field. In view of the need to carry the flow calculation to an arbitrary downstream distance, the most computationally efficient procedure to employ is to subdivide the flow field into two regions, as indicated in the sketch below:



Illustrated here are the essential features of the high supersonic Mach number flow typical of solar wind flows past terrestrial planets. Of special note is the embedded subsonic pocket, located at the nose of the magneto/ionopause. The presence of this pocket necessitates use of a computational technique capable of treating a mixed subsonic/supersonic flow. Downstream of this region, the flow becomes supersonic and remains so for the convex shapes typical of magneto/ionopause boundaries. In that ar field region, a more economical procedure than that used near the nose can be employed.

We note that this subdivision of flow field and use of different solution procedures in each region is common practice for calculating hypersonic blunt-body flows, and was employed in the former solar wind method (ref. 3) as well as in a more recent application to space shuttle re-entry flows (ref. 20). Indicated in figure 3 is a comparison of the former and present computational procedures which illustrates the breakdown of the various solution domains. The previous method treated the nose region by the inverse iteration method, and the remainder of the flow field by the method of characteristics. In light of recent advances, both of those techniques, particularly the inverse method, are now considered dated and inferior to more current methods. In the present study, those two methods have been superseded by: (1) a new axisymmetric implicit unsteady Euler equation solver (IMP) specifically developed under this contract (ref. 21) and which determines the steady state solution in the nose region by a time-marching procedure, and (2) a shock-capturing marching solution (SCT) which spatially advances the solution downstream as far as required by solving the steady Euler equations. As indicated in figure 3, the implicit unsteady Euler equation (IMP) code is used to determine the flow in the region between the stagnation streamline on the symmetry axis and a body-axis normal plane (x = constant) conveniently chosen at the $\beta = 90^\circ$ (x = 0) location. For the magneto/ionopause shapes considered here, purely supersonic flow exists on this plane and the conditions there provide the starting data for the SCT code. The SCT code then marches the solution downstream in axis normal planes to the final specified location.

Obstacle nose solution – implicit unsteady Euler equation (IMP) code.- The partial differential equations employed in the implicit (IMP) code are the unsteady axisymmetric versions of the gasdynamic Euler equations given

by equations (7). To implement the calculation, those equations are written in conservation-law form under the generalized independent variable transformation

$$\left.\begin{aligned} \tau &= T \\ \xi &= \xi(T,X,R) \\ \eta &= \eta(T,X,R) \end{aligned}\right\} \tag{27}$$

as follows:

$$(U/J)_\tau + [(\xi_T U + \xi_X E + \xi_R F)/J]_\xi + [(\eta_T U + \eta_X E + \eta_R F)/J]_\eta + G = 0 \tag{28}$$

where

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix} \qquad E = \begin{pmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ (e + p)u \end{pmatrix} \tag{29}$$

$$F = \begin{pmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ (e + p)v \end{pmatrix} \qquad G = \frac{1}{RJ} \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 \\ (e + p)v \end{pmatrix}$$

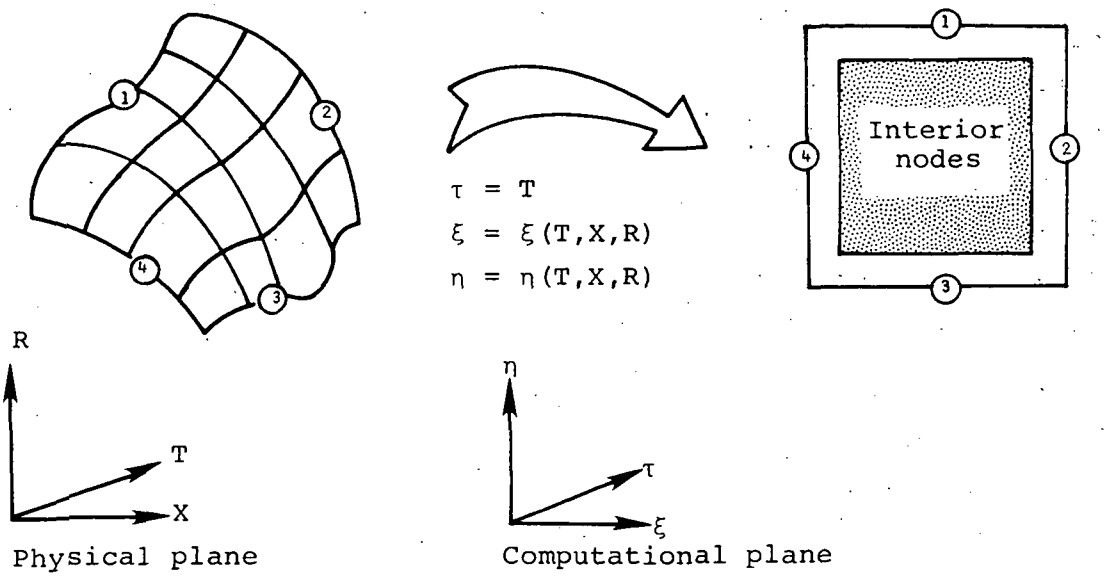
and the Jacobian

$$J = \xi_X \eta_R - \xi_R \eta_X \tag{30}$$

In equations (27) to (30), T denotes time, X is the axial downstream direction, and R is the cylindrical radial distance; p represents the pressure; ρ the density; u and v the velocity components in the X and R directions, respectively; and e the total energy per unit volume. The following equation relates the pressure, density, and velocity components to the energy for an ideal gas

$$e = p/(\gamma - 1) + \rho(u^2 + v^2)/2 \tag{31}$$

Using the independent variable transformation equation (27), the physical plane with boundaries 1-4 in X,Y,T space is mapped into the rectangular computational plane ξ, η, and τ space as shown in the sketch below. Generally, at each time step the transformation metrics are not known beforehand and must be determined numerically. Integration step size is established by using the eigenvalues of the Jacobian matrices $\bar{A}$ and $\bar{B}$, where $\bar{A} = \partial\hat{E}/\partial\hat{U}$, $\bar{B} = \partial\hat{F}/\partial\hat{U}$ and $\hat{U} = U/J$, $\hat{E} = (\xi_T U + \xi_X E + \xi_R F)/J$, and $\hat{F} = (\eta_T U + \eta_X E + \eta_R F)/J$.



The boundaries, as indicated in the sketch, at which appropriate conditions are imposed on the present problem consist of: (1) the bow shock surface at which the Rankine-Hugoniot relations are satisfied, (2) the downstream outflow boundary where the flow is assumed to be entirely supersonic, (3) the obstacle surface at which a no flow condition in the normal direction is imposed, and (4) the stagnation streamline where symmetry conditions are implemented. Initial flow field conditions are determined by guessing a bow shock shape for the particular blunt obstacle at hand and by prescribing a Newtonian pressure distribution on the body. Since the maximum entropy streamline wets the body, that fact plus the known surface flow direction serve to determine the remainder of the flow properties on the obstacle surface. A linear variation for the flow properties between the bow shock and the obstacle is then prescribed.

This provides the initial flow field which is then integrated in a time-asymptotic fashion until the steady-state solution is obtained.

The basic numerical algorithm used in the IMP code was developed by Beam and Warming (ref. 22) and is second-order accurate, noniterative, and spatially factored. In particular, the "delta form" with Euler time differencing is employed. When applied to equation (28) the algorithm assumes the form

$$(I + \Delta\tau\delta_\xi \bar{A}^n)(I + \Delta\tau\delta_\eta \bar{B}^n)(\hat{U}^{n+1} - \hat{U}^n) = -\Delta\tau(\delta_\xi \hat{E}^n + \delta_\eta \hat{F}^n + G) \qquad (32)$$

where $\bar{A}$ and $\bar{B}$ are the Jacobian matrices, I is the identity matrix, δ_ξ and δ_η are second-order, central difference operators, $U^{n+1} = U(n\Delta\tau)$ and $\Delta\tau$ is the integration step size.

Equation (32) is solved at the interior points only. It requires two 4×4 block tridiagonal inversions at each time step of the integration. The solution procedure is as follows:

1. Define $\Delta\hat{U} = U^{n+1} - \hat{U}^n$.

2. Form the right-hand side of equation (32) and store results in the $\hat{U}^{n+1}$ array.

3. Define $\bar{U} = (I + \Delta\tau\delta_\eta \bar{B}^n)\Delta\hat{U}$ and solve the matrix equation $(I + \Delta\tau\delta_\xi \bar{A}^n)\bar{U} = \hat{U}^{n+1}$ for $\bar{U}$ storing the result in the $\hat{U}^{n+1}$ array.

4. Solve the matrix equation $(I + \Delta\tau\delta_\eta \bar{B}^n)\Delta\hat{U} = \hat{U}^{n+1}$ for $\Delta\hat{U}$.

5. Obtain the values of $\hat{U}^{n+1}$ from the relation $\hat{U}^{n+1} = \Delta\hat{U} + \hat{U}^n$.

6. Obtain the final solution after applying the smoothing; $\hat{U}^{n+1} = \hat{U}^{n+1} - (\varepsilon/8)S/J$.

A fourth-order smoothing term S is usually required to eliminate nonlinear instabilities which may arise since the use of central differences in the spatial directions results in a neutrally stable algorithm.

Downstream solution – shock capturing steady Euler equations (SCT) code.- Since the shock-capturing technique employed has been established previously (refs. 23, 24, 25) only a brief description will be provided here. The steady, full three-dimensional Euler equations are written in cylindrical coordinates in the form:

$$\frac{\partial \tilde{U}}{\partial X} + \frac{\partial \tilde{F}}{\partial R} + \frac{\partial \tilde{G}}{\partial \omega} + \tilde{H} = 0 \tag{33}$$

where

$$\left.\begin{aligned} \tilde{U} &= \begin{pmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ \rho uw \end{pmatrix} \qquad \tilde{G} = \frac{1}{R}\begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ p + \rho w^2 \end{pmatrix} \\ \tilde{F} &= \begin{pmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ \rho vw \end{pmatrix} \qquad \tilde{H} = \frac{1}{R}\begin{pmatrix} \rho v \\ \rho uv \\ \rho (v^2 - w^2) \\ 2\rho vw \end{pmatrix} \end{aligned}\right\} \tag{34}$$

Here p and ρ represent dimensional pressure and density and (u,v,w) denote velocity components in the cylindrical coordinate directions (X,R,ω), respectively, where X is the axial downstream direction, R is the cylindrical radius in an axis-normal plane, and ω is the azimuthal angle measured in the axis-normal plane. This set of equations is closed by the use of energy conservation as provided by the equation for total enthalpy

$$H_t = h(p,\rho) + \frac{q^2}{2} = \text{constant} \tag{35}$$

where

$$q = \sqrt{u^2 + v^2 + w^2} \tag{36}$$

is the magnitude of the velocity vector and $h(p,\rho)$ is the caloric equation of state.

The equations are then transformed to a computational plane by using the independent variable transformation

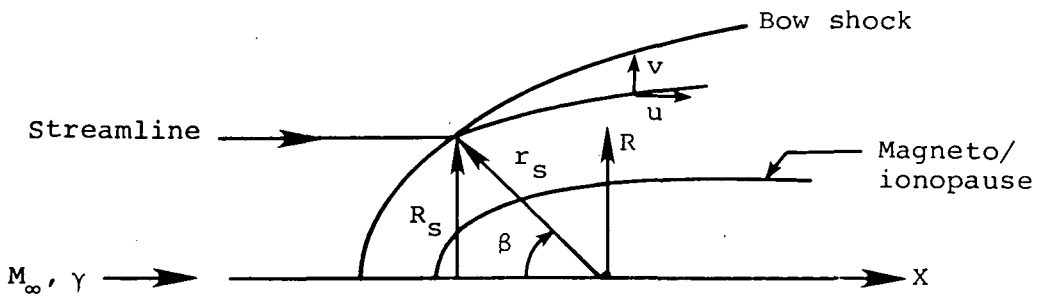
$$\left.\begin{aligned} X &= X \\ \xi(X,R,\omega) &= \frac{(R - R_b)}{(R_s - R_b)} \\ \xi(\omega) &= f(\omega) \end{aligned}\right\} \tag{37}$$

where the subscripts (b,s) denote body and shock surfaces, respectively. This change of variable is employed in equation (33) and the result then rearranged into the following conservation law form (refs. 23,24,25).

$$\frac{\partial U}{\partial X} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + H = 0 \tag{38}$$

The finite-difference formulation of MacCormack (ref. 26) is used to integrate equation (38) with respect to the hyperbolic coordinate X to yield values of the conservative variable U. This procedure is calculable as long as the flow velocity in the axial (X) direction is greater than the local speed of sound. We note that for the axisymmetric obstacle shapes considered here, $\partial/\partial\omega = 0$; however, the version of the SCT code employed has the capability for treating full three-dimensional geometries.

Streamline and contour calculations.- A number of special purpose subroutines were written to calculate streamlines and contour maps of various flow field properties, and also to provide automated plots of these quantities. The streamlines are determined by integrating trajectories through the known velocity field as this procedure was found to be more accurate than a mass flow calculation. The calculation of a particular streamline is initiated at the point where the streamline crosses the bow shock, as shown below:



At that point, exact values of the streamline slope dR_s/dX are known in terms of the local shock angle δ_s and free-stream quantities according to the relation

$$\frac{dR_s}{dX} = \frac{2 \cdot \cot \delta_s \cdot (M_\infty^2 \sin^2 \delta_s - 1)}{2 + M_\infty^2 (\gamma + 1 - 2 \sin^2 \delta_s)} \tag{39}$$

which is implicit in the blunt-body (IMP) and marching (SCT) code solutions. At other points in the flow field, the local streamline slope is given by the ratio of radial to downstream velocity, i.e.

$$\frac{dR_s}{dX} = \frac{v}{u} \tag{40}$$

and the streamline determination is made by stepwise integration in X of equation (40) according to a modified third-order Euler predictor/corrector method. Two-dimensional linear interpolation from the flow field grid points is employed to obtain the velocity components (u,v) required at the stepwise points along the streamline trajectory. Separate streamline calculations are made for the near field (IMP results) and far field (SCT results) because of the different coordinate systems employed in those two regions.

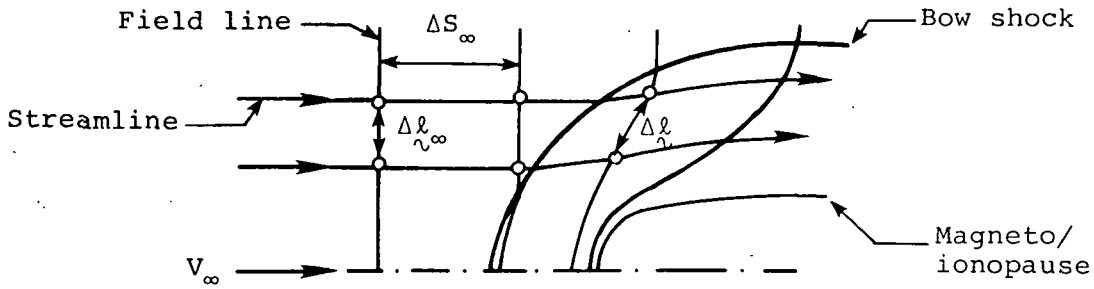
Contours of various flow field quantities are determined for the entire flow field (near field plus far field) by using a modified version of the contour package developed by R. Sorensen at NASA/Ames Research Center. Pertinent details of that package are provided in the user's manual section of this report.

Calculation of the Magnetic Field

With the flow properties known from the gasdynamic calculations, determination of the magnetic field $\underset{\sim}{B}$ proceeds by integrating equations (8). Those equations are commonly interpreted as indicating the field lines move with the fluid. This leads to a straightforward, although tedious, calculation in which the vector distance from each point on an arbitrarily selected fluid line to its corresponding point on an adjacent field line in the downstream direction is determined by numerically integrating $\int \underset{\sim}{v}\, dt$ over a fixed time interval δt. While that calculation determines the location of the field lines, the intensity of the magnetic field at any point may then be calculated from the relation

$$\frac{|\underset{\sim}{B}|}{|\underset{\sim}{B}_\infty|} = \frac{\rho\,|\Delta\underset{\sim}{\ell}|}{\rho_\infty\,|\Delta\underset{\sim}{\ell}_\infty|} \tag{41}$$

where $\Delta\underset{\sim}{\ell}$ is the vector length of a small element of a flux tube. The sketch below clarifies these quantities for the case when the oncoming $\underset{\sim}{B}$ field is perpendicular to the flow:



where the open symbol $\odot$ denotes locations of points on the streamlines corresponding to a fixed time interval $\Delta t = \Delta S_\infty / V_\infty$.

Because of the axisymmetry of the gasdynamic quantities and the linearity of the equations for $\underset{\sim}{B}$, Alksne and Webster (ref. 27) have shown that the magnetic field at any point in the flow field can be calculated by vectorially summing the contributions of the three component fields indicated in figure 4. At any point P, the magnetic field $\underset{\sim}{B}_P$ is given by

$$\underset{\sim}{B}_P = \left(\frac{\underset{\sim}{B}_P}{B_\infty}\right)_{\parallel} B_{\infty_{\parallel}} + \left(\frac{\underset{\sim}{B}_P}{B_\infty}\right)_{\perp} B_{\infty_{\perp}} + \hat{e}_n \left(\frac{B_P}{B_\infty}\right)_n B_{\infty n} \tag{42}$$

where the subscripts ($\parallel$, $\perp$, n) refer respectively to the contributions associated with the components of $\underset{\sim}{B}_\infty$ parallel to $\underset{\sim}{V}_\infty$; perpendicular to $\underset{\sim}{V}_\infty$ in the plane that contains the point P, the center of the planet, and the vector $\underset{\sim}{V}_\infty$; and normal to the latter plane.

For the contribution associated with the parallel component $B_{\infty_{\parallel}}$, it has been shown (refs. 28,29) that $(\underset{\sim}{B}_P)_{\parallel}$ is proportional to $\rho\underset{\sim}{V}$ throughout the entire flow field. Consequently, this component can be calculated directly from the gasdynamic solution by the expression

$$\left(\frac{\underset{\sim}{B}_P}{B_\infty}\right)_{\parallel} = \frac{\rho_P \underset{\sim}{V}_P}{\rho_\infty |\underset{\sim}{V}_\infty|} \tag{43}$$

Similarly, the normal component $(B_p/B_\infty)_n$ can be shown to be equal to

$$\left(\frac{B_p}{B_\infty} \right)_n = \frac{R_p \rho_p}{R_{p_\infty} \rho_\infty} \tag{44}$$

where R_p is the radial cylindrical coordinate of the streamline, as indicated in figure 4.

The determination of the contribution $(\underset{\sim}{B}_p/B_\infty)_\perp$ cannot be made directly from the gasdynamic properties as can the other two components. For this component, recourse must be made to the general computation associated with equation (41). Details of the calculation procedure for $(\underset{\sim}{B}_p/B_\infty)_\perp$ are provided in the user's manual section of this report.

RESULTS AND DISCUSSION

In order both to verify the correctness of the procedures developed under this study as well as to demonstrate their flexibility and power for calculating solar wind flows for a variety of conditions, a large number of test cases have been run. A representative sample of the results obtained from those calculations is reported here. Because the implicit blunt-body procedure (IMP code) is new, that code was tested and compared to previously established theoretical methods to substantiate its validity prior to solar wind applications. We have prepared figure 5 to exhibit such a comparison. Here, flow field results predicted by the present implicit method are compared with those of other theoretical techniques and also experimental data for supersonic flow past a sphere with M_∞ = 4.926 and γ = 1.4. The variation of density and Mach number along the stagnation streamline are provided in the two plots on the left, while the variation of surface pressure and shock standoff distance with spherical angle are given in the two plots on the right. In all of these comparisons, the present theoretical results are in excellent agreement with both established theoretical methods and experiment.

Figure 6 exhibits a comparison of results predicted by the implicit method for a solar wind calculation. The upper plot of the figure displays the locations of the bow wave and sonic line for flow past the equatorial trace of the magnetopause for M_∞ = 8 and γ = 5/3. The density distribution along the magnetosphere boundary and along the shock wave for this flow are given in the lower plot. Included in both of these figures is the solution calculated by Spreiter et al. (ref. 3) using the inverse method of Inouye and Lomax (ref. 30). Excellent agreement is obtained between the present implicit technique and the inverse method. Figure 7 provides analogous results for the equatorial trace of the magnetopause for a variety of Mach numbers typical of solar wind flows, and illustrates the Mach number capability of the present IMP code.

With regard to solar wind flows past nonmagnetic planets, figure 8 displays results for the bow shock location for M_∞ = 8 and γ = 5/3 flow past various ionopause shapes with H/R_o = 0.01, 0.1, 0.2, 0.25, 0.50, 0.75 and 1.0. Not included in that figure for clarity of presentation are the results of Spreiter et al. (ref. 5) obtained by using the inverse method. Those results and the predictions of the present implicit method given in figure 8 are essentially indistinguishable.

As an illustration of the geometric flexibility of the present implicit method and also as a critical test of its ability to capture embedded shock waves besides the bow shock, a feature which the inverse method cannot duplicate, figures 9 and 10 have been prepared. This calculation corresponds to the $M_\infty = 5$ and $\gamma = 5/3$ flow around the axisymmetric shape generated by rotating the principal meridian of the magnetopause about its axis and compares with an experimental test by Spreiter et al. (ref. 3). This particular profile shape contains a pronounced dent with a concave corner in the vicinity of the neutral point. Spreiter et al. (ref. 3) have argued that the presence of a concave corner in the magnetopause is unrealistic and would not occur in nature, but should be replaced by a free surface created by drawing a tangent line across the corner. In figure 9, we present results for both the concave and the free surfaces, denoted by solid and dashed lines, respectively. For the concave surface, a pocket of embedded subsonic flow is seen to form behind the embedded shock wave which was caused by the indented profile. If the surface is modified to the free surface shape, the embedded wave disappears completely as expected. The corresponding pressure coefficients on the magnetosphere boundary are presented in figure 10. For the concave surface, the shock wave is located on the body at approximately $\beta = 80°$, while for the free surface, the pressure coefficient displays a constant value as anticipated. Finally, we note that the calculation of a supersonic flow with an embedded shock and subsequent subsonic pocket provides a severe test for any blunt body procedure. The ability of the present code to provide convergent results for such a flow demonstrates the capability for further extension and application to more generalized profiles than was heretofore possible.

As a final indication of the range of solar wind flows over which the implicit code has been tested, we have prepared figure 11 which displays the variation of shock stand-off distance with oncoming Mach number and ratio of specific heats for three different magneto/ionopause shapes. These include the rotated equatorial trace of the magnetopause, and the ionopause shapes for $H/R_o = 0.01$ and 0.5. Those shapes and the corresponding ranges of M_∞ and γ essentially span the entire range of interest of geometry and solar wind conditions for which the computer programs developed herein would be normally applied.

In order to demonstrate the ability of the marching code to continue the blunt-body IMP solution downstream to some arbitrarily-specified

downstream location, and also to verify its stability and convergence for application to these classes of flows, we have calculated marching solutions for a variety of cases typical of solar wind interactions. Figure 12 exhibits the location of the bow wave for flows with $\gamma = 5/3$ and $M_\infty = 3$, 8, and 25 past the rotated equatorial trace of the magnetopause as the flow proceeds downstream from the nose region to $x/D = -11$. For this calculation, starting conditions for the marching code (SCT) have been provided by the blunt-body (IMP) code on the line $x/D = 0.0$, which is the usual location at which the two solutions are joined by the computer program. The marching code then determines the remainder of the flow field back to the specified downstream location. Since for the particular shapes considered herein the downstream flows are quite smooth, the marching calculation is very efficient (less than 30 sec., CDC 7600, OPT=2 compiler). Similar results are presented in figure 13 for $M_\infty = 8$ and $\gamma = 5/3$ flow past an ionopause shape with $H/R_o = 0.1$. Those results have been carried downstream to $x/R_o = -20$ since this distance is more typical of the region of interest for nonmagnetic planets. This is so because of the different scalings (D, R_o) used for the magnetic and nonmagnetic applications. For Venus and Mars, the location of the ionopause nose is at a distance only slightly greater than the planetary radius, as compared with the location of the magnetopause nose for the Earth which is of the order of 10 planetary radii.

In order to illustrate the capability of the present procedures to determine streamlines, contour maps of various flow properties, and magnetic field lines and contours, as well as to demonstrate the automated plotting capability for displaying these results, figures 14 and 15 have been prepared. Figure 14 illustrates the computer generated streamline locations and contour maps of density ρ/ρ_∞ and velocity ratio V/V_∞ for the complete near and far field flow about the equatorial trace of the magnetosphere for $M = 8$ and $\gamma = 5/3$. Based on this gasdynamic solution, figure 15 exhibits the corresponding results for the magnetic field components $(\underset{\sim}{B}/B_\infty)_\perp$ and $(\underset{\sim}{B}/B_\infty)_{\shortparallel}$ in the plane of symmetry. Both magnetic field line locations and contours are provided. In addition to demonstrating the overall smoothness of the computed results, these two figures illustrate the ability of the present techniques to provide the completely automated production of report quality plots of both gasdynamic and magnetic field properties for solar wind flows past axisymmetric magneto/ionopause shapes

CONCLUDING REMARKS

Theoretical analysis and associated development of computer programs were carried out for the purpose of developing computational techniques for predicting the interaction of solar wind flows with both magnetic and nonmagnetic terrestrial planets. Based on the identical theoretical model employed in previous work, i.e. the steady, dissipationless, magnetohydrodynamic model for axisymmetric, supersonic, super-Alfvénic solar wind flow a new and more powerful solution procedure has been developed. The procedure is built upon an assemblage of computer codes, including:

- Blunt-body code – to determine gasdynamic solution near obstacle nose
- Marching code – to determine gasdynamic solution downstream of obstacle nose
- Streamline code – to determine flow field streamlines
- Contour code – to determine contour maps of flow and magnetic field properties
- Magnetic field code – to determine frozen-in magnetic field
- Plotting code – to plot selected flow and magnetic field results

Comparisons are reported which demonstrate the accuracy of the present techniques by comparison with previously-established theoretical methods as well as limited experimental data. Furthermore, new results are presented for a variety of solar wind flows which illustrate the flexibility and generality of the component methods. The computational procedures are fully automated and provide detailed flow field and magnetic field properties in convenient output format, including an automatic plot capability. The programs are documented and are presented in a general user's manual included as part of this report.

With regard to improvements of the present techniques to provide a more accurate mathematical representation of solar wind flow, a number of items for further study are recommended. The first of these involves extension of the present capability from axisymmetric to three-dimensional nonaxisymmetric obstacle shapes representative of planetary magnetopauses. The second involves the merging of the present procedures for determining the magnetic field exterior to the magnetosphere boundary with a method, such as the Olsen-Pfitzer model, for calculating the confined magnetic field so as to provide a complete magnetic field representation. Lastly, the capability for determining numerical solutions of the complete magnetohydrodynamic equations, rather than the frozen magnetic field approximation,

should be actively pursued. Although this extension represents a major advance in magnetospheric physics and computational fluid dynamics, in view of the success of the currently employed methods, its successful achievement appears highly promising. The component methods on which the present procedure is based are directly capable of extension in these as well as other directions. Moreover, the efficiency of the present calculations indicates that the improved representation achieved by these improvements will not result in prohibitive computational costs.

APPENDIX A

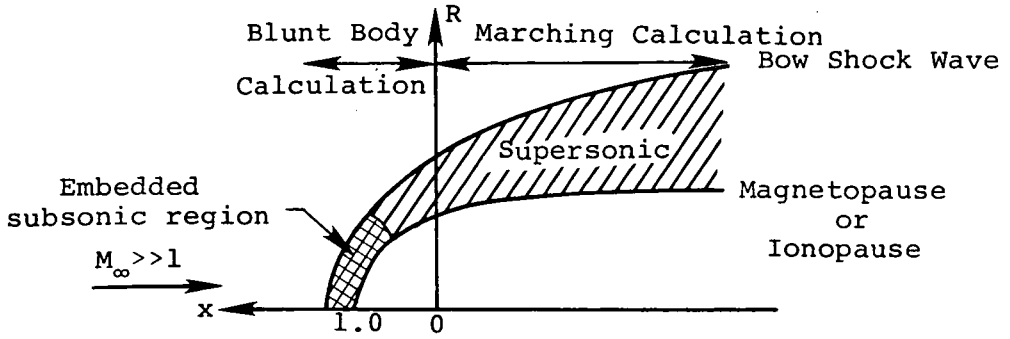
A.1 Introduction

The purpose of this appendix is to describe the operation of the assemblage of computer codes which were developed in conjunction with the theoretical work presented in this report and organized into one program, and to provide sufficient detail to permit understanding and use of the program. The program computes the flow field of the solar wind about a terrestrial planet, using a procedure for the calculation of supersonic/hypersonic flow about an axisymmetric blunt body. The corresponding frozen-in magnetic field is then calculated from the previously determined velocity and density fields. Streamlines and contour lines of various flow field properties and magnetic field components are also determined.

A description of the general operating procedure of the program is given, with descriptions of input and output. The program is written in FØRTRAN IV and has been developed on a CDC 7600 computer. University Computing Company (UCC) Standard Plotting Software and Functional Software packages are used to produce automated plots. Files used, besides TAPE5 for INPUT and TAPE6 for ØUTPUT, are TAPE1 for the plot file (system default), TAPE4 for input file for rerun option, and TAPE9 for storing data for rerun. Typical run times for cases using the default parameters are 40 to 50 seconds, using the ØPT=2 compiler. For a case using the rerun option, which employs a previously calculated flow field, the run time is approximately 5 seconds.

A.2 Program Description

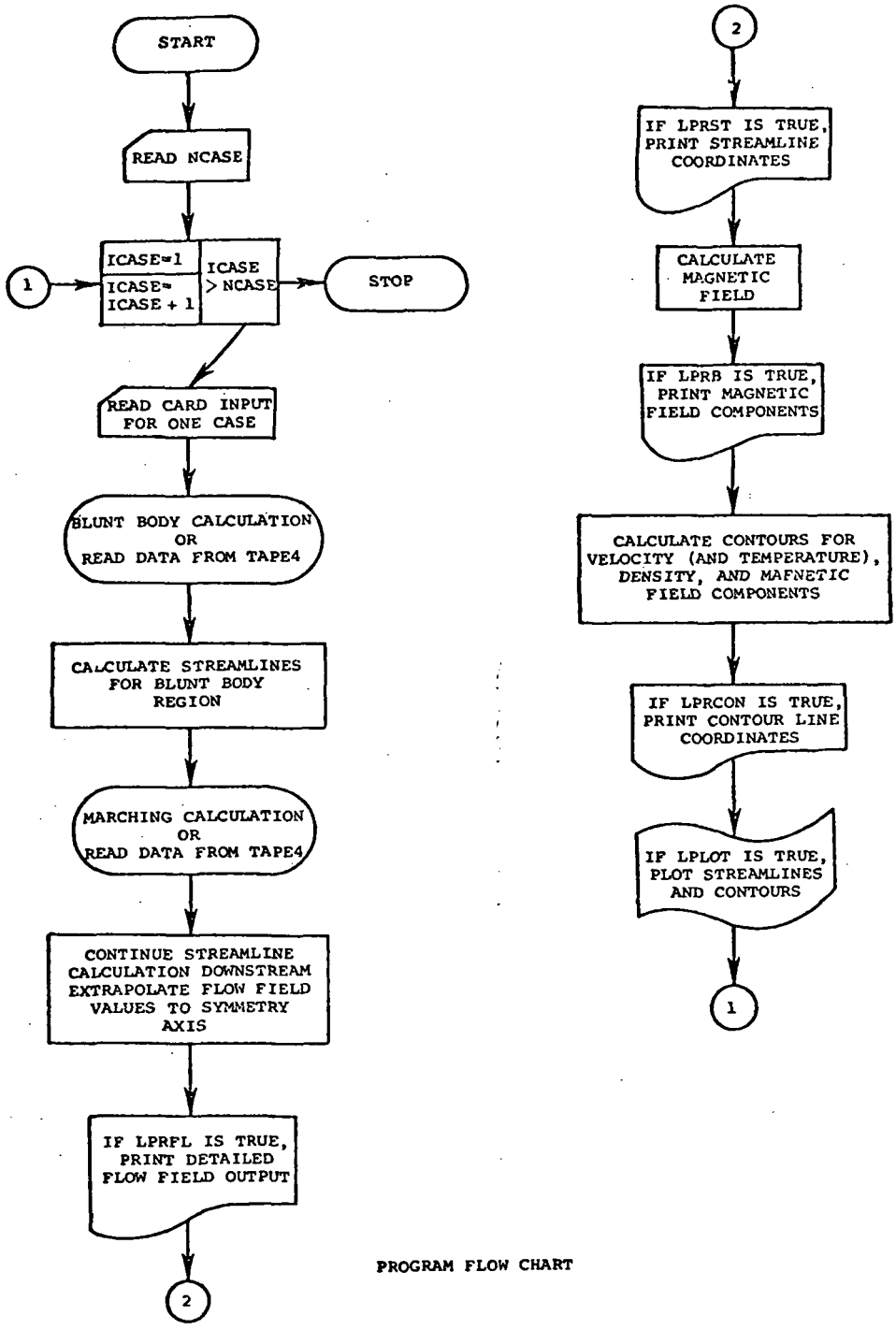
For computational purposes, the flow is subdivided into two regions, as indicated in the sketch below, with the center of the planet as origin.



The region near the nose of the magnetopause/ionopause includes all of the imbedded subsonic flow and part of the supersonic flow. An axisymmetric

implicit unsteady Euler equation solver is used to calculate this part of the flow field. Using the solution plane at x = 0.0 to provide starting conditions, the flow field in the purely supersonic downstream region is determined by integrating the steady Euler equations using a spatial marching procedure. Streamlines, the magnetic field, and contours are calculated using the entire flow field, distinguishing between the two regions as required by the different forms of the computational grids. A rerun capability is provided, where flow field data is read from a file written on a previous run, rather than repeating the blunt body and marching calculations. The computations proceed as shown in the sketch below, which provides an overall flow chart of the complete program. The program provides for several cases to be run consecutively.

APPENDIX A

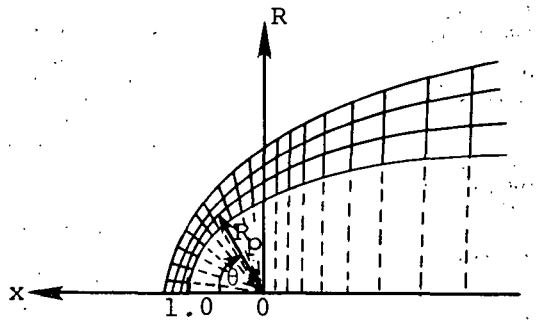


PROGRAM FLOW CHART

A.2.1 Calculation Procedure

After reading in the number of cases in the run, each case is calculated independently. Subroutine INPUT reads in all card input required for one case, viz. a title, flow conditions, obstacle geometry, calculation and print control parameters, and desired contour values. The user may supply the obstacle geometry in the form of a shape table for an axisymmetric body, or use one of the default shapes which are calculated internally by the program. These default shapes are the magnetopause equatorial trace and a constant scale-height ionopause for which H/R_o is specified by the user. The input is printed as the first item of output.

A computational mesh in polar (R_p,θ) coordinates is established for the blunt body calculation; then, for the marching calculation, this is extended into a cylindrical (x,R) system, as indicated below:



All lengths, x, R, R_p, are scaled so that the nose of the obstacle is at x = 1.0. For the default shapes, rays at equal angular increments of $\Delta\theta$ are used, starting at $-\Delta\theta/2$, up to $90° + \Delta\theta$, where $\Delta\theta = 90°/(\text{NBLUNT}-1.5)$ and NBLUNT is an input parameter describing the number of angular mesh points to be used in the blunt-body calculation. Program default value is NBLUNT = 24, so that for the default mesh, $\Delta\theta = 4°$. The obstacle shape is determined by integrating the appropriate differential equation by a trapezoidal predictor-corrector method. For a user-supplied shape, the θ grid is determined by rays from the origin through the first NBLUNT points, and the reflection of the first ray in the x-axis. Values for R_p are determined by dividing the line segments between the body and bow shock wave into NR-1 equal intervals. Thus, including the obstacle and bow shock wave, the grid forms NR arcs around the obstacle. A starting solution for the blunt-body calculation is obtained by guessing a bow shock shape and by prescribing a Newtonian pressure distribution on the body.

APPENDIX A

Noting that the maximum entropy streamline wets the body, other flow properties on the body surface can then be calculated. An initial flow field is then established by linear interpolation between the obstacle and the guessed bow shock, where the Rankine-Hugoniot relations hold. The integration proceeds in time for ITER steps. The initial bow shock shape used for the magnetopause equatorial trace and for an ionopause with $H/R_o \geq 0.1$ is a correlation shape depending on $(M_\infty, \gamma, H/R_o)$ and given by the parabola $R_p = \delta_1\sqrt{\delta_o - x}/\sqrt{\delta_o}$ where

$$\delta_o = 1.0 + 1.1\left((\gamma-1)M_\infty^2 + 2\right)/\left((\gamma+1)M_\infty^2\right) \times (0.9 + 0.5\ H/R_o)$$

$$\delta_1 = \Delta_o\{(1.273 + 0.009\ M_\infty^2)(0.904 + 0.655\ H/R_o)$$

$$\times (3.95 - 5.3\ H/R_o + 3.85\ (H/R_o)^2)\} + (R_{body})_x = 0.0$$

$$\Delta_o = \left((\gamma-1)M_\infty^2 + 2\right)/\left((\gamma+1)M_\infty^2\right) \times 0.78$$

For a user-supplied obstacle shape and for an ionopause with $H/R_o < 0.1$, the initial shock shape used is the curve $R_p = \sqrt{(1 + \Delta_o(1 + 0.68\theta^2 + 0.16\theta^4))}$. Information on convergence, the final sonic line locations, and the body and final bow shock shape are printed from this calculation.

The results at $\theta = 90°$, i.e. at $x = 0.0$, are used as starting conditions for the marching calculation, after proper normalization. For default geometries, the integration of the appropriate differential equation is continued downstream at equal θ increments to form a body shape table. The stepsize along the x-axis is recalculated at every ICØNST(49) with ICØNST(49) being set to 10. At each x-location, R_{body} is determined by linear interpolation. The computational mesh is extended by adding the line perpendicular to the x-axis at each step, divided in the same manner as for the blunt nose. The calculation marches downstream with a maximum stepsize of 1.0 until the terminal location specified by the user has been passed. However, the number of steps is limited to 75, after which the calculation will end regardless of the x-location. The coordinates of the obstacle and bow shock are printed at each step.

The grid coordinates and flow field values are written to a file, TAPE9, which may be saved to use as input for a later run. This rerun option, which replaces constructing the computational mesh and performing

the blunt body and marching calculations with reading the rerun input file, TAPE4, is described in section A.2.2.

The streamlines are calculated in two sections, following each of the flow field calculations. Using the results of the blunt body calculation, i.e. the (x, R) grid coordinates, (R_p, θ) grid coordinates, density ρ/ρ_∞, and velocity components v_x/v_∞ and v_R/v_∞, the velocity magnitude $|\underset{\sim}{v}|/v_\infty$ and flow angle are calculated. Density ρ/ρ_∞ and velocity $|\underset{\sim}{v}|/v_\infty$ are smoothed along the rays of constant-θ, using a third degree least squares fit with respect to R_p. Streamlines are then calculated downstream to $x = 0.0$, using the trajectory method, integrating through the velocity field by means of a third-order modified Euler integration procedure, using the grid locations on the bow shock as starting positions. The flow angle $\phi = \tan^{-1}(v_R/v_x)$ at each point is determined using bivariate linear interpolation in θ, then R_p. Points for which $\theta < 0°$ or $\theta > 90°$ are discarded in the interpolation.

The marching calculation provides (x, R) grid coordinates, and values of density ρ/ρ_t, and velocity components v_x/v_t and v_R/v_t, where t denotes free-stream stagnation conditions. For compatibility with the blunt body solution, the flow field values are converted to ρ/ρ_∞, v_x/v_∞, v_R/v_∞ before calculating the resultant velocity magnitude $|\underset{\sim}{v}|/v_\infty$ and flow angle ϕ. The streamline calculation is continued downstream, employing the same method as in the nose region. Starting positions on the shock wave for the streamline calculation in the marching zone are set at equal R-increments, with a maximum of 50 streamlines calculated. The flow angle is determined using bivariate linear interpolation first in x, then in R.

Along the symmetry axis, values of x, ρ/ρ_∞, and $|\underset{\sim}{v}|/v_\infty$ are determined by extrapolation, using a third-order Lagrangian polynomial in θ on each arc of the computational grid. Exact values for the stagnation streamline are used where possible, viz.

at the bow shock

$$\rho/\rho_\infty = (\gamma+1)M_\infty^2/\left[(\gamma-1)M_\infty^2 + 2\right]$$

$$|\underset{\sim}{v}|/v_\infty = 1/(\rho/\rho_\infty)$$

at the body surface

$$\rho/\rho_\infty = (\rho/\rho_\infty)_{shock}\left\{\left[\left[(\gamma+1)M_\infty\right]^2/\left[4\gamma M_\infty^2 - 2(\gamma-1)\right]\right]^{1/(\gamma-1)}\right\}$$

$$|\underset{\sim}{v}|/v_\infty = 0.0$$

$$x = 1.0$$

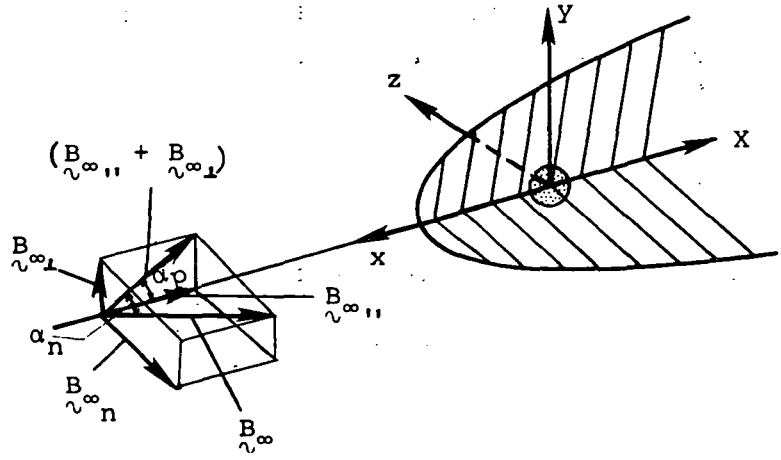
Detailed flow field output may now be printed by subroutine FLØUT, with LPRFL as print control variable. In addition to grid coordinates, density, velocities and flow angle, values of temperature T/T_∞ and pressure P/P_∞ are output, where

$$T/T_\infty = 1 + \left[(\gamma-1)/2\right]M_\infty^2\left[1 - (|\underset{\sim}{v}|/v_\infty)^2\right]$$

$$P/P_\infty = (\rho/\rho_\infty)(T/T_\infty)$$

Streamline coordinates may also be printed by subroutine STØUT, with LPRST as print control variable.

The magnetic field is determined by separately calculating the components whose field lines are parallel, perpendicular, and normal to the flow, in the undisturbed solar wind. These components are then added vectorially, the resultant being expressed in orthogonal (x,y,z) components. The angles in the free stream α_p and α_n between the magnetic field and the flow, as shown in the sketch below, are input variables. The components are calculated using the following formulae, in which $\underset{\sim}{e}$ signifies a vector of magnitude e, in the direction of the component field line, and $\hat{n}$ the unit normal vector.



$$\left(\frac{\underset{\sim}{B}}{B_\infty}\right)_{\shortparallel} = \left(\frac{\underset{\sim}{V}}{V_\infty}\right)\left(\frac{\rho}{\rho_\infty}\right) \; ; \; \left(\frac{\underset{\sim}{B}}{B_\infty}\right)_{\perp} = \left(\frac{\Delta\underset{\sim}{\ell}}{\Delta\ell_\infty}\right)\left(\frac{\rho}{\rho_\infty}\right) \; ; \; \left(\frac{B}{B_\infty}\right)_n = \left(\frac{R}{R_\infty}\right)\left(\frac{\rho}{\rho_\infty}\right)$$

$$\left(\frac{\underset{\sim}{B}}{B_\infty}\right) = \left(\frac{\underset{\sim}{B}}{B_\infty}\right)_{\shortparallel}\left(\frac{B_{\infty\shortparallel}}{B_\infty}\right) + \left(\frac{\underset{\sim}{B}}{B_\infty}\right)_{\perp}\left(\frac{B_{\infty\perp}}{B_\infty}\right) + \hat{n}\left(\frac{B}{B_\infty}\right)_n\left(\frac{B_{\infty n}}{B_\infty}\right)$$

The magnetic field line vector component $\underset{\sim}{B}_{\shortparallel}$ which results from the interplanetary component $B_{\infty\shortparallel}$ that is parallel to the undisturbed solar flow has local magnitude given by $(|\underset{\sim}{v}|/v_\infty)(\rho/\rho_\infty)$, and the same local direction ϕ as the fluid flow. Determination of the normal magnetic field component B_n requires calculation of R/R_∞, where R_∞ is the free stream cylindrical R-ordinate of the streamline through the point under consideration. This is calculated by linearly interpolating in the local radial cylindrical coordinate R between the streamlines, with $R/R_\infty = 1.0$ along the x-axis. The magnetic field vector component $\underset{\sim}{B}_{\perp}$ resulting from the interplanetary component $B_{\infty\perp}$ which is perpendicular to the undistrubed solar wind flow requires the distance vector $\Delta\underset{\sim}{\ell}/\Delta\ell_\infty$, whose magnitude is $\Delta\ell/\Delta\ell_\infty$ and direction is ψ, where $\Delta\ell/\Delta\ell_\infty$ is the stretching factor of the perpendicular field at the point, and ψ is the direction of the field line through the point. The perpendicular field lines are determined by integrating $\int\underset{\sim}{v}dt$ along each streamline, using trapezoidal integration to locate points along the streamline at regular increments in time, Δt, starting at a perpendicular field line ahead of the bow shock. Values for $\Delta\ell/\Delta\ell_\infty$ and ψ are calculated at the points where the perpendicular field lines and streamlines intersect, interpolating only along the field lines. A generalized quadrilateral interpolation scheme is then employed to determine $\Delta\ell/\Delta\ell_\infty$ and ψ at the computational grid points, using the quadrilateral containing the point formed by the intersection of pairs of adjacent streamlines and perpendicular field lines. At the bow shock, an exact formula is used, viz.

$$(\Delta\ell/\Delta\ell_\infty)^2 = 1 + \cot^2\theta(1+D^2) - 2D \times \csc\theta \times \cot\theta \times \cos(\theta-\delta)$$

$$\psi = \theta + \sin^{-1}\{D \times \cot\theta \times \sin(\theta-\delta)/(\Delta\ell/\Delta\ell_\infty)\}$$

where

$$D^2 = 1 - 4(M_\infty^2 \sin^2\theta - 1)(\gamma M_\infty^2 \sin^2\theta + 1)/\left((\gamma+1)^2 M_\infty^4 \sin^2\theta\right)$$

$$\cot\delta = \tan\theta \times \left\{(\gamma+1)M_\infty^2/\left[2(M_\infty^2 \sin^2\theta - 1)\right] - 1\right\}$$

$$\theta = \tan^{-1}\left\{\frac{dR_{shock}}{dx}\right\}$$

The values of $\Delta\ell/\Delta\ell_\infty$ at the grid points are smoothed using fifth-order least squares fit with respect to arc length along the arcs of the grid. The resultant magnetic field is expressed in orthogonal (x,y,z) components, where

$$B_x/B_\infty = \cos\alpha_n \times \left[\cos\phi \times \cos\alpha_p \times (|\underset{\sim}{B}|/B_\infty)_{\parallel} + \cos\psi \times \sin\alpha_p \times (|\underset{\sim}{B}|/B_\infty)_{\perp}\right]$$

$$B_y/B_\infty = \cos\alpha_n \times \left[\sin\phi \times \cos\alpha_p \times (|\underset{\sim}{B}|/B_\infty)_{\parallel} + \sin\psi \times \sin\alpha_p \times (|\underset{\sim}{B}|/B_\infty)_{\perp}\right]$$

$$B_z/B_\infty = \sin\alpha_n \times (B/B_\infty)_n$$

Magnetic field components may now be printed by subroutine BØUT, with LPRB as print control parameter. The magnetic field is not calculated when LPRB = .FALSE. and KBCØN=0.

Contours are calculated for velocity $|\underset{\sim}{v}|/v_\infty$, density ρ/ρ_∞, and magnetic field components $(\underset{\sim}{B}/B_\infty)_{\parallel}$ and $(\underset{\sim}{B}/B_\infty)_{\perp}$. The method used is a modified version of a procedure developed by R. Sorenson of NASA/Ames Research Center. The boundary is searched for intervals which bracket a contour point. Having found one point, the remainder of the contour is determined by 'walking' around the contour, searching at each step for the interval through which the contour line next passes, until a boundary point is reached. Then closed contours are found in a similar manner. Linear interpolation is used throughout the process. Note that since T/T_∞ is a function of $|\underset{\sim}{v}|/v_\infty$ only, velocity contours may also be considered as temperature contours. The coordinates of the contour lines can be printed by subroutine CØNØUT, with LPRCON as print control parameter. The section of the program which creates the plots is accessed only when the control parameter LPLOT = .TRUE.

A.2.2 Rerun Option

The rerun option is used when LRERUN = .TRUE. The blunt body and marching calculations are replaced with reading the grid coordinates and flow field values from the rerun file, TAPE4, which contains data written to TAPE9, then saved, on a previous run. Different values for any parameter not used in the flow field calculations may be specified, viz. contour values, plot length, magnetic field angles, and output options. Values of AMACH, GAMMA, and HRØ are required input, to ensure that the input rerun file does contain the case desired for rerun. If the geometry is user-supplied, the body shape table will be read from TAPE4, and should not be input from cards.

After reading the card input, AMACH, GAMMA, and HRØ are tested against values from TAPE4. The grid coordinates and flow field values from the blunt body calculation are read in, then smoothed, and streamlines calculated for this region, as previously described. The results of the marching calculation are then read, and the streamline calculation continued downstream. The calculations then proceed as described in section A.2.1.

A run must not contain more than one case which uses the rerun option.

A.2.3 Program Limitations and Precautions

The program makes some assumptions about the geometry of the obstacle shape around which flow is to be calculated, and about the flow field. The obstacle shape is assumed to be monotonically increasing in R, going downstream. The nose of the obstacle is at x = 1.0. The origin of the (x, R) coordinate system is the center of the planet. Obstacle shapes with sharp corners should be avoided. The flow at x = 0.0, which is used as the starting plane for the marching calculation, must be such that the axial (x-direction) Mach number is purely supersonic. This is a basic requirement of the marching code and is associated with the manner in which the code advances the solution. That condition is violated for ionopause shapes with H/RØ > 0.5, so that with the present joining procedure employed (i.e. at x = 0.0) those shapes cannot be treated. In the magnetic field calculation, the first streamline is assumed to be inside the arc described by the grid points immediately off the body, downstream of x = 0.0. To reduce computational costs, a grid using NR = 10 may be used, in which case a lower value of CN may be required. This would

reduce the running line by approximately 40 percent. A free stream Mach number less than 3.0 is not advised.

A.2.4 Convergence Criteria for Blunt Body Calculation

The output provides two measures of the convergence of the blunt body calculation. The RMS of shock speed and maximum shock speed are printed at each iteration. These quantities should both tend to zero. A value for q_{RMS}, RMS of shock speed, of

$$q_{RMS} < \sqrt{\gamma} \times M_\infty \times 10^{-3}$$

where γ is the specific heat ratio, and M_∞ is the free stream Mach number, usually indicates a converged solution. The RMS of error in enthalpy, HT, should be less than 1 percent, with the maximum enthalpy error also of that order.

The Courant number, CN, determines the time step size used by the calculation. A value not greater than the default of 3.0 should be used. For low Mach numbers or a coarser mesh than the default, a lower value may be preferable. If the default value does not generate a converged solution, or if the error message from subroutine SHØCK is printed, try lowering CN in increments of 0.5 to find a better value of CN. User-supplied bodies may also require a lower Courant number.

A.3 Description of Input

This section describes the card input for the program. An alphabetized dictionary of input variables is provided, defining the variables, listing default values and limitations. A discussion of the preparation of the card input is then presented, followed by a description of the input card format.

A.3.1 Dictionary of Input Variables

AMACH — free-stream Mach number; $3.0 \leq$ AMACH ≤ 25.0 is recommended

ANGN — the angle, in degrees, measuring the deviation of the free-stream magnetic field from the plane in which $\underset{\sim}{B}_{\infty\parallel}$ and $\underset{\sim}{B}_{\infty\perp}$ lie; equal to $\tan^{-1}(B_{\infty n}/\sqrt{|B_{\infty\parallel}|^2 + |B_{\infty\perp}|^2})$

ANGP — the angle, in degrees, measuring the deviation of the in-plane magnetic component $(\underset{\sim}{B}_{\infty\parallel} + \underset{\sim}{B}_{\infty\perp})$ from the direction of flow; equal to $\tan^{-1}(B_{\infty\perp}/B_{\infty\parallel})$

BCØN(I) KBCØN-dimensional array specifying values to be used for magnetic field strength contours

CN Courant number used for blunt body calculation; program default value is 3.0

GAMMA free-stream ratio of specific heats

HRØ obstacle geometry indicator:

HRØ > 0 — ionopause with H/R_O = HRØ
HRØ = 0 — magnetopause equatorial trace
HRØ < 0 — geometry is user-supplied

ITER integer, number of iterations for blunt body calculation; program default value is 300

KBCØN integer, number of values specified for magnetic field contours; $0 \leq KBCØN \leq 20$

KRCØN integer, number of values specified for density contours; $0 \leq KRCØN \leq 20$

KVCØN integer, number of values specified for velocity contours; $0 \leq KVCØN \leq 20$

LPLOT logical variable indicating whether to create plots or plot file

FALSE — no
TRUE — yes

LPRB logical variable indicating whether to print magnetic field output

FALSE — no
TRUE — yes

LPRCØN logical variable indicating whether to print coordinates of contour lines

FALSE — no
TRUE — yes

LPRFL logical variable indicating whether to print detailed flow field output

FALSE — no
TRUE — yes

LPRST logical variable indicating whether to print coordinates of streamlines

FALSE — no
TRUE — yes

LRERUN — logical variable indicating whether this case uses rerun option
FALSE – perform blunt body and marching calculations
TRUE – read results of a previous calculation from TAPE4

NBLUNT — integer, number of angular mesh points for blunt body calculation; for user-supplied geometry, XX(NBLUNT-1)=0.0; program default value, and maximum, is 24

NBØD — integer, number of points in body shape table when geometry is user-supplied; $1 \leq$ NBØD ≤ 100

NCASE — integer, number of cases to be run consecutively; NCASE ≥ 1

NR — integer, number of radial mesh points; program default value, and maximum, is 19

RCØN(I) — KRCØN – dimensional array specifying values to be used for density contours

RR(I) — NBØD – dimensional array representing the R-locations, in cylindrical (x,R) coordinates, of the user-supplied body shape

TITLE — descriptive heading of the case, to be printed on the first page of output; may contain up to 80 characters, including blanks

VCØN(I) — KVCØN – dimensional array specifying values to be used for velocity contours

XCALC — terminal downstream x-location for marching calculation of flow field; XCALC < 0.0; program default value is -1.0

XPLØT — terminal downstream x-location for calculation of streamlines, magnetic field, and contours; XCALC $\leq$ XPLØT < 0.0; program default value is -1.0

XX(I) — NBØD – dimensional array representing the x-locations, in cylindrical (x,R) coordinates, of the user-supplied body shape

A.3.2 Preparation of Input Data

The card input for a run consists of one card containing the number of cases to be run consecutively, Item 0, followed by a set of input for each case, Item 1 through Item 7, and Item 8 if required. Where a default value is to be used, the input field should be left blank. For each case, all required variables which do not assume their default values should be specified. The input format for all cards is described in section A.3.3.

Item 0 – This item consists of one card, containing the number of cases in this run, NCASE.

Item 1 – This card provides identification of the case, TITLE, which is printed on the first page of the output for this case.

Item 2 – This card contains information on the flow conditions and body geometry, and parameters required for the blunt body and marching calculations. AMACH, GAMMA, and HRØ must be specified for each case. For the rerun option, the values are tested against the values from the rerun file. The parameters XCALC, NR, NBLUNT, CN, ITER are used only when the flow field is to be calculated. These variables each assume a default value if the input field is blank.

Item 3 – This item consists of one card containing the rerun indicator, LRERUN, and the output control variables LPRFL, LPRST, LPRCØN, LPRB, and LPLØT.

Item 4 – This card contains the variables XPLØT, ANGP, and ANGN. The value for XPLØT is changed by the program to be the x-location of the marching calculation immediately upstream of the input value for XPLØT. The angles describing the deviation of the magnetic field from the flow, ANGP and ANGN, are not required when LPRB=.FALSE. and KBCØN=0, since the magnetic field is not calculated under these conditions. ANGP is the angle between the vectors $(\underset{\sim}{B}_{\infty\perp} + \underset{\sim}{B}_{\infty\parallel})$ and $\underset{\sim}{v}_{\infty}$, while ANGN is the angle between $\underset{\sim}{B}_{\infty}$ and $(\underset{\sim}{B}_{\infty\perp} + \underset{\sim}{B}_{\infty\parallel})$, where $\underset{\sim}{B}_{\infty\parallel}$, $\underset{\sim}{B}_{\infty\perp}$, $\underset{\sim}{B}_{\infty n}$ are the components of the free-stream magnetic field, $\underset{\sim}{B}_{\infty}$, which are parallel, perpendicular, and normal to $\underset{\sim}{v}_{\infty}$, and are as indicated in figure 4 of the text. The two angles ANGP and ANGN fully determine the half plane for which the magnetic field is to be calculated. The magnetic field for the other half of the plane may be calculated by rerunning with the sign of ANGP reversed. When $(\underset{\sim}{B}_{\infty\perp} + \underset{\sim}{B}_{\infty\parallel}) = 0$, ANGN = ±90°, ANGP = 0°; and, when $\underset{\sim}{B}_{\infty n} = 0$, ANGN = 0°.

Item 5 – This item contains the values for the velocity contours. The first card contains KVCØN, the number of values specified for VCØN. If KVCØN=0, only this card is required, and no velocity contours are calculated. If KVCØN > 0, the contour values are then read. Up to three cards may be required to accommodate the values, eight per card, maximum of 20. The contour values should be monotonically increasing, with at least one value within the range of the magnitude of the velocity in the region for which contours are to be calculated.

Item 6 – This item contains the values for the density contours. The description is similar to that for item 5, with KRCØN being the number of values specified, and RCØN the array of values.

Item 7 – This item contains the values for the magnetic field contours. The description is similar to that for item 5, with KBCØN

being the number of values specified, and BCØN the array of values. Note that the same contour values are used for the parallel and perpendicular components.

Item 8 – This optional item is required when HRØ < 0.0 and LRERUN= .FALSE., and contains the body shape table for the user-supplied geometry. The first card contains NBØD, the number of points in the shape table. The next NBØD cards contain the cylindrical (x,R) coordinates of these points, (XX(I), RR(I)), one point per card. The points supplied by the user determine the θ-spacing of the mesh used for the blunt body calculation. The first point should be near, but not on, the x-axis. A suggested location is such that the θ-spacing between the first point and the x-axis is half the θ-spacing between the first two points. The blunt body calculation adds a point which is the reflection about the x-axis of the first point in the body shape table. The $(NBLUNT-1)^{th}$ point should be at x=0.0. The $BLUNT^{th}$ point is also used to create the grid for the blunt body calculation. The coordinates must be normalized so that the planet center is at (0.,0.) and the nose of the body at (1.,0.).

A.3.3 Format of Input Data

Four format types are used for the input data. For real numbers (F-format), a decimal point is required. Integers (I-format) should be right-adjusted in the field. For logical variables (L-format), the first non-blank character in the field, which should be 'T' or 'F', determines the value. Note that a blank input field is interpreted as 'FALSE'. The title, which is in A-format, may contain any valid character.

A description of the card format of the input data follows, with item numbers corresponding to those in section A.3.2:

Item No. 0: 1 card

Variable	NCASE
Card Column	10
Format Type	I

Item No. 1: 1 card

Variable	TITLE
Card Column	1 ... 80
Format Type	A

Item No. 2: 1 card

Variable	AMACH	GAMMA	HRØ	XCALC	NR	NBLUNT	CN	ITER
Card Column	10	20	30	40	50	60	70	80
Format Type	F	F	F	F	I	I	F	I

Item No. 3: 1 card

Variable	LRERUN	LPRL	LPRST	LPRCØN	LPRB	LPLØT
Card Column	10	20	30	40	50	60
Format Type	L	L	L	L	L	L

Item No. 4: 1 card

Variable	XPLØT	ANGP	ANGN
Card Column	10	20	30
Format Type	F	F	F

Item No. 5:

Variable	KVCØN
Card Column	10
Format Type	I

(1) 1 card

(2) 0 to 3 cards as needed for up to 20 values, 8 per card

Variable	VCØN(1)	VCØN(2)			VCØN(KVCØN)			
Card Column	10	20	30	40	50	60	70	80
Format Type	F	F	F	F	F	F	F	F

Input Data Card Format

Item No. 6:

(1) 1 card

Variable	KRCØN
Card Column	10
Format Type	I

(2) 0 to 3 cards as needed for up to 20 values, 8 per card

Variable	RCØN(1)	RCØN(2)			RCØN(KRCØN)			
Card Column	10	20	30	40	50	60	70	80
Format Type	F	F	F	F	F	F	F	F

Item No. 7:

(1) 1 card

Variable	KRCØN
Card Column	10
Format Type	I

(2) 0 to 3 cards as needed for up to 20 values, 8 per card

Variable	BCØN(1)	BCØN(2)			BCØN(KBCØN)			
Card Column	10	20	30	40	50	60	70	80
Format Type	F	F	F	F	F	F	F	F

Item No. 8: (required only when HRØ < 0.0 and LRERUN=.FALSE.)

(1) 1 card

Variable	NBØD
Card Column	10
Format Type	I

(2) I = 1 to NBØD; NBØD cards

Variable	XX(I)	RR(I)
Card Column	10	20
Format Type	F	F

A.4 Description of Output

This section describes the output of the computer program. The contents of each output item are specified and discussed. The printed output consists of six items, four of which are optional, with input control parameters. Plotted output is also optional.

The first output item consists of the input data. The title for the case is printed first, followed by all input variables, with identification of each variable. Default values are printed as if they were input. Parameters CN, NR, NBLUNT, ITER for the blunt body calculation and XCALC, the terminal location for the marching calculation, are printed only when the flow field is to be calculated. When the geometry is user-supplied, the input body shape table is printed. For a default geometry, the body shape is indicated by the description "default magnetopause coordinates – equatorial trace" or "default ionopause coordinates – H/RØ = ".

The second output item is not printed when LRERUN=.TRUE. From the blunt body calculation, the shock speed at each iteration, the final enthalpy error, final sonic line location, and body and final bow shock shape are printed. For convergence criteria of this calculation, the downstream location and body and shock ordinates are output. There is no control variable allowing the user to suppress this item of output when the flow field is calculated.

Detailed flow field output is the third item, and is printed only when LPRFL=.TRUE. Coordinates are labeled as X/D, R/D, RP/D, or X/RO, R/RO, RP/RO, to emphasize that distances are normalized by the distance from the center of the planet to the nose of the body, D for the magnetopause, RO for an ionopause. Along the symmetry axis, the values printed are velocity magnitude V/VINF, density RHØ/RHØINF, temperature T/TINF, and pressure P/PINF. Over the rest of the flow field values are also given for velocity components VX/VINF, VR/VINF, and flow angle ϕ. Note that the flow angle is the deviation of the flow about the obstacle, and so $0° \leq \phi < 90°$.

The next output item is the (x,R) coordinates of the streamlines. For the blunt body region, the (R_p,θ) coordinates of the starting position on the bow shock wave are also given. This item is printed only when LPRST=.TRUE.

The magnetic field components are then printed, if LPRB=.TRUE. . The location of each point is defined in (R_p,θ) coordinates for the blunt body region, and (x,R) coordinates for the downstream marching region. The components along field lines parallel, perpendicular, and normal to the flow in the free stream are printed as B/BINF(PARALLEL), B/BINF(PERP), B/BINF(NØRMAL). The orthogonal (x,y,z) components of the resultant are printed as BX/BINF(RESULTANT), BY/BINF(RESULTANT), BZ/BINF(RESULTANT). The magnetic field in the symmetry (x,y) plane, defined by the vector sum $\{(\underset{\sim}{B}/B_\infty)_{\shortparallel} + (\underset{\sim}{B}/B_\infty)_{\perp}\}$, is also printed, and is given by the magnitude B/BINF(IN-PLANE) and direction B-ANGLE(IN-PLANE) of the vector. The B-ANGLE ψ is the angle between the magnetic field and the flow, and generally has the same sign as ANGP.

The last item printed is the (x,R) coordinates of the contours, for which LPRCØN is the logical control variable. Noting that temperature and velocity contours coincide, the corresponding value of T/TINF is printed along with V/VINF for the velocity contours. There are three nonfatal error messages which may occur – see section A.5.

The program also has the capability of producing plotted output, using UCC plot routines AXIS, CHAR, DASH, DØTLN, ENPLT, GREEK, MATH, NUMPLT, PLØT, PLTLN, PØLAR, SCALF, VECTØR. Plots are not produced when LPLØT=.FALSE. The plots provide pictorial representation of the streamlines and contours, with a maximum of six frames produced. The first frame is a plot of the streamlines. Then contour plots of velocity, temperature, and density are drawn. The last two frames are contour plots of the parallel and perpendicular magnetic field components, with field lines added to indicate direction. Frames are omitted when no contour levels are provided.

A.5 Program Error Messages

This section lists the messages printed by the program, and indicates what action should be taken by the user.

(1) ***** EXECUTIØN TERMINATED *****
RERUN DATA ØN TAPE4 DØES NØT AGREE
WITH CASE SPECIFIED ØN CARD INPUT:
MACH NØ. GAMMA M/RO

FRØM CARDS
FRØM TAPE4

The first three parameters of item 2 of the input for a case using the rerun option should agree with those used when creating the file. The tolerance used in comparing the values is 10^{-5}. For a user-supplied geometry, it is sufficient for both values of H/R_o to be negative.

(2) ***** EXECUTIØN TERMINATED *****
ARRAY ØF CØNTØUR VALUES IMPRØPERLY SPECIFIED

When specified, the contour values should be monotonically increasing, with at least one value in the range of the velocity, density, or magnetic field strength for the region under consideration. This error does not inhibit generation of the rerun file.

(3) CØNTØUR SEARCH ABØRTED — TABLE ØVERFLØW IN NAD

The program allows for 29 contour lines to be found, storing the starting address of each contour in array NAD. This message indicates that at least one more contour line could be found. If the user requires all the contours of the levels specified, the case should be rerun in two parts. Otherwise, reduce the number of contour levels specified.

(4) CØNTØUR SEARCH ABØRTED — TABLE ØVERFLØW IN (X,Y)

The contour lines may be described by up to 1000 points, stored in arrays X and Y. This message indicates that more points would be required for the contour lines requested. The last contour line found will be incomplete. As with (3), either reduce the number of contour levels or run as two cases.

(5) NEGATIVE PRESSURE DETECTED BY SHØCK AT J=
PN= PØ= PTAU=

This message is printed when a negative pressure has been calculated at the shock on this iteration, at radial location J. The quantities printed are: PN, the pressure calculated on this step; PØ, the pressure from the previous step; and PTAU, the partial derivative of pressure with

respect to time. This condition indicates that the shock wave motion is too extreme. Lowering the value of CN, and thus reducing the time step, may remove the problem.

The following messages (6)-(10) usually result from using an obstacle geometry which is in some way too severe for the program to handle in its present form. The obstacle slope may be sufficiently high at x = 0.0 that the axial Mach number becomes subsonic in the starting solution for the marching calculation, or there may be a sharp corner in the profile. Check input, particularly free stream Mach number and body geometry.

(6) NEGATIVE PRESSURE ØN BØDY DETECTED BY BNDRY, PB= AT J=

This message indicates that a negative pressure on the body, PB, has been calculated at radial location J.

(7) NEGATIVE PRESSURE ØR DENSITY ØN BØDY DETECTED BY BNDRYM AT X=
PB= RHØB= VXB= VRB=

The program makes internal corrections when this condition occurs, resulting pressure PB, density RHØB, and velocity components VXB and VRB.

(8) NEGATIVE SIGMA-BAR-1 IN EIGENM INDICATES SUBSØNIC FLØW AT I=

(9) NEGATIVE SIGMA-BAR-2 IN EIGENM INDICATES SUBSØNIC FLØW AT I=

These messages are printed when subsonic flow is detected by the marching calculation. The computed stepsize for this region will be quite small.

(10) ----------BODY TURN STOPPED AT M2=100----------

This message indicates that the body has a sharp corner, which has been limited to 100° when being transformed.

A.6 Sample Case

A sample case is presented in this section. Figures A.1, A.2, and A.3 illustrate the input data, portions of the printed output, and the plotted output, respectively.

The sample case is run alone and is set up so as to produce all possible output. Flow is to be calculated about the rotated equatorial

trace of the magnetopause to a downstream location of x = -10.0, with M_∞ = 8, γ = 5/3. Streamlines, magnetic field components, and contours are desired to a downstream location of x = -5.0. The magnetic field forms angles of α_p = -45°, α_n = 45° with the solar wind flow. Contour values are specified for all quantities.

The input data is tabulated in figure A.1, with item numbers corresponding to those in sections A.3.2 and A.3.3. The first card, item 0, indicates that there is one case to be run. The remaining 13 cards provide the data required for this case. Item 1 contains the identifying title. On the next card, item 2, values are specified for AMACH, GAM, HRØ, and XCALC. The other data fields are left blank to indicate that the default values will be used. The values of the logical variables of item 3 specify that the flow field is to be calculated, and that full printed and plotted output is to be produced. Item 4 defines the plot length to be -5.0, and the magnetic field angles as ANGP = -45.0 and ANGN = 45.0. Items 5, 6, 7 specify the contour levels to be used – 12 for velocity and temperature, 10 for density, and 12 for magnetic field strength. Item 8 is omitted, since the obstacle geometry is one of the default shapes for which the coordinates are calculated internally by the program.

Figure A.2 presents portions of the printed output from the sample case using the data deck of figure A.1. The full printed output for this case is approximately 5,000 lines. The first item of printed output, figure A.2(a), is the input variables. Then in figure A.2(b), the output from the blunt body calculation is shown, omitting shock speed values for iterations 6 to 295. Note that the convergence criteria described in section A.2.4 are satisfied. Figure A.2(c) provides the output from the marching calculation. Figures A.2(d),(e),(f) show the detailed flow field output in four areas: near the symmetry axis; near x = 0.0, when the two calculation schemes are joined; around x = -5.0, the value of XPLØT; and the last two locations calculated. An example of the streamline output is shown in figures A.2(g),(h). Figures A.2(h),(i),(j) provide the magnetic field components at the same locations chosen to illustrate the flow field output. Examples of the printed output of the contour lines are shown in figures A.2(k),(l),(m), choosing two contour lines for each quantity. Note that for $(|\underset{\sim}{B}|/B_\infty)_\perp$, the different contour lines for the same contour level are considered to be two separate and distinct lines.

Figure A.3 shows the six plots which are produced by the program for this case.

Column No.	10	20	30	40	50	60	70	80
Item No.								
0	1							
1	SAMPLE CASE		DEFAULT MAGNETØPAUSE EQUATØRIAL TRACE					
2	8.0	1.66666667	0.0	-10.0				
3	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE		
4	-5.0	-45.0	45.0					
5	12							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.75
	0.8	0.85	0.88	0.9				
6	10							
	0.8	1.2	1.6	2.0	2.5	3.0	3.5	3.8
	4.0	4.2						
7	12							
	0.5	0.75	1.0	1.25	1.5	2.0	2.5	3.0
	3.5	4.0	5.0	6.0				

Figure A-1. - Card input for sample case.

FIGURE A.2

ABBREVIATED OUTPUT FOR SAMPLE CASE

SAMPLE CASE DEFAULT MAGNETOPAUSE EQUATORIAL TRACE

INPUT VARIABLES
****

FREESTREAM MACH NO. = 8.00

SPECIFIC HEAT RATIO = 1.667

OBSTACLE GEOMETRY: DEFAULT MAGNETOPAUSE COORDINATES - EQUATORIAL TRACE

PARAMETERS FOR BLUNT BODY CALCULATION

NO. OF RADIAL MESH POINTS = 19
NO. OF ANGULAR MESH POINTS = 24
COURANT NUMBER = 3.[illegible]
NO. OF ITERATIONS = 3[illegible]

TERMINAL DOWNSTREAM LOCATION FOR MARCHING CALCULATION =-16.00

TERMINAL DOWNSTREAM LOCATION FOR PLOTTING = -5.[illegible]

IN-PLANE DEVIATION OF MAGNETIC FIELD FROM FLOW = -45.00 DEGREES

DEVIATION OF MAGNETIC FIELD FROM PLANE OF FLOW = 45.00 DEGREES

LRERUN = F

LPRFL = T

LPRST = T

LPRCON = T

LPRB = T

LPLOT = T

VALUES SPECIFIED FOR CONTOUR CALCULATION
****

12 CONTOUR LEVELS FOR VELOCITY:

.2[illegible]	.2[illegible]	.3[illegible]	.400	.500	.600
.700	.7[illegible]	.8[illegible]	.8[illegible]	.8[illegible]	.9[illegible]

12 CONTOUR LEVELS FOR DENSITY:

.800	1.200	1.600	2.000	2.500	3.000
3.500	3.8[illegible]	4.[illegible]	4.200		

12 CONTOUR LEVELS FOR MAGNETIC FIELD STRENGTH:

.500	.750	1.000	1.250	1.500	2.000
2.500	3.000	3.500	4.000	5.000	6.000

Figure A.2(a)

BLUNT BODY CALCULATION
****

```
ITERATION   1    RMS OF SHOCK SPEED= 1.2458E-02   MAXIMUM SHOCK SPEED= 3.6519E-02 AT J=25
ITERATION   2    RMS OF SHOCK SPEED= 1.2445E-02   MAXIMUM SHOCK SPEED= 3.1748E-02 AT J=25
ITERATION   3    RMS OF SHOCK SPEED= 2.1994E-02   MAXIMUM SHOCK SPEED= 3.2075E-02 AT J=25
ITERATION   4    RMS OF SHOCK SPEED= 3.4842E-02   MAXIMUM SHOCK SPEED=-5.037CE-02 AT J=16
ITERATION   5    RMS OF SHOCK SPEED= 4.9444E-02   MAXIMUM SHOCK SPEED=-7.2259E-02 AT J=16

ITERATION 296    RMS OF SHOCK SPEED= 3.790BE-03   MAXIMUM SHOCK SPEED= 9.6232E-03 AT J=15
ITERATION 297    RMS OF SHOCK SPEED= 3.7524E-03   MAXIMUM SHOCK SPEED= 9.1112E-03 AT J=15
ITERATION 298    RMS OF SHOCK SPEED= 3.7140E-03   MAXIMUM SHOCK SPEED=-9.1274E-03 AT J=17
ITERATION 299    RMS OF SHOCK SPEED= 3.6752E-03   MAXIMUM SHOCK SPEED=-8.9772E-03 AT J=17
ITERATION 300    RMS OF SHOCK SPEED= 3.6352E-03   MAXIMUM SHOCK SPEED=-8.6374E-03 AT J=14
```

FINAL SONIC LINE LOCATION

```
XSL=  .5982   RSL=  .9449
XSL=  .6171   RSL=  .9570
XSL=  .6347   RSL=  .9702
XSL=  .6540   RSL=  .9812
XSL=  .6742   RSL=  .9905
XSL=  .6959   RSL=  .9985
XSL=  .7187   RSL= 1.0043
XSL=  .7425   RSL= 1.0086
XSL=  .7665   RSL= 1.0110
XSL=  .7914   RSL= 1.0104
XSL=  .8172   RSL= 1.0091
XSL=  .8437   RSL= 1.0053
XSL=  .8694   RSL= 1.0024
XSL=  .8967   RSL=  .9951
XSL=  .9240   RSL=  .9871
XSL=  .9518   RSL=  .9776
XSL=  .9795   RSL=  .9661
XSL= 1.0077   RSL=  .9529
XSL= 1.0370   RSL=  .9372
```

<<<< % ERROR IN HT = .5822E+00   RMS OF % ERROR IN HT = .6475E-01 >>>>

BODY AND FINAL SHOCK SHAPE

J	X(BODY)	R(BODY)	X(SHOCK)	R(SHOCK)
2	-.9995	.0349	-1.2735	.0444
3	-.9957	.1046	-1.2662	.1331
4	-.9879	.1742	-1.2592	.2216
5	-.9764	.2434	-1.2456	.3136
6	-.9610	.3122	-1.2292	.3994
7	-.9417	.3805	-1.2082	.4882
8	-.9188	.4482	-1.1834	.5772
9	-.8925	.5147	-1.1537	.6661
10	-.8607	.5808	-1.1220	.7554
11	-.8250	.6453	-1.0814	.8449
12	-.7874	.7069	-1.0383	.9349
13	-.7446	.7713	-.9898	1.0250
14	-.6994	.8323	-.9357	1.1153
15	-.6480	.8929	-.8777	1.2082
16	-.5935	.9499	-.8130	1.3013
17	-.5358	1.0062	-.7418	1.3951
18	-.4723	1.0609	-.6648	1.4924
19	-.4054	1.1137	-.5793	1.5886
20	-.3346	1.1648	-.4864	1.6892
21	-.2584	1.2138	-.3859	1.7925
22	-.1772	1.2610	-.2668	1.8987
23	-.0913	1.3067	-.1495	2.0088
24	.0000	1.3491	-.0000	2.1245

Figure A.2(b) - continued

MARCHING CALCULATION
****

STEP NO.	DOWNSTREAM LOCATION	BODY ORDINATE	SHOCK ORDINATE
1	-.6302	1.3622	2.1483
2	-.6604	1.3751	2.1719
3	-.6906	1.3879	2.1951
4	-.1206	1.3993	2.2181
5	-.1510	1.4108	2.2408
6	-.1812	1.4219	2.2633
7	-.2114	1.4326	2.2855
8	-.2416	1.4430	2.3074
9	-.2718	1.4530	2.3292
10	-.3253	1.4699	2.3671
11	-.3786	1.4858	2.4043
12	-.4322	1.5008	2.4409
13	-.4857	1.5149	2.4769
14	-.5392	1.5282	2.5123
15	-.5926	1.5408	2.5472
16	-.6461	1.5526	2.5815
17	-.6996	1.5638	2.6153
18	-.7530	1.5743	2.6487
19	-.8065	1.5843	2.6816
20	-.8964	1.5998	2.7359
21	-.9863	1.6139	2.7892
22	-1.0762	1.6267	2.8412
23	-1.1661	1.6384	2.8922
24	-1.2560	1.6491	2.9423
25	-1.3459	1.6587	2.9914
26	-1.4358	1.6676	3.0397
27	-1.5257	1.6757	3.0872
28	-1.6156	1.6831	3.1339
29	-1.7055	1.6899	3.1799
30	-1.8527	1.6999	3.2537
31	-1.9999	1.7086	3.3257
32	-2.1471	1.7161	3.3962
33	-2.2943	1.7228	3.4651
34	-2.4415	1.7287	3.5327
35	-2.5887	1.7338	3.5989
36	-2.7359	1.7384	3.6640
37	-2.8831	1.7425	3.7278
38	-3.0303	1.7462	3.7906
39	-3.1775	1.7494	3.8524
40	-3.4261	1.7542	3.9545
41	-3.6747	1.7582	4.0541
42	-3.9232	1.7616	4.1513
43	-4.1718	1.7645	4.2465
44	-4.4204	1.7670	4.3396
45	-4.6689	1.7692	4.4312
46	-4.9175	1.7711	4.5206
47	-5.1661	1.7726	4.6086
48	-5.4147	1.7742	4.6951
49	-5.6632	1.7753	4.7802
50	-6.0798	1.7771	4.9198
51	-6.4963	1.7786	5.0561
52	-6.9128	1.7799	5.1894
53	-7.3293	1.7809	5.3198
54	-7.7458	1.7818	5.4476
55	-8.1623	1.7826	5.5730
56	-8.5788	1.7833	5.6961
57	-8.9954	1.7839	5.8173
58	-9.4119	1.7844	5.9364
59	-9.8284	1.7848	6.0538
60	-10.2500	1.7854	6.2394

Figure A.2(c) - continued

DETAILED FLOW FIELD OUTPUT
****

FLOW FIELD VALUES EXTRAPOLATED TO SYMMETRY AXIS, THETA = 0.00 DEGREES

I	X/D	V/VINF	RHO/RHOINF	T/TINF	P/PINF
1	1.0050	0.0110	4.2291	22.3333	94.4497
2	1.0150	.0164	4.2220	22.3276	94.2214
3	1.0301	.0217	4.2156	22.3133	94.0645
4	1.0451	.0271	4.2089	22.2899	93.8164
5	1.0602	.0597	4.1998	22.2572	93.4766
6	1.0752	.0745	4.1883	22.2149	93.0490
7	1.0903	.0894	4.1744	22.1629	92.5177
8	1.1053	.1043	4.1582	22.1012	91.9039
9	1.1204	.1192	4.1395	22.0300	91.1941
10	1.1354	.1341	4.1185	21.9494	90.3992
11	1.1505	.1490	4.0951	21.8599	89.5189
12	1.1655	.1637	4.0693	21.7618	88.5562
13	1.1805	.1782	4.0412	21.6558	87.5148
14	1.1956	.1925	4.0106	21.5424	86.3989
15	1.2106	.2066	3.9777	21.4225	85.2129
16	1.2257	.2204	3.9424	21.2969	83.9618
17	1.2407	.2339	3.9048	21.1666	82.6508
18	1.2558	.2469	3.8647	21.0326	81.2854
19	1.2708	.2617	3.8209	20.8721	79.7593

FLOW FIELD VALUES FROM BLUNT BODY CALCULATION

ANGULAR LOCATION NO. 2, AT THETA = 2.0000 DEGREES

I	RP/D	R/D	X/D	VR/VINF	VX/VINF	FLOW ANGLE	V/VINF	RHO/RHOINF	T/TINF	P/PINF
1	1.0071	.0349	.9995	.0117	.0003	88.4170	.0103	4.2264	22.3311	94.3802
2	1.0152	.0354	1.0146	.0179	.0159	48.3361	.0245	4.2238	22.3206	94.2596
3	1.0302	.0360	1.0296	.0198	.0315	32.0664	.0386	4.2175	22.3016	94.0567
4	1.0453	.0365	1.0446	.0194	.0486	21.7493	.0526	4.2099	22.2742	93.7714
5	1.0603	.0370	1.0597	.0184	.0638	16.1277	.0667	4.2001	22.2384	93.4034
6	1.0754	.0375	1.0747	.0181	.0788	12.9421	.0808	4.1882	22.1941	92.9527
7	1.0904	.0380	1.0898	.0179	.0934	10.8578	.0948	4.1740	22.1415	92.4194
8	1.1055	.0386	1.1048	.0177	.1079	9.3428	.1088	4.1577	22.0806	91.8036
9	1.1205	.0391	1.1199	.0176	.1221	8.1878	.1228	4.1390	22.0114	91.1056
10	1.1356	.0396	1.1349	.0174	.1361	7.2955	.1368	4.1181	21.9339	90.3258
11	1.1507	.0402	1.1499	.0173	.1501	6.5784	.1508	4.0948	21.8482	89.4648
12	1.1657	.0407	1.1650	.0172	.1639	5.9916	.1648	4.0692	21.7542	88.5233
13	1.1808	.0412	1.1800	.0171	.1778	5.4997	.1787	4.0413	21.6521	87.5019
14	1.1958	.0417	1.1951	.0171	.1916	5.0828	.1926	4.0109	21.5419	86.4017
15	1.2109	.0423	1.2101	.0170	.2054	4.7228	.2065	3.9780	21.4235	85.2237
16	1.2259	.0428	1.2252	.0169	.2193	4.4113	.2204	3.9428	21.2970	83.9689
17	1.2410	.0433	1.2402	.0169	.2334	4.1354	.2343	3.9050	21.1625	82.6387
18	1.2560	.0438	1.2553	.0169	.2474	3.9001	.2481	3.8646	21.0200	81.2346
19	1.2711	.0444	1.2703	.0169	.2621	3.6835	.2620	3.8218	20.8695	79.7581

ANGULAR LOCATION NO. 3, AT THETA = 6.0000 DEGREES

I	RP/D	R/D	X/D	VR/VINF	VX/VINF	FLOW ANGLE	V/VINF	RHO/RHOINF	T/TINF	P/PINF
1	1.0021	.1046	.9957	.0350	.0029	85.2522	.0409	4.2139	22.2977	93.9609
2	1.0153	.1058	1.0107	.0543	.0185	71.1386	.0525	4.2102	22.2745	93.7792
3	1.0304	.1078	1.0257	.0585	.0369	58.4397	.0642	4.2045	22.2454	93.5316
4	1.0465	.1094	1.0408	.0573	.0516	48.0188	.0760	4.1970	22.2102	93.2169
5	1.0616	.1110	1.0558	.0559	.0670	39.4164	.0878	4.1875	22.1689	92.8335
6	1.0767	.1125	1.0708	.0541	.0817	33.5384	.0997	4.1761	22.1212	92.3799
7	1.0918	.1141	1.0859	.0536	.0963	29.0690	.1117	4.1625	22.0670	91.8546
8	1.1069	.1157	1.1009	.0531	.1107	25.6199	.1239	4.1469	22.0058	91.2559
9	1.1221	.1173	1.1159	.0526	.1249	22.8317	.1362	4.1291	21.9376	90.5823
10	1.1372	.1189	1.1309	.0522	.1389	20.5983	.1487	4.1391	21.8619	89.8321
11	1.1522	.1204	1.1460	.0519	.1529	18.7485	.1613	4.0868	21.7784	89.0035
12	1.1674	.1220	1.1610	.0516	.1667	17.2044	.1741	4.0622	21.6886	88.0951
13	1.1825	.1236	1.1760	.0514	.1806	15.8878	.1871	4.0352	21.5863	87.1050
14	1.1976	.1252	1.1911	.0512	.1944	14.7565	.2004	4.0058	21.4768	86.0317
15	1.2128	.1268	1.2061	.0510	.2083	13.7666	.2139	3.9739	21.3576	84.8737
16	1.2279	.1283	1.2211	.0509	.2222	12.8966	.2276	3.9395	21.2282	83.6293
17	1.2430	.1299	1.2362	.0507	.2364	12.1097	.2416	3.9025	21.0881	82.2972
18	1.2581	.1315	1.2512	.0506	.2514	11.4144	.2559	3.8629	20.9364	80.8760
19	1.2732	.1331	1.2662	.0503	.2652	10.7386	.2705	3.8206	20.7726	79.3645

Figure A.2(d) - continued

ANGULAR LOCATION NO.23, AT THETA = 86.[illegible] DEGREES

I	RP/D	R/D	X/D	VR/VINF	VX/VINF	FLOW ANGLE	V/VINF	RHO/RHOINF	T/TINF	P/PINF
1	1.3092	1.3066	.0913	.3105	.6243	26.4427	.6975	1.6544	11.9557	19.7801
2	1.3484	1.3451	.0941	.3133	.6233	26.6813	.6974	1.7271	11.9569	20.6503
3	1.3875	1.3841	.0968	.3124	.6248	26.5626	.6983	1.8011	11.9305	21.4879
4	1.4256	1.4232	.0995	.3133	.6251	26.5618	.7000	1.8770	11.8787	22.2965
5	1.4658	1.4622	.1022	.3140	.6283	26.5547	.7026	1.9554	11.8036	23.0794
6	1.5049	1.5013	.1050	.3154	.6313	26.5492	.7058	2.0367	11.7052	23.8401
7	1.5441	1.5403	.1077	.3172	.6348	26.5527	.7097	2.1215	11.5870	24.5817
8	1.5832	1.5793	.1104	.3194	.6388	26.5638	.7142	2.2102	11.4502	25.3076
9	1.6223	1.6184	.1132	.3219	.6433	26.5802	.7193	2.3035	11.2962	26.0208
10	1.6615	1.6574	.1159	.3246	.6481	26.6045	.7248	2.4017	11.1270	26.7242
11	1.7006	1.6965	.1186	.3275	.6532	26.6338	.7307	2.5055	10.9441	27.4207
12	1.7397	1.7355	.1214	.3308	.6586	26.6691	.7369	2.6153	10.7494	28.1133
13	1.7789	1.7746	.1241	.3341	.6641	26.7088	.7434	2.7317	10.5446	28.8048
14	1.8180	1.8136	.1268	.3376	.6698	26.7519	.7501	2.8551	10.3317	29.4986
15	1.8571	1.8526	.1295	.3412	.6755	26.7986	.7569	2.9862	10.1126	30.1980
16	1.8963	1.8917	.1323	.3449	.6813	26.8469	.7637	3.1253	9.8894	30.9071
17	1.9354	1.9307	.1350	.3485	.6872	26.8922	.7706	3.2730	9.6641	31.6306
18	1.9745	1.9697	.1377	.3522	.6931	26.9339	.7774	3.4299	9.4389	32.3742
19	2.0137	2.0088	.1405	.3560	.6987	26.9992	.7841	3.5964	9.2161	33.1445

ANGULAR LOCATION NO.24, AT THETA = 90.0000 DEGREES

I	RP/D	R/D	X/D	VR/VINF	VX/VINF	FLOW ANGLE	V/VINF	RHO/RHOINF	T/TINF	P/PINF
1	1.3491	1.3491	-.0[illegible]	.2924	.6563	24.0135	.7185	1.5205	11.3291	17.2127
2	1.3922	1.3922	-.0[illegible]	.2986	.6540	24.3941	.7180	1.5937	11.3353	18.0653
3	1.4352	1.4352	-.0[illegible]	.2979	.6544	24.4082	.7185	1.6685	11.3199	18.8878
4	1.4783	1.4783	-.0[illegible]	.2987	.6550	24.5124	.7199	1.7456	11.2760	19.6836
5	1.5214	1.5214	-.0[illegible]	.3007	.6564	24.6095	.7222	1.8255	11.2059	20.4563
6	1.5645	1.5645	-.0[illegible]	.3030	.6589	24.6965	.7253	1.9088	11.1114	21.2095
7	1.6076	1.6076	-.0[illegible]	.3056	.6618	24.7844	.7290	1.9961	10.9947	21.9464
8	1.6506	1.6506	-.0[illegible]	.3085	.6654	24.8727	.7334	2.0880	10.8577	22.6704
9	1.6937	1.6937	-.0000	.3117	.6695	24.9622	.7384	2.1850	10.7025	23.3846
10	1.7368	1.7368	-.0[illegible]	.3151	.6739	25.0535	.7438	2.2877	10.5309	24.0920
11	1.7799	1.7799	-.0[illegible]	.3186	.6787	25.1466	.7496	2.3968	10.3451	24.7955
12	1.8230	1.8230	-.0000	.3223	.6837	25.2414	.7558	2.5129	10.1471	25.4981
13	1.8661	1.8661	-.0000	.3262	.6889	25.3375	.7622	2.6364	9.9391	26.2029
14	1.9091	1.9091	-.0000	.3302	.6943	25.4343	.7688	2.7679	9.7233	26.9135
15	1.9522	1.9522	-.0000	.3342	.6997	25.5315	.7755	2.9082	9.5020	27.6336
16	1.9953	1.9953	-.0000	.3383	.7052	25.6292	.7823	3.0577	9.2776	28.3681
17	2.0384	2.0384	.0[illegible]	.3424	.7107	25.7247	.7890	3.2170	9.0525	29.1223
18	2.0814	2.0814	.0[illegible]	.3465	.7162	25.8168	.7956	3.3868	8.8294	29.9034
19	2.1245	2.1245	.0[illegible]	.3505	.7216	25.9033	.8020	3.5676	8.6138	30.7195

FLOW FIELD VALUES FROM MARCHING CALCULATION

ADDITIONAL AXIAL LOCATION NO. 1, AT X/D = -.0302

I	R/D	VR/VINF	VX/VINF	FLOW ANGLE	V/VINF	RHO/RHOINF	T/TINF	P/PINF
1	1.3623	.2870	.6670	23.2802	.7251	1.4791	11.0865	16.3985
2	1.4066	.2906	.6663	23.6535	.7262	1.5463	11.1463	17.2359
3	1.4497	.2923	.6635	23.7733	.7256	1.6289	11.1284	18.1135
4	1.4933	.2941	.6639	23.8961	.7261	1.7062	11.0855	18.9140
5	1.5370	.2965	.6667	24.0394	.7278	1.7885	11.0321	19.7305
6	1.5807	.2992	.6666	24.1753	.7306	1.8736	10.9656	20.5369
7	1.6243	.3022	.6692	24.3023	.7343	1.9621	10.8813	21.2519
8	1.6680	.3054	.6725	24.4243	.7386	2.0550	10.6965	21.9810
9	1.7117	.3088	.6762	24.5438	.7434	2.1531	10.5439	22.7019
10	1.7553	.3124	.6804	24.6629	.7487	2.2571	10.3757	23.4198
11	1.7990	.3162	.6849	24.7817	.7543	2.3677	10.1939	24.1363
12	1.8427	.3201	.6896	24.9012	.7603	2.4856	10.0006	24.8574
13	1.8863	.3242	.6946	25.0183	.7665	2.6113	9.7979	25.5854
14	1.9300	.3283	.6998	25.1355	.7730	2.7454	9.5877	26.3217
15	1.9737	.3325	.7050	25.2515	.7795	2.8882	9.3715	27.0671
16	2.0173	.3366	.7103	25.3662	.7861	3.0407	9.1511	27.8254
17	2.0610	.3407	.7156	25.4784	.7927	3.2035	8.9283	28.6019
18	2.1047	.3452	.7209	25.5870	.7993	3.3775	8.7051	29.4012
19	2.1483	.3492	.7261	25.6889	.8057	3.5617	8.4847	30.2195

ADDITIONAL AXIAL LOCATION NO. 2, AT X/D = -.0694

I	R/D	VR/VINF	VX/VINF	FLOW ANGLE	V/VINF	RHO/RHOINF	T/TINF	P/PINF
1	1.3701	.2811	.6770	22.3497	.7331	1.4358	10.8690	15.6058
2	1.4194	.2868	.6728	22.9425	.7306	1.5056	10.9463	16.4854
3	1.4636	.2872	.6717	23.1539	.7306	1.5925	10.9472	17.4339
4	1.5079	.2896	.6717	23.3241	.7314	1.6709	10.9198	18.2457
5	1.5522	.2926	.6724	23.4962	.7332	1.7523	10.8650	19.0386
6	1.5964	.2954	.6738	23.6725	.7357	1.8388	10.7852	19.8318
7	1.6407	.2987	.6761	23.8391	.7392	1.9287	10.6780	20.5943
8	1.6850	.3022	.6790	23.9938	.7433	2.0229	10.5482	21.3378
9	1.7292	.3059	.6825	24.1426	.7479	2.1224	10.3998	22.0720
10	1.7735	.3098	.6864	24.2880	.7531	2.2278	10.2352	22.8024
11	1.8178	.3135	.6907	24.4339	.7586	2.3401	10.0566	23.5334
12	1.8620	.3179	.6952	24.5717	.7645	2.4598	9.8661	24.2687
13	1.9063	.3221	.7000	24.7105	.7706	2.5875	9.6661	25.0112
14	1.9506	.3264	.7049	24.8474	.7769	2.7238	9.4583	25.7627
15	1.9948	.3306	.7100	24.9817	.7833	2.8692	9.2445	26.5241
16	2.0391	.3352	.7151	25.1125	.7898	3.0244	9.0263	27.2995
17	2.0834	.3395	.7203	25.2397	.7963	3.1904	8.8059	28.0944
18	2.1276	.3438	.7254	25.3609	.8023	3.3678	8.5853	28.9136
19	2.1719	.3481	.7304	25.4693	.8081	3.5555	8.3663	29.7463

Figure A.2(e) - continued

ADDITIONAL AXIAL LOCATION NO.45, AT X/D = -4.6689

I	R/D	VR/VINF	VX/VINF	FLOW ANGLE	V/VINF	RHO/RHOINF	T/TINF	P/PINF
1	1.7691	.0073	.9043	.4607	.9044	.4326	4.8848	2.1132
2	1.9276	.0135	.9075	.8494	.9077	.4504	4.7561	2.1421
3	2.0649	.0201	.9112	1.2599	.9114	.4728	4.6136	2.1813
4	2.2128	.0272	.9144	1.7062	.9148	.4989	4.4819	2.2358
5	2.3676	.0356	.9175	2.2076	.9182	.5317	4.3473	2.3116
6	2.5035	.0447	.9203	2.7800	.9214	.5726	4.2267	2.4167
7	2.6554	.0553	.9228	3.4322	.9244	.6236	4.1024	2.5584
8	2.8043	.0672	.9248	4.1552	.9272	.6871	3.9912	2.7422
9	2.9522	.0800	.9265	4.9336	.9299	.7652	3.8848	2.9725
10	3.1006	.0934	.9279	5.7464	.9325	.8603	3.7812	3.2531
11	3.2479	.1071	.9289	6.5772	.9351	.9753	3.6795	3.5887
12	3.3958	.1209	.9297	7.4115	.9376	1.1128	3.5812	3.9859
13	3.5437	.1347	.9302	8.2382	.9399	1.2753	3.4885	4.4490
14	3.6916	.1482	.9312	9.0545	.9419	1.4658	3.4052	4.9915
15	3.8394	.1614	.9299	9.8443	.9438	1.6855	3.3316	5.6155
16	3.9873	.1743	.9296	10.6241	.9452	1.9374	3.2740	6.3429
17	4.1352	.1862	.9279	11.3488	.9464	2.2159	3.2265	7.1496
18	4.2831	.1965	.9258	12.1035	.9468	2.5309	3.2082	8.1195
19	4.4314	.2089	.9240	12.7372	.9473	2.8557	3.1897	9.1089

ADDITIONAL AXIAL LOCATION NO.46, AT X/D = -4.9175

I	R/D	VR/VINF	VX/VINF	FLOW ANGLE	V/VINF	RHO/RHOINF	T/TINF	P/PINF
1	1.7716	.0083	.9055	.4616	.9055	.4269	4.8417	2.0869
2	1.9237	.0124	.9089	.7818	.9090	.4444	4.7070	2.0917
3	2.0765	.0168	.9126	1.1776	.9128	.4663	4.5583	2.1256
4	2.2293	.0220	.9164	1.6232	.9154	.4918	4.4183	2.1731
5	2.3820	.0334	.9194	2.0776	.9200	.5238	4.2764	2.2398
6	2.5348	.0422	.9225	2.5194	.9234	.5635	4.1412	2.3334
7	2.6875	.0524	.9252	3.0396	.9256	.6131	4.0151	2.4618
8	2.8403	.0638	.9274	3.9352	.9296	.6752	3.8974	2.6314
9	2.9931	.0763	.9293	4.6915	.9324	.7519	3.7863	2.8468
10	3.1456	.0894	.9309	5.4892	.9351	.8458	3.6798	3.1124
11	3.2985	.1030	.9320	6.3093	.9377	.9596	3.5770	3.4325
12	3.4512	.1168	.9328	7.1369	.9401	1.0960	3.4790	3.8131
13	3.6040	.1305	.9333	7.9591	.9424	1.2576	3.3879	4.2606
14	3.7568	.1441	.9333	8.7735	.9444	1.4472	3.3073	4.7862
15	3.9095	.1572	.9330	9.5633	.9461	1.6659	3.2371	5.3925
16	4.0623	.1701	.9320	10.3447	.9474	1.9168	3.1836	6.1023
17	4.2151	.1821	.9309	11.0702	.9485	2.1937	3.1406	6.8896
18	4.3678	.1945	.9287	11.9280	.9488	2.5068	3.1272	7.8393
19	4.5206	.2049	.9268	12.4641	.9492	2.8289	3.1137	8.8081

ADDITIONAL AXIAL LOCATION NO.59, AT X/D = -9.8284

I	R/D	VR/VINF	VX/VINF	FLOW ANGLE	V/VINF	RHO/RHOINF	T/TINF	P/PINF
1	1.7848	.0109	.9196	.3577	.9196	.3564	4.2931	1.5302
2	2.0220	.0063	.9245	.3892	.9245	.3738	4.0994	1.5322
3	2.2591	.0111	.9296	.5816	.9297	.3953	3.8951	1.5398
4	2.4963	.0157	.9348	.9635	.9349	.4208	3.6854	1.5508
5	2.7335	.0203	.9402	1.2867	.9404	.4519	3.4675	1.5671
6	2.9706	.0250	.9454	1.5168	.9458	.4890	3.2515	1.5902
7	3.2078	.0300	.9505	1.3683	.9509	.5334	3.0425	1.6229
8	3.4450	.0355	.9551	2.1318	.9557	.5867	2.8474	1.6705
9	3.6821	.0421	.9590	2.3163	.9600	.6516	2.6742	1.7426
10	3.9193	.0502	.9622	2.9890	.9635	.7321	2.5288	1.8514
11	4.1565	.0596	.9645	3.5531	.9663	.8318	2.4129	2.0069
12	4.3936	.0707	.9659	4.1939	.9685	.9536	2.3246	2.2166
13	4.6308	.0823	.9665	4.8797	.9701	1.0988	2.2598	2.4831
14	4.8680	.0945	.9664	5.6002	.9714	1.2693	2.2180	2.8154
15	5.1051	.1064	.9658	6.3153	.9716	1.4622	2.1942	3.2084
16	5.3423	.1188	.9648	7.0594	.9716	1.6887	2.1949	3.6890
17	5.5795	.1302	.9625	7.7513	.9713	1.9134	2.2068	4.2225
18	5.8166	.1424	.9598	8.4982	.9703	2.1720	2.2468	4.8800
19	6.0538	.1527	.9573	9.1393	.9694	2.4304	2.2863	5.5565

ADDITIONAL AXIAL LOCATION NO.60, AT X/D = -10.3603

I	R/D	VR/VINF	VX/VINF	FLOW ANGLE	V/VINF	RHO/RHOINF	T/TINF	P/PINF
1	1.7854	.0088	.9209	.0476	.9209	.3501	4.2424	1.4854
2	2.0329	.0141	.9259	.3753	.9259	.3676	4.0439	1.4866
3	2.2803	.0197	.9311	.6598	.9312	.3894	3.8350	1.4935
4	2.5278	.0252	.9365	.9330	.9366	.4154	3.6203	1.5039
5	2.7752	.0307	.9419	1.1985	.9421	.4472	3.3972	1.5191
6	3.0227	.0362	.9473	1.4646	.9476	.4851	3.1763	1.5407
7	3.2701	.0419	.9524	1.7406	.9529	.5302	2.9628	1.5710
8	3.5175	.0342	.9572	2.0415	.9578	.5841	2.7637	1.6143
9	3.7650	.0402	.9618	2.3948	.9621	.6491	2.5869	1.6793
10	4.0124	.0477	.9645	2.8318	.9647	.7289	2.4391	1.7777
11	4.2599	.0568	.9668	3.3631	.9635	.8273	2.3227	1.9215
12	4.5073	.0672	.9683	3.9775	.9716	.9475	2.2360	2.1186
13	4.7548	.0785	.9689	4.6429	.9721	1.0908	2.1742	2.3716
14	5.0022	.0914	.9688	5.3477	.9730	1.2587	2.1365	2.6891
15	5.2496	.1022	.9681	6.0493	.9735	1.4478	2.1171	3.0653
16	5.4971	.1144	.9666	6.7819	.9733	1.6615	2.1223	3.5262
17	5.7445	.1257	.9648	7.4849	.9736	1.8881	2.1385	4.0377
18	5.9920	.1377	.9621	8.2043	.9719	2.1395	2.1818	4.6679
19	6.2394	.1486	.9595	8.9371	.9709	2.3901	2.2247	5.3173

Figure A.2(f) - continued

STREAMLINE TRAJECTORY CALCULATION
****

43 STREAMLINES CALCULATED

STREAMLINE NO. 1, STARTING AT X/D = 1.2703, R/D = .0444
(CORRESPONDS TO THETA = 2.00 DEGREES, RP/D = 1.2711)

X/D	R/D
1.2703	.0444
1.2403	.0465
1.2153	.0483
1.1903	.0515
1.1653	.0552
1.1403	.0600
1.1153	.0674
1.0903	.0780
1.0653	.0988
1.0403	.1233
1.0153	.1889
.9903	.2680
.9653	.3498
.9353	.4353
.9103	.4954
.8853	.5509
.8603	.6002
.8353	.6454
.8103	.6859
.7853	.7257
.7603	.762.
.7353	.7961
.7103	.8284
.6853	.8539
.6603	.8879
.6353	.9209
.6053	.9471
.5803	.9721
.5553	.9961
.5303	1.0191
.5053	1.0412
.4803	1.0624
.4553	1.0829
.4303	1.1026
.4053	1.1216
.3803	1.1400
.3553	1.1577
.3253	1.1782
.3003	1.1947
.2753	1.2106
.2503	1.2263
.2253	1.2409
.2003	1.2554
.1753	1.2694
.1503	1.2830
.1253	1.2962
.1003	1.3090
.0753	1.3214
.0503	1.3334
.0253	1.3451
-.0047	1.3580
-.0161	1.4420
-.0276	1.5077
-.0684	1.5539
-.0202	1.5959
-1.0336	1.6505
-1.2129	1.6543
-1.4243	1.6773
-1.6455	1.6936
-1.8170	1.7093
-2.224	1.7223
-2.2096	1.7315
-2.4210	1.7407
-2.6325	1.7484
-2.8137	1.754.
-3.0251	1.7597
-3.2064	1.7639
-3.4178	1.7681
-3.6292	1.7718
-3.8104	1.7746
-4.0219	1.7774
-4.2333	1.7799
-4.4145	1.7819
-4.6259	1.7839
-4.8072	1.7854
-4.9175	1.7863

Figure A.2(g) - continued

STREAMLINE NO.42, STARTING AT X/D = -4.3769, R/D = 4.3233

X/D	R/D
-4.3769	4.3233
-4.5683	4.3717
-4.7998	4.4187
-4.9175	4.4442

STREAMLINE NO.43, STARTING AT X/D = -4.6915, R/D = 4.4391

X/D	R/D
-4.6915	4.4391
-4.9029	4.4861
-4.9175	4.4893

MAGNETIC FIELD COMPONENTS

ANGULAR LOCATION NO. 1, AT THETA = 0.0000 DEGREES

I	RP/D	B/BINF (PARALLEL)	B/BINF (PERP)	B/BINF (IN-PLANE)	B-ANGLE (IN-PLANE)	B/BINF (NORMAL)	BX/BINF (RESULTANT)	BY/BINF (RESULTANT)	BZ/BINF (RESULTANT)
1	1.0000	0.0000							
2	1.0150	.0693	19.5068	13.7949	-89.7964	4.2200	.0347	-9.7544	2.9840
3	1.0302	.2292	15.0305	10.6321	-89.5076	4.2156	.0646	-7.5177	2.9809
4	1.0451	.1898	11.7607	8.2804	-89.0712	4.2089	.0949	-5.8544	2.9761
5	1.0602	.2509	9.8738	6.9841	-88.5445	4.1998	.1254	-4.9369	2.9697
6	1.0752	.3121	8.5656	6.0615	-87.9137	4.1885	.1560	-4.2833	2.9616
7	1.0903	.3731	7.5905	5.3667	-87.1821	4.1744	.1868	-3.7903	2.9518
8	1.1053	.4337	6.8484	4.8521	-86.3759	4.1582	.2169	-3.4241	2.9403
9	1.1204	.4956	6.2993	4.4551	-85.5100	4.1395	.2468	-3.1476	2.9271
10	1.1354	.5529	5.8912	4.1849	-84.6424	4.1185	.2762	-2.9456	2.9122
11	1.1505	.6101	5.5159	3.9241	-83.6888	4.0951	.3059	-2.7589	2.8957
12	1.1655	.6660	5.2139	3.7167	-82.7232	4.0693	.3330	-2.6369	2.8775
13	1.1805	.7212	4.9402	3.5301	-81.7056	4.0412	.3601	-2.4731	2.8575
14	1.1956	.7722	4.7264	3.3879	-80.7244	4.0106	.3861	-2.3642	2.8360
15	1.2106	.8219	4.5212	3.2454	-79.6968	3.9777	.4110	-2.2606	2.8127
16	1.2257	.8690	4.3294	3.1101	-78.6277	3.9424	.4345	-2.1602	2.7877
17	1.2407	.9132	4.1699	3.0185	-77.6478	3.9248	.4566	-2.0850	2.7611
18	1.2558	.9543	4.0179	2.9202	-76.6392	3.8647	.4772	-2.0493	2.7328
19	1.2708	1.0000	3.8829	2.7928	-75.3336	3.8209	.5000	-1.9104	2.7018

ANGULAR LOCATION NO. 2, AT THETA = 2.0000 DEGREES

I	RP/D	B/BINF (PARALLEL)	B/BINF (PERP)	B/BINF (IN-PLANE)	B-ANGLE (IN-PLANE)	B/BINF (NORMAL)	BX/BINF (RESULTANT)	BY/BINF (RESULTANT)	BZ/BINF (RESULTANT)
1	1.0000	.0437							
2	1.0152	.1038	19.6130	13.8129	-91.2200	6.8133	-.2080	-9.7650	4.8177
3	1.0302	.1626	14.9733	10.5231	-90.7384	6.6524	-.0959	-7.4417	4.7039
4	1.0453	.2216	11.6961	8.2066	-90.0175	6.4249	-.0618	-5.8031	4.5431
5	1.0603	.2812	9.8647	6.9237	-89.2630	6.2297	.0630	-4.8954	4.4051
6	1.0754	.3363	8.5682	6.0071	-88.4184	6.0036	.1172	-4.2460	4.2492
7	1.0904	.3958	7.5660	5.3180	-87.5814	5.7295	.1587	-3.7571	4.0514
8	1.1055	.4525	6.8536	4.8053	-86.6891	5.5336	.1962	-3.3922	3.9129
9	1.1205	.5084	6.3059	4.4195	-85.7462	5.3252	.2318	-3.1167	3.7655
10	1.1356	.5625	5.8963	4.1350	-84.8951	5.1446	.2602	-2.9123	3.6378
11	1.1507	.6175	5.5201	3.8743	-84.0020	4.9735	.2863	-2.7246	3.5168
12	1.1657	.6714	5.2160	3.6645	-83.1898	4.8030	.3073	-2.5729	3.3962
13	1.1808	.7221	4.9469	3.4755	-82.3432	4.6550	.3276	-2.4357	3.2916
14	1.1958	.7725	4.7240	3.3289	-81.4203	4.5030	.3512	-2.3275	3.1841
15	1.2109	.8215	4.5131	3.1872	-80.4476	4.3606	.3740	-2.2225	3.0830
16	1.2259	.8690	4.3112	3.0529	-79.4017	4.2217	.3971	-2.1219	2.9852
17	1.2410	.9148	4.1274	2.9534	-78.3632	4.0825	.4212	-2.0454	2.8868
18	1.2561	.9589	4.0049	2.8555	-77.2883	3.9556	.4446	-1.9696	2.7971
19	1.2711	1.0011	3.8298	2.7368	-75.0275	3.8218	.4673	-1.8780	2.7324

Figure A.2(h) - continued

ANGULAR LOCATION NO. 3, AT THETA = 6.0000 DEGREES

I	RP/D	B/BINF (PARALLEL)	B/BINF (PERP)	B/BINF (IN-PLANE)	B-ANGLE (IN-PLANE)	B/BINF (NORMAL)	BX/BINF (RESULTANT)	BY/BINF (RESULTANT)	BZ/BINF (RESULTANT)
1	1.0011	.1723							
2	1.0163	.2211	19.7354	13.8038	-94.0993	12.0951	-.6977	-9.7358	8.5525
3	1.0314	.2700	14.8020	10.2979	-93.4057	11.6519	-.4326	-7.2588	8.2391
4	1.0465	.3188	11.6097	8.0349	-92.1443	10.9525	-.2126	-5.6776	7.7446
5	1.0616	.3676	9.8224	6.7739	-91.0505	10.3597	-.0878	-4.7891	7.3254
6	1.0767	.4164	8.5379	5.8698	-89.9227	8.6352	.0056	-4.1506	6.1060
7	1.0918	.4651	7.5758	5.1945	-88.9634	7.5021	.0664	-3.6725	5.3048
8	1.1069	.5138	6.8514	4.6877	-87.9932	6.8321	.1161	-3.3127	4.8310
9	1.1221	.5624	6.3034	4.3063	-87.0325	6.3063	.1576	-3.0459	4.4592
10	1.1372	.6108	5.8859	4.0167	-86.1842	5.9080	.1890	-2.8339	4.1776
11	1.1523	.6592	5.5098	3.7568	-35.2620	5.5322	.2194	-2.6474	3.9119
12	1.1674	.7073	5.2036	3.5460	-84.4222	5.1748	.2437	-2.4955	3.6591
13	1.1825	.7551	4.9275	3.3572	-83.4894	4.9048	.2692	-2.3586	3.4682
14	1.1976	.8027	4.7026	3.2041	-82.6489	4.6555	.2899	-2.2470	3.2919
15	1.2128	.8499	4.4867	3.0578	-81.7628	4.4481	.3098	-2.1399	3.1453
16	1.2279	.8966	4.2843	2.9220	-80.7752	4.2662	.3312	-2.0395	3.0166
17	1.2430	.9429	4.1259	2.8199	-79.6469	4.0980	.3583	-1.9615	2.8977
18	1.2581	.9885	3.9738	2.7231	-78.4669	3.9595	.3850	-1.8866	2.7998
19	1.2732	1.0334	3.8123	2.6216	-77.1725	3.8206	.4116	-1.8075	2.7016

ANGULAR LOCATION NO.23, AT THETA = 86.0000 DEGREES

I	RP/D	B/BINF (PARALLEL)	B/BINF (PERP)	B/BINF (IN-PLANE)	B-ANGLE (IN-PLANE)	B/BINF (NORMAL)	BX/BINF (RESULTANT)	BY/BINF (RESULTANT)	BZ/BINF (RESULTANT)
1	1.3092	1.1539							
2	1.3484	1.2145	8.5172	5.1726	-149.3591	11.8723	-3.1469	-1.8541	8.3950
3	1.3875	1.2577	5.7915	3.2113	-146.2984	7.1832	-1.8891	-1.2999	5.0793
4	1.4266	1.3240	4.5546	2.3029	-142.7691	5.6294	-1.2965	-.9952	3.9806
5	1.4658	1.3738	3.8240	1.7525	-138.9333	4.8498	-.9343	-.8141	3.4293
6	1.5049	1.4376	3.3967	1.4204	-133.7244	4.3286	-.6942	-.7258	3.0608
7	1.5441	1.5057	3.1093	1.1903	-127.5731	4.0145	-.5132	-.6671	2.8387
8	1.5832	1.5787	2.9117	1.0300	-120.5963	3.7970	-.3707	-.6269	2.6849
9	1.6223	1.6569	2.7665	.9114	-112.9630	3.6526	-.2513	-.5931	2.5828
10	1.6615	1.7407	2.6584	.8274	-104.3051	3.5498	-.1446	-.5669	2.5101
11	1.7006	1.8307	2.5789	.7682	-95.5531	3.4850	-.0526	-.5497	2.4643
12	1.7397	1.9272	2.5202	.7392	-85.8703	3.4386	.0376	-.5213	2.4314
13	1.7789	2.0307	2.4826	.7373	-76.2889	3.4192	.1236	-.5165	2.4177
14	1.8180	2.1415	2.4593	.7494	-67.5293	3.4158	.2025	-.4897	2.4154
15	1.8571	2.2601	2.4524	.7576	-59.3933	3.4262	.2836	-.4793	2.4227
16	1.8963	2.3869	2.4456	.8376	-51.7332	3.4525	.3668	-.4650	2.4413
17	1.9354	2.5223	2.4446	.8986	-44.9234	3.4900	.4499	-.4487	2.4678
18	1.9745	2.6666	2.4489	.9694	-38.9013	3.5389	.5335	-.4305	2.5024
19	2.0137	2.8201	2.4452	1.0402	-33.6877	3.5964	.6162	-.4915	2.5430

ANGULAR LOCATION NO.24, AT THETA = 90.0000 DEGREES

I	RP/D	B/BINF (PARALLEL)	B/BINF (PERP)	B/BINF (IN-PLANE)	B-ANGLE (IN-PLANE)	B/BINF (NORMAL)	BX/BINF (RESULTANT)	BY/BINF (RESULTANT)	BZ/BINF (RESULTANT)
1	1.3491	1.0925							
2	1.3922	1.1443	7.7661	4.6839	-151.7632	10.8083	-2.9163	-1.5700	7.6424
3	1.4352	1.1989	5.3267	2.9243	-148.1538	6.5661	-1.7565	-1.0912	4.6430
4	1.4783	1.2567	4.1933	2.0885	-144.2240	5.1963	-1.1981	-.8634	3.6744
5	1.5214	1.3184	3.5221	1.5809	-139.7697	4.4745	-.8538	-.7216	3.1639
6	1.5645	1.3844	3.1343	1.2765	-133.7248	4.0266	-.6239	-.6523	2.8472
7	1.6076	1.4552	2.8746	1.0666	-127.0528	3.7522	-.4544	-.6019	2.6532
8	1.6506	1.5314	2.7001	.9232	-119.2654	3.5687	-.3191	-.5695	2.5234
9	1.6937	1.6133	2.5728	.8176	-110.9357	3.4484	-.2066	-.5493	2.4384
10	1.7368	1.7016	2.4808	.7494	-101.2582	3.3767	-.1034	-.5197	2.3834
11	1.7799	1.7968	2.4151	.7039	-91.6648	3.3186	-.0145	-.4975	2.3466
12	1.8230	1.8992	2.3696	.6984	-80.6905	3.2954	.0799	-.4873	2.3302
13	1.8660	2.0095	2.3428	.6981	-71.5558	3.2924	.1562	-.4682	2.3281
14	1.9091	2.1281	2.3295	.7355	-62.1836	3.3035	.2427	-.4600	2.3359
15	1.9522	2.2554	2.3311	.7869	-54.2423	3.3327	.3251	-.4515	2.3565
16	1.9953	2.3920	2.3315	.8436	-46.9440	3.3742	.4073	-.4359	2.3859
17	2.0384	2.5383	2.3372	.9186	-40.4107	3.4282	.4946	-.4211	2.4241
18	2.0814	2.6946	2.3469	.9964	-34.6920	3.4919	.5793	-.4010	2.4691
19	2.1245	2.8613	2.3530	1.0732	-29.3214	3.5676	.6617	-.3716	2.5226

Figure A.2(i) - continued

APPENDIX A

ADDITIONAL AXIAL LOCATION NO. 1, AT X/D = -.0302

I	R/D	B/BINF (PARALLEL)	B/BINF (PERP)	B/BINF (IN-PLANE)	B-ANGLE (IN-PLANE)	B/BINF (NORMAL)	BX/BINF (RESULTANT)	BY/BINF (RESULTANT)	BZ/BINF (RESULTANT)
1	1.3623	1.0746							
2	1.4060	1.1198	7.5038	4.5158	-152.3782	10.2705	-2.8292	-1.4805	7.2623
3	1.4497	1.1809	5.1878	2.8388	-148.7239	6.3551	-1.7156	-1.0421	4.4938
4	1.4933	1.2389	4.0864	2.0260	-144.5321	5.0624	-1.1667	-.8312	3.5797
5	1.5579	1.3017	3.4377	1.5325	-140.0458	4.3698	-.8307	-.6959	3.0899
6	1.5807	1.3689	3.0636	1.2386	-133.7180	3.9452	-.6053	-.6330	2.7897
7	1.6243	1.4407	2.8129	1.0351	-126.8436	3.6841	-.4389	-.5857	2.6051
8	1.6680	1.5177	2.6449	.8961	-119.9136	3.5105	-.3064	-.5547	2.4823
9	1.7117	1.6006	2.5229	.7961	-113.1683	3.3967	-.1941	-.5284	2.4018
10	1.7553	1.6898	2.4355	.7314	-133.3643	3.3254	-.0930	-.5087	2.3514
11	1.7990	1.7861	2.3740	.6878	-90.7821	3.2776	-.0066	-.4863	2.3178
12	1.8427	1.8899	2.3316	.6889	-79.5218	3.2598	.0888	-.4790	2.3051
13	1.8863	2.0017	2.3090	.5939	-73.2615	3.2610	.1657	-.4618	2.3059
14	1.9300	2.1220	2.2988	.7336	-61.0233	3.2769	.2513	-.4538	2.3172
15	1.9737	2.2513	2.3030	.7886	-53.1616	3.3101	.3343	-.4463	2.3406
16	2.0173	2.3902	2.3157	.8459	-45.9926	3.3557	.4156	-.4302	2.3729
17	2.0610	2.5394	2.3138	.9237	-39.5445	3.4136	.5036	-.4158	2.4138
18	2.1047	2.6995	2.3259	1.0036	-33.8930	3.4824	.5891	-.3957	2.4625
19	2.1483	2.8696	2.3347	1.0809	-28.6030	3.5617	.6710	-.3659	2.5185

ADDITIONAL AXIAL LOCATION NO. 2, AT X/D = -.0634

I	R/D	B/BINF (PARALLEL)	B/BINF (PERP)	B/BINF (IN-PLANE)	B-ANGLE (IN-PLANE)	B/BINF (NORMAL)	BX/BINF (RESULTANT)	BY/BINF (RESULTANT)	BZ/BINF (RESULTANT)
1	1.3751	1.0526							
2	1.4194	1.1000	7.2744	4.3676	-153.3331	9.8632	-2.7526	-1.4395	6.9743
3	1.4636	1.1635	5.0606	2.7608	-149.3021	6.1736	-1.6786	-.9966	4.3654
4	1.5079	1.2222	3.9900	1.9699	-144.8720	4.9481	-1.1392	-.8915	3.4988
5	1.5522	1.2848	3.3557	1.4868	-140.3229	4.2715	-.8092	-.8713	3.0234
6	1.5964	1.3529	2.9944	1.2221	-133.6909	3.8695	-.5872	-.6146	2.7361
7	1.6407	1.4256	2.7529	1.0059	-126.6392	3.6204	-.4241	-.5702	2.5600
8	1.6850	1.5035	2.5917	.8707	-119.5724	3.4564	-.2945	-.5407	2.4440
9	1.7292	1.5874	2.4751	.7765	-109.3692	3.3485	-.1821	-.5180	2.3678
10	1.7735	1.6777	2.3925	.7151	-99.4893	3.2837	-.0834	-.4987	2.3219
11	1.8178	1.7752	2.3350	.6785	-89.4659	3.2405	.0045	-.4798	2.2913
12	1.8620	1.8804	2.2963	.6806	-78.3675	3.2273	.0976	-.4714	2.2821
13	1.9063	1.9939	2.2769	.6918	-68.9846	3.2320	.1754	-.4566	2.2834
14	1.9506	2.1160	2.2694	.7324	-59.9162	3.2526	.2596	-.4481	2.3000
15	1.9948	2.2474	2.2761	.7910	-52.1266	3.2894	.3434	-.4415	2.3259
16	2.0391	2.3887	2.2869	.8514	-45.0421	3.3390	.4254	-.4260	2.3610
17	2.0834	2.5405	2.2911	.9297	-38.7107	3.4064	.5129	-.4111	2.4045
18	2.1276	2.7036	2.3051	1.0106	-33.1342	3.4733	.5984	-.3906	2.4560
19	2.1719	2.8769	2.3153	1.0886	-27.9289	3.5555	.6801	-.3605	2.5141

ADDITIONAL AXIAL LOCATION NO.45, AT X/D = -4.0089

I	R/D	B/BINF (PARALLEL)	B/BINF (PERP)	B/BINF (IN-PLANE)	B-ANGLE (IN-PLANE)	B/BINF (NORMAL)	BX/BINF (RESULTANT)	BY/BINF (RESULTANT)	BZ/BINF (RESULTANT)
1	1.7691	.3952							
2	1.9170	.4088	2.4912	1.4725	-178.8044	2.6546	-1.0410	-.0217	1.8771
3	2.0649	.4309	1.2732	.5957	-176.8662	1.7587	-.4206	-.0230	1.2436
4	2.2128	.4564	.9545	.3767	-147.7656	1.4419	-.2253	-.1421	1.0196
5	2.3606	.4882	.8351	.3236	-121.7165	1.3097	-.1203	-.1947	.9261
6	2.5085	.5276	.7866	.3235	-103.3526	1.2566	-.0528	-.2226	.8885
7	2.6564	.5765	.7729	.3391	-93.6363	1.2483	-.0002	-.2398	.8827
8	2.8043	.6371	.7937	.3869	-79.6481	1.2757	.0492	-.2691	.9021
9	2.9522	.7115	.8152	.4227	-71.5340	1.3309	.0947	-.2835	.9411
10	3.1000	.8023	.8664	.4882	-64.7550	1.4128	.1472	-.3123	.9998
11	3.2479	.9126	.9458	.5396	-58.1646	1.5189	.2010	-.3238	1.0740
12	3.3958	1.0432	.9818	.5979	-53.8015	1.6429	.2641	-.3238	1.1617
13	3.5437	1.1987	.9712	.6677	-44.6333	1.7834	.3394	-.3281	1.2610
14	3.6916	1.3807	1.0212	.7664	-38.0186	1.9401	.4270	-.3338	1.3718
15	3.8394	1.5907	1.0736	.8769	-32.3673	2.1101	.5237	-.3319	1.4921
16	3.9873	1.8312	1.1344	1.0192	-27.6554	2.2924	.6383	-.3345	1.6210
17	4.1352	2.0971	1.1811	1.1505	-22.7621	2.4780	.7502	-.3148	1.7522
18	4.2831	2.3963	1.2699	1.3244	-19.5913	2.6749	.8823	-.3140	1.8914
19	4.4310	2.7252	1.3554	1.5093	-16.9048	2.8557	1.0211	-.3103	2.0193

ADDITIONAL AXIAL LOCATION NO.46, AT X/D = -4.9175

I	R/D	B/BINF (PARALLEL)	B/BINF (PERP)	B/BINF (IN-PLANE)	B-ANGLE (IN-PLANE)	B/BINF (NORMAL)	BX/BINF (RESULTANT)	BY/BINF (RESULTANT)	BZ/BINF (RESULTANT)
1	1.7710	.3865							
2	1.9237	.4039	2.5379	1.5090	179.8520	2.5828	-1.0670	.0028	1.8263
3	2.0765	.4256	1.2652	.5936	-178.9165	1.7169	-.4196	-.0079	1.2140
4	2.2293	.4507	.9602	.3883	-149.3779	1.4095	-.2363	-.1398	.9967
5	2.3820	.4819	.8578	.3469	-123.5857	1.2818	-.1333	-.2008	.9064
6	2.5348	.5203	.8060	.3377	-105.4436	1.2307	-.0636	-.2302	.8702
7	2.6875	.5682	.7936	.3545	-92.3712	1.2231	-.0104	-.2505	.8649
8	2.8403	.6276	.8141	.4012	-81.7924	1.2507	.0405	-.2808	.8844
9	2.9930	.7031	.8345	.4358	-73.6228	1.3066	.0869	-.2957	.9239
10	3.1458	.7979	.8849	.4993	-66.7349	1.3890	.1394	-.3243	.9822
11	3.2985	.8998	.9212	.5478	-59.9594	1.4962	.1939	-.3353	1.0580
12	3.4513	1.0134	.9681	.5980	-52.4654	1.6196	.2576	-.3353	1.1452
13	3.6040	1.1451	.9923	.6740	-46.0303	1.7603	.3309	-.3430	1.2447
14	3.7568	1.3007	1.0433	.7721	-39.8629	1.9170	.4191	-.3500	1.3555
15	3.9095	1.5761	1.0859	.8805	-33.7636	2.0868	.5176	-.3460	1.4756
16	4.0623	1.8160	1.1502	1.0193	-28.9304	2.2694	.6308	-.3487	1.6047
17	4.2151	2.0867	1.1860	1.1500	-23.5757	2.4542	.7453	-.3252	1.7354
18	4.3678	2.3765	1.2617	1.3237	-20.5710	2.6500	.8763	-.3269	1.8738
19	4.5206	2.6851	1.3390	1.5138	-17.3815	2.8289	1.0211	-.3138	2.0093

Figure A.2(j) - continued

VELOCITY AND TEMPERATURE CONTOURS

12 VELOCITY (TEMPERATURE) CONTOUR LINES FOUND

11 POINTS IN CONTOUR LINE OF V/VINF = .100, T/TINF = 22.120

X/D	R/D
.9853	.1901
1.0019	.1832
1.0109	.1785
1.0186	.1741
1.0353	.1604
1.0527	.1465
1.0707	.1133
1.0712	.1126
1.0886	.0614
1.0933	.0383
1.1010	0.0000

20 POINTS IN CONTOUR LINE OF V/VINF = .200, T/TINF = 21.480

X/D	R/D
.9447	.3697
.9614	.3689
.9785	.3662
.9961	.3617
1.0142	.3552
1.0328	.3466
1.0453	.3396
1.0516	.3356
1.0704	.3215
1.0897	.3044
1.1095	.2842
1.1151	.2780
1.1294	.2577
1.1498	.2253
1.1614	.2148
1.1705	.1811
1.1906	.1251
1.1913	.1211
1.2681	.0420
1.2636	0.0000

DENSITY CONTOURS

10 DENSITY CONTOUR LINES FOUND

9 POINTS IN CONTOUR LINE OF RHO/RHOINF = 4.200

X/D	R/D
.9930	.1282
1.0086	.1251
1.0247	.1171
1.0346	.1086
1.0417	.0925
1.0597	.0376
1.0598	.0378
1.0599	.0242
1.0599	0.0000

Figure A.2(k) - continued

29 POINTS IN CONTOUR LINE OF RHO/RHOINF = 4.000

X/D	R/D
.9483	.3572
.9617	.3678
.9754	.3776
.9895	.3861
1.0039	.3933
1.0189	.3986
1.0344	.4016
1.0508	.4016
1.0681	.3977
1.0867	.3886
1.1071	.3722
1.1296	.3626
1.1298	.3406
1.1558	.2882
1.1559	.2881
1.1795	.2185
1.1874	.1641
1.1939	.1255
1.2011	.0419
1.2005	0.0000

MAGNETIC FIELD CONTOURS
****

11 MAGNETIC FIELD CONTOUR LINES FOUND
(FOR COMPONENT ALONG FIELD LINES PARALLEL TO FLOW IN FREESTREAM)

13 POINTS IN CONTOUR LINE OF B/BINF (PARALLEL) = .500

X/D	R/D
.9789	.2283
.9954	.2226
1.0123	.2144
1.0296	.2033
1.0473	.1892
1.0654	.1854
1.0849	.1689
1.0831	.1405
1.0966	.1153
1.1018	.0983
1.1176	.0789
1.1202	.0268
1.1220	0.0000

19 POINTS IN CONTOUR LINE OF B/BINF (PARALLEL) = .750

X/D	R/D
.9513	.3464
.9684	.3437
.9860	.3391
1.0041	.3326
1.0122	.3289
1.0222	.3242
1.0403	.3136
1.0589	.3004
1.0780	.2845
1.0911	.2721
1.0973	.2651
1.1166	.2406
1.1364	.2116
1.1427	.2015
1.1584	.1691
1.1744	.1234
1.1767	.1108
1.1883	.0415
1.1892	0.0000

Figure A.2(1) - continued

12 MAGNETIC FIELD CONTOUR LINES FOUND
(FOR COMPONENT ALONG FIELD LINES PERPENDICULAR TO FLOW IN FREESTREAM)

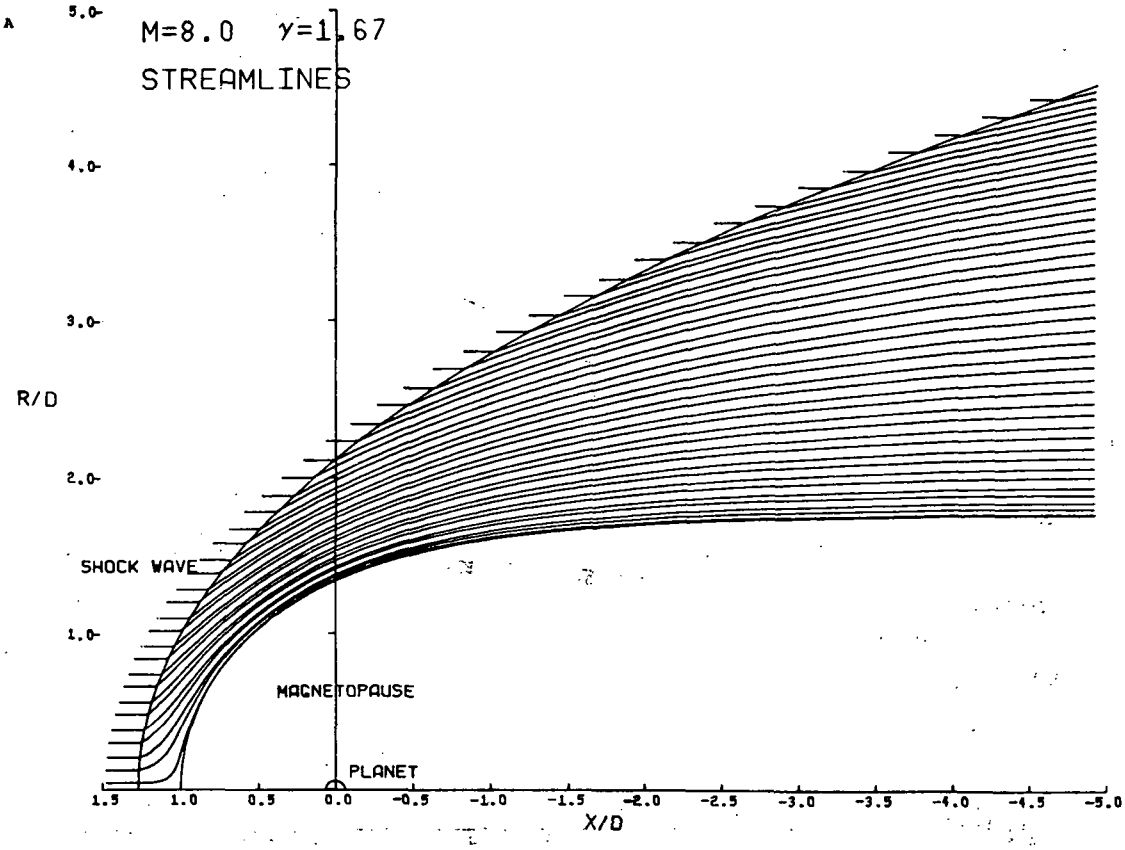
45 POINTS IN CONTOUR LINE OF B/BINF (PERPENDICULAR) = 2.000

X/D	R/D
-1.8975	1.7899
-1.8527	1.7957
-1.7055	1.8142
-1.6156	1.8226
-1.5257	1.8283
-1.5027	1.8293
-1.4358	1.8346
-1.3459	1.8394
-1.2566	1.8414
-1.1661	1.8407
-1.1279	1.8396
-1.0762	1.8425
-.9863	1.8446
-.8964	1.8430
-.8542	1.8409
-.8165	1.8353
-.7538	1.8486
-.6996	1.8499
-.6814	1.8550
-.6461	1.8559
-.5926	1.8625
-.5628	1.8651
-.5392	1.8728
-.4857	1.8877
-.4838	1.8882
-.4322	1.9146
-.4277	1.9165
-.3885	1.9497
-.3786	1.9667
-.3651	1.9890
-.3555	2.0345
-.3628	2.0697
-.3788	2.1363
-.3845	2.1524
-.4257	2.2281
-.4322	2.2404
-.4766	2.3074
-.4857	2.3311
-.5307	2.3978
-.5392	2.4095
-.5926	2.4803
-.6096	2.5017
-.6461	2.5431
-.6996	2.6010
-.7337	2.6367

6 POINTS IN CONTOUR LINE OF B/BINF (PERPENDICULAR) = 2.000

X/D	R/D
-3.6973	1.8865
-3.9232	1.9133
-4.1718	1.9388
-4.4274	1.9599
-4.6669	1.9766
-4.9175	1.9883

Figure A.2(m) - concluded



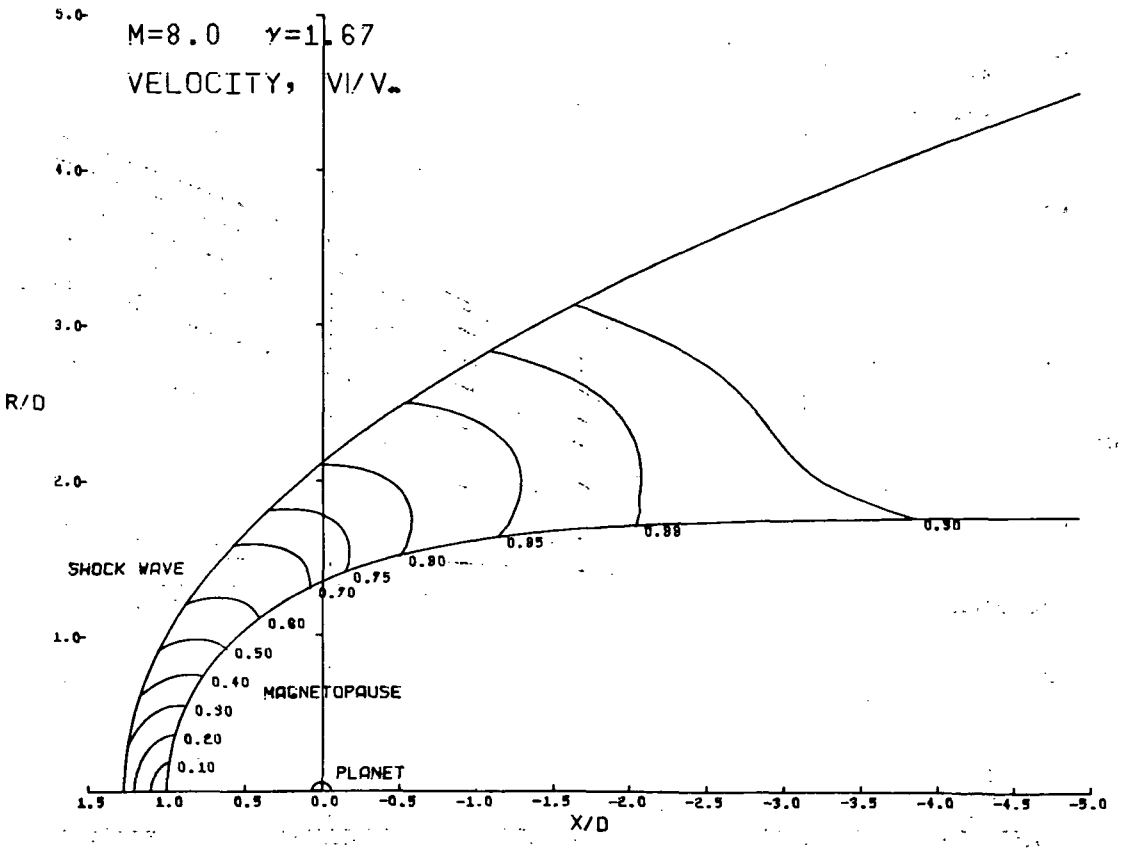
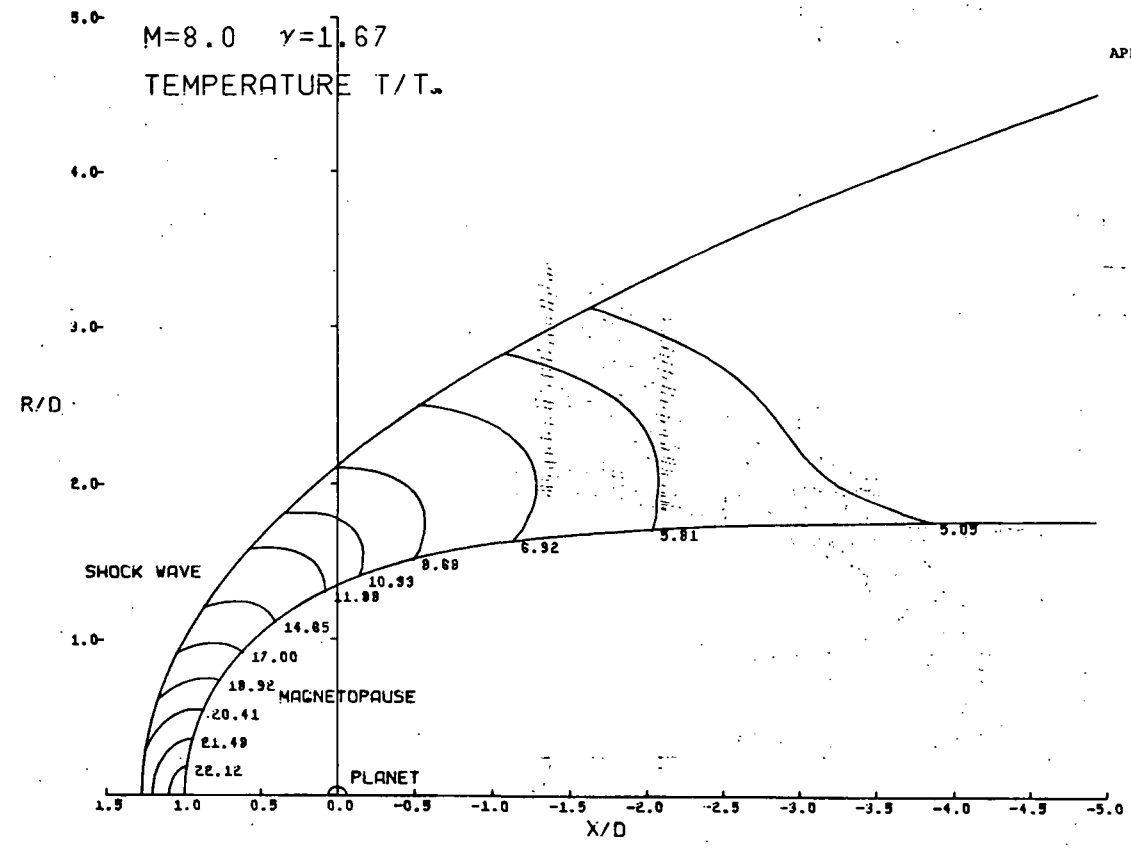


Figure A.3. - Plot output for sample case.



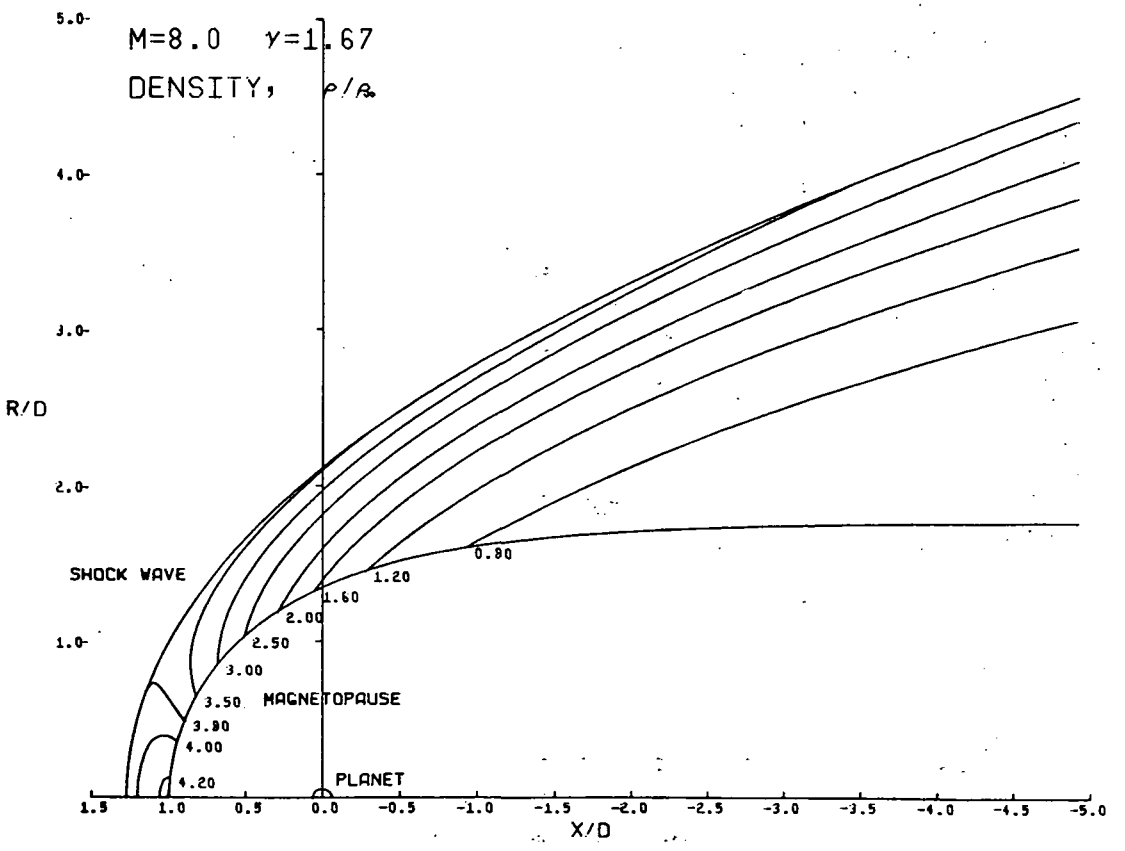
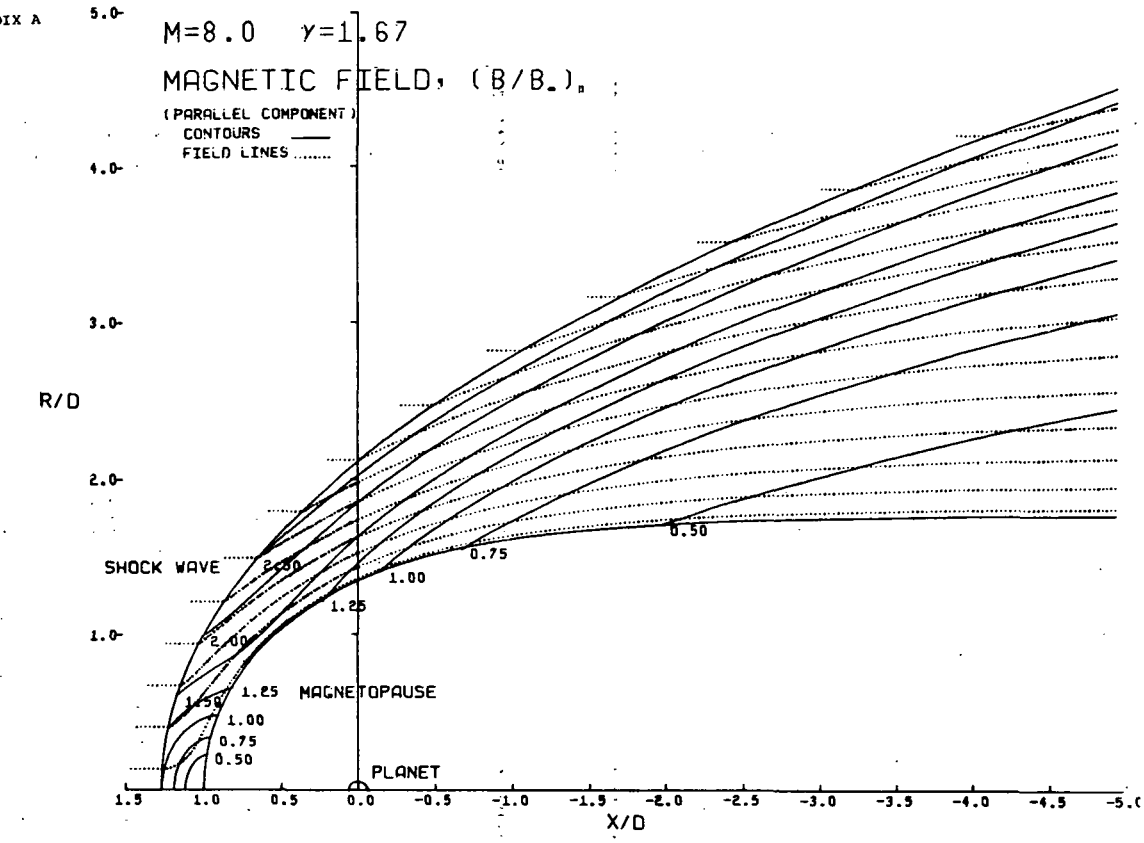


Figure A.3. - Continued.



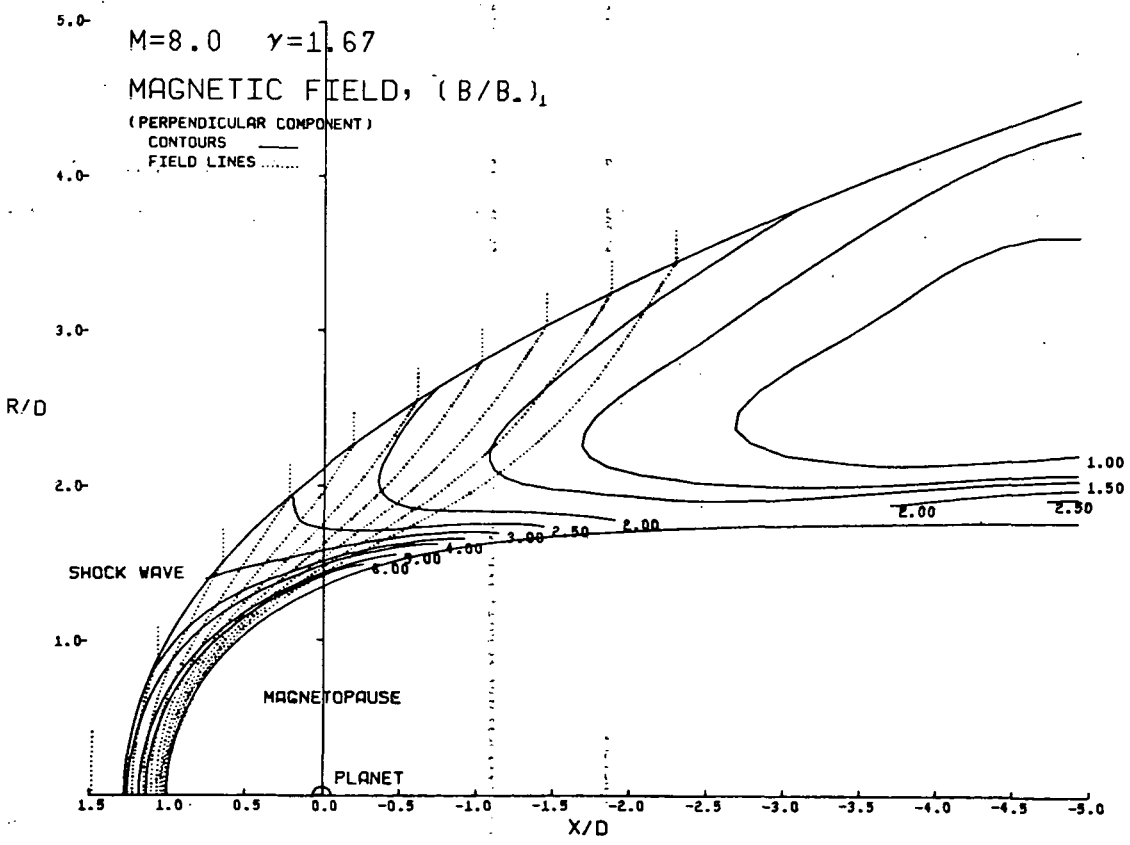


Figure A.3. - Concluded.

LISTING OF COMPUTER PROGRAM

```
      PROGRAM MAIN (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,          MAIN     2
     *              TAPE1,TAPE4,TAPE9)                             MAIN     3
      LOGICAL LRERUN,LPRFL,LPRST,LPRCON,LPRB,LPLOT                  PROPT    2
      COMMON /PROPT/ LRERUN,LPRFL,LPRST,LPRCON,LPRB,LPLOT           PROPT    3
      READ(5,100) NCASE                                             MAIN     5
      DO 20 ICASE=1,NCASE                                           MAIN     6
      CALL INPUT                                                    MAIN     7
      IF (LRERUN) GO TO 10                                          MAIN     8
      CALL BLUNTB                                                   MAIN     9
      CALL FLOWB                                                    MAIN    10
      CALL MARCH                                                    MAIN    11
      CALL FLOWM                                                    MAIN    12
      GO TO 15                                                      MAIN    13
   10 CALL RERUN                                                    MAIN    14
   15 CALL EXTRAP                                                   MAIN    15
      IF (LPRFL) CALL FLOUT                                         MAIN    16
      IF (LPRST) CALL STOUT                                         MAIN    17
      CALL BCOMP                                                    MAIN    16
      IF (LPRB) CALL BOUT                                           MAIN    19
      CALL CONTUR                                                   MAIN    20
   20 CONTINUE                                                      MAIN    21
      STOP                                                          MAIN    22
                                                                    MAIN    23
  100 FORMAT(I1O)                                                   MAIN    24
      END                                                           MAIN    25


      SUBROUTINE BCOMP                                              BCOMP    2
C                                                                   BCOMP    3
C     THIS SUBROUTINE CALCULATES THE COMPONENTS OF THE MAGNETIC FIELD BCOMP    4
C     PARALLEL, PERPENDICULAR AND NORMAL TO THE FLOW.               BCOMP    5
C                                                                   BCOMP    6
      COMMON/CONT/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100),TRANSF1  2
     *  VF(20,100),RHOF(20,100)                                     TRANSF1  3
      COMMON /BLUNT/ THETA(25),RP(25,25)                            TRANSF1  4
      LEVEL 2, VX,VY                                               TRANSF1  5
      COMMON/VCOMP/ VX(25,100),VY(25,100)                           TRANSF1  6
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,NZMAX,BCOMP    8
     1  MACH,GAMMA,RKO,NNINOX                                       BCOMP    9
      LEVEL 2, BPARA,BPERP,BNORM,BMAG,BANG                          BCOMPS   2
      COMMON /BCOMPS/ BPARA(20,100),BPERP(20,100),BNORM(20,100),    BCOMPS   3
     *  BMAG(20,100),BANG(20,100)                                   BCOMPS   4
      LOGICAL LRERUN,LPRFL,LPRST,LPRCON,LPRB,LPLOT                  PROPT    2
      COMMON /PROPT/ LRERUN,LPRFL,LPRST,LPRCON,LPRB,LPLOT           PROPT    3
      COMMON /BIN/ ANGP,ANGN,KBCON,BCON(20)                         BIN      2
      DIMENSION S(100,6),W(100),A(6),YLSQ(100),YLSQI(6)             BCOMP   13
      DATA W/100*1.0/                                               BCOMP   14
C                                                                   BCOMP   15
C     CALCULATE PERPENDICULAR FIELD LINES                           BCOMP   16
C                                                                   BCOMP   17
      IF (KBCON .EQ. 0 .AND. .NOT.LPRB) RETURN                      BCOMP   18
      CALL BSTEP                                                    BCOMP   19
      CALL BELGAM                                                   BCOMP   20
C                                                                   BCOMP   21
C     CALCULATE R/RINF AND EL/ELINF AT EACH GRID POINT, THEN SMOOTH BCOMP   22
C     ALONG CONSTANT-I LINES, USING FIFTH ORDER LEAST SQUARES FIT   BCOMP   23
C                                                                   BCOMP   24
      NRM=NRMAX-1                                                   BCOMP   25
      NZM=NZMAX-1                                                   BCOMP   26
      DO 10 J=1,NZMAX                                               BCOMP   27
      BRAPP(NRMAX,J)=RINF(NRMAX,J)                                  BCOMP   28
      BPERP(NRMAX,J)=ELLINF(NRMAX,J)                                BCOMP   29
      BANG(NRMAX,J)=SLOPEL(NRMAX,J)                                 BCOMP   30
      DO 10 I=2,NRM                                                 BCOMP   31
      BNORM(I,J)=RRINF(I,J)                                         BCOMP   32
      BPERP(I,J)=ELLINF(I,J)                                        BCOMP   33
      BANG(I,J)=SLOPEL(I,J)                                         BCOMP   34
   10 CONTINUE                                                      BCOMP   35
      XLSQ(1)=0.0                                                   BCOMP   36
      DO 60 I=2,NRM                                                 BCOMP   37
      DO 20 J=1,NZM                                                 BCOMP   38
      XLSQ(J+1)=XLSQ(J)+SQRT((YC(I,J+1)-YC(I,J))**2                 BCOMP   39
     *          +(XC(I,J+1)-XC(I,J))**2)                            BCOMP   40
   20 CONTINUE                                                      BCOMP   41
      DO 40 J=1,NZMAX                                               BCOMP   42
      YLSQ(J)=BPERP(I,J)                                            BCOMP   43
   40 CONTINUE                                                      BCOMP   44
      CALL FLSQFY(NZMAX,5,XLSQ,YLSQ,W,100,S,A,IER)                  BCOMP   45
      DO 50 J=1,NZMAX                                               BCOMP   46
      XA=XLSQ(J)                                                    BCOMP   47
      BPERP(I,J)=((((A(6)*XA+A(5))*XA+A(4))*XA+A(3))*XA+A(2))*XA+A(1) BCOMP   48
   50 CONTINUE                                                      BCOMP   49
   60 CONTINUE                                                      BCOMP   50
C                                                                   BCOMP   51
C     CALCULATE COMPONENTS OF MAGNETIC FIELD - PARALLEL, PERPENDICULAR, BCOMP   52
C     AND NORMAL TO DIRECTION FLOW                                  BCOMP   53
C                                                                   BCOMP   54
      DO 70 J=1,NZMAX                                               BCOMP   55
      BPARA(1,J)=VF(1,J)*RHOF(1,J)                                  BCOMP   56
      DO 70 I=2,NRMAX                                               BCOMP   57
      BPARA(I,J)=VF(I,J)*RHOF(I,J)                                  BCOMP   58
      BPERP(I,J)=BPERP(I,J)*RHOF(I,J)                               BCOMP   59
      BNORM(I,J)=BNORM(I,J)*RHOF(I,J)                               BCOMP   60
   70 CONTINUE                                                      BCOMP   61
C                                                                   BCOMP   62
      RETURN                                                        BCOMP   63
      END                                                           BCOMP   64


      SUBROUTINE BELGAM                                             BELGAM   2
      LEVEL 2,NB,NBF,XBF,RBF,ELBF,GAMBF                             BVAL     2
      COMMON /BVAL/ NB,NBF(51),XBF(51,100),RBF(51,100),ELBF(51,100),BVAL     3
     *  GAMBF(51,100)                                               BVAL     4
      COMMON/ONSTRM/ IPLOT,NBLUNT,NZEND,NZADD                       ONSTRM   2
      LEVEL 2, XST,RST,NUM,NST                                      STREAM   2
      COMMON /STREAM/ XST(50,150),RST(50,150),NUM(50),NST           STREAM   3
      COMMON /SHOCKS/ DRSDX(100),DST(50)                            SHOCKS   2
      DATA PIONE /1.57079632/                                       BELGAM   7
C                                                                   BELGAM   8
C     THIS SUBROUTINE CALCULATES THE MAGNITUDE AND DIRECTION OF     BELGAM   9
C     EL/ELINF AT THE POINTS WHERE THE STREAMLINES INTERSECT THE    BELGAM  10
C     MAGNETIC FIELD LINES WHICH ARE PERPENDICULAR TO THE FLOW      BELGAM  11
C     IN FREESTREAM                                                 BELGAM  12
C                                                                   BELGAM  13
      DST(1)=0.5/RST(1,1)                                           BELGAM  14
      DO 10 I=2,NST                                                 BELGAM  15
      DST(I)=1.0/(RST(I,1)-RST(I-1,1))                              BELGAM  16
   10 CONTINUE                                                      BELGAM  17
C                                                                   BELGAM  18
C     SET ARRAYS TO FREE STREAM VALUES                              BELGAM  19
C                                                                   BELGAM  20
      NSTP1=NST+1                                                   BELGAM  21
      JBFMAX=NBF(1)                                                 BELGAM  22
      DO 30 I=1,NSTP1                                               BELGAM  23
      IF (NBF(I) .LT. JBFMAX) GO TO 30                              BELGAM  24
      JBFMAX=NBF(I)                                                 BELGAM  25
   30 CONTINUE                                                      BELGAM  26
      DO 20 J=1,JBFMAX                                              BELGAM  27
      DO 20 I=1,NSTP1                                               BELGAM  28
      ELBF(I,J)=1.0                                                 BELGAM  29
      GAMBF(I,J)=PIONE                                              BELGAM  30
   20 CONTINUE                                                      BELGAM  31
C                                                                   BELGAM  32
C     VALUES ALONG FIELD LINES WHICH CROSS SHOCK                    BELGAM  33
C                                                                   BELGAM  34
      DO 130 J=3,NB                                                 BELGAM  35
      IF (J .GT. NBF(1)) GO TO 110                                  BELGAM  36
      D1=SQRT((XBF(2,J)-XBF(1,J))**2+RBF(2,J)**2)*2.0               BELGAM  37
      ELBF(1,J)=D1*DST(1)                                           BELGAM  38
      GAM1=PIONE                                                    BELGAM  39
      DX2=XBF(3,J)-XBF(2,J)                                         BELGAM  40
      DR2=RBF(3,J)-RBF(2,J)                                         BELGAM  41
      D2=SQRT(DX2*DX2+DR2*DR2)                                      BELGAM  42
      GAM2=ATAN(DR2/DX2)                                            BELGAM  43
      ELBF(2,J)=D1*D2/(D1+D2)*(DST(2)+DST(1))                       BELGAM  44
      GAMBF(2,J)=(GAM1*D2+GAM2*D1)/(D1+D2)                          BELGAM  45
      GO TO 130                                                     BELGAM  46
  110 CONTINUE                                                      BELGAM  47
      D1=SQRT((XBF(2,J)-XBF(2,J-1))**2+(RBF(2,J)-RBF(2,J-1))**2)*2.0 BELGAM  48
      DX2=XBF(3,J)-XBF(2,J)                                         BELGAM  49
      DR2=RBF(3,J)-RBF(2,J)                                         BELGAM  50
      D2=SQRT(DX2*DX2+DR2*DR2)                                      BELGAM  51
      GAM2=ATAN(DR2/DX2)                                            BELGAM  52
      ELBF(2,J)=(ELBF(2,J-1)+D1*DST(2))*D2/(D1+D2)                  BELGAM  53
      GAMBF(2,J)=(D1*GAM2+D2*GAMBF(2,J-1))/(D1+D2)                  BELGAM  54
  130 CONTINUE                                                      BELGAM  55
      DO 120 I=3,NST                                                BELGAM  56
      D1=D2                                                         BELGAM  57
      GAM1=GAM2                                                     BELGAM  58
      IF (XBF(I+1,J) .LT. XST(I,1)) GO TO 120                       BELGAM  59
      DX2=XBF(I+1,J)-XBF(I,J)                                       BELGAM  60
      DR2=RBF(I+1,J)-RBF(I,J)                                       BELGAM  61
      D2=SQRT(DX2*DX2+DR2*DR2)                                      BELGAM  62
      ELBF(I,J)=(1.0+D2/(D1+D2)*(DST(I-1)+DST(I))                   BELGAM  63
      GAM2=ATAN(DR2/DX2)                                            BELGAM  64
      GAMBF(I,J)=(GAM1*D2+GAM2*D1)/(D1+D2)                          BELGAM  65
  120 CONTINUE                                                      BELGAM  66
      XSH=XBF(1+1,J)                                                BELGAM  67
      CALL BSHK(XSH,PS1,GAM,M,ELSH)                                 BELGAM  68
```

```
      DX2=XSH-XBF(I,J)                                                  BELGAM  69
      DR2=RSH-RBF(I,J)                                                  BELGAM  70
      D2=SQRT(DX2*DX2+DR2*DR2)*2.0                                      BELGAM  71
      FLBF(I,J)=D1*(DST(I-1)+D2+ELS4)/(D1+D2)                           BELGAM  72
      GAMBF(I,J)=(D2*GAM1+D1*GAMSH)/(D1+D2)                             BELGAM  73
  130 CONTINUE                                                          BELGAM  74
      JBFMAX=NBF(3)                                                     BELGAM  75
C                                                                       BELGAM  76
C     VALUES ALONG FIELD LINES WHICH END AT DOWNSTREAM BOUNDARY         BELGAM  77
C                                                                       BELGAM  78
      NB1=NB+1                                                          BELGAM  79
      DO 190 J=NB1,JBFMAX                                               BELGAM  80
      IF (J .GT. NBF(1)) GO TO 150                                      BELGAM  81
      D1=SQRT((XBF(2,J)-XBF(1,J))**2+RBF(2,J)**2)*2.0                   BELGAM  82
      FLBF(1,J)=D1*DST(1)                                               BELGAM  83
      GAM1=PION2                                                        BELGAM  84
      DX2=XBF(3,J)-XBF(2,J)                                             BELGAM  85
      DR2=RBF(3,J)-RBF(2,J)                                             BELGAM  86
      D2=SQRT(DX2*DX2+DR2*DR2)                                          BELGAM  87
      GAM2=ATAN(DR2/DX2)                                                BELGAM  88
      FLBF(2,J)=D1*D2/(D1+D2)*(DST(2)+DST(1))                           BELGAM  89
      GAMBF(2,J)=(GAM1*D2+GAM2*D1)/(D1+D2)                              BELGAM  90
      GO TO 160                                                         BELGAM  91
  150 D1=SQRT((XBF(2,J)-XBF(2,J-1))**2+(RBF(2,J)-RBF(2,J-1))**2)*2.0    BELGAM  92
      DX2=XBF(3,J)-XBF(2,J)                                             BELGAM  93
      DR2=RBF(3,J)-RBF(2,J)                                             BELGAM  94
      D2=SQRT(DX2*DX2+DR2*DR2)                                          BELGAM  95
      GAM2=ATAN(DR2/DX2)                                                BELGAM  96
      ELBF(2,J)=(ELBF(2,J-1)+D1*DST(2))*D2/(D1+D2)                      BELGAM  97
      GAMBF(2,J)=(D1*GAM2+D2*GAMBF(2,J-1))/(D1+D2)                      BELGAM  98
  160 CONTINUE                                                          BELGAM  99
      DO 170 I=3,NST                                                    BELGAM 100
      IF (NBF(I+1) .LT. J) GO TO 180                                    BELGAM 101
      D1=D2                                                             BELGAM 102
      GAM1=GAM2                                                         BELGAM 103
      DX2=XBF(I+1,J)-XBF(I,J)                                           BELGAM 104
      DR2=RBF(I+1,J)-RBF(I,J)                                           BELGAM 105
      D2=SQRT(DX2*DX2+DR2*DR2)                                          BELGAM 106
      ELBF(I,J)=D1*D2/(D1+D2)*(DST(I-1)+DST(I))                         BELGAM 107
      GAM2=ATAN(DR2/DX2)                                                BELGAM 108
      GAMBF(I,J)=(GAM1*D2+GAM2*D1)/(D1+D2)                              BELGAM 109
  170 CONTINUE                                                          BELGAM 110
  180 CONTINUE                                                          BELGAM 111
      EL1=D1*DST(I-2)                                                   BELGAM 112
      EL2=D2*DST(I-1)                                                   BELGAM 113
      D1=D1+2.0*D2                                                      BELGAM 114
      D2=-D2                                                            BELGAM 115
      ELBF(I,J)=(EL1*D2+EL2*D1)/(D1+D2)                                 BELGAM 116
      GAMBF(I,J)=(GAM1*D2+GAM2*D1)/(D1+D2)                              BELGAM 117
      IF (GAMBF(I,J) .LT. 0.0) GAMBF(I,J)=0.0                           BELGAM 118
  190 CONTINUE                                                          BELGAM 119
C                                                                       BELGAM 120
C     EXTRAPOLATE ALONG STREAMLINES TO LAST GRID LINE                   BELGAM 121
C                                                                       BELGAM 122
      CALL BSHK1(ZPLOT,Y1,GAM1,EL1)                                     BELGAM 123
      DO 200 I=3,NSTP1                                                  BELGAM 124
      N=NBF(I)                                                          BELGAM 125
      NUMST=NUM(I-1)                                                    BELGAM 126
      IF (XBF(I,N-1) .LT. XST(I-1,1)) GO TO 220                         BELGAM 127
      FAC=(ZPLOT-XBF(I,N-1))/(XBF(I,N)-XBF(I,N-1))                      BELGAM 128
      XBF(I,N+1)=XST(I-1,NUMST)                                         BELGAM 129
      RBF(I,N+1)=RST(I-1,NUMST)                                         BELGAM 130
      ELBF(I,N+1)=FAC*ELBF(I,N)+(1.0-FAC)*ELBF(I,N-1)                   BELGAM 131
      GAMBF(I,N+1)=FAC*GAMBF(I,N)+(1.0-FAC)*GAMBF(I,N-1)                BELGAM 132
      IF (GAMBF(I,N+1) .LT. 0.0) GAMBF(I,N+1)=0.0                       BELGAM 133
      N1=N                                                              BELGAM 134
      I1=I                                                              BELGAM 135
      GO TO 200                                                         BELGAM 136
  220 RBF(I,N+1)=RST(I-1,NUMST)                                         BELGAM 137
      XBF(I,N+1)=XST(I-1,NUMST)                                         BELGAM 138
      FAC=(Y1-RBF(I,N+1))/(Y1-RBF(I1,N1+1))                             BELGAM 139
      ELBF(I,N+1)=ELBF(I1,N1+1)*FAC+(1.0-FAC)*EL1                       BELGAM 140
      GAMBF(I,N+1)=GAMBF(I1,N1+1)*FAC+(1.0-FAC)*GAM1                    BELGAM 141
      IF (GAMBF(I,N+1) .LT. 0.0) GAMBF(I,N+1)=0.0                       BELGAM 142
  200 CONTINUE                                                          BELGAM 143
      RETURN                                                            BELGAM 144
      END                                                               BELGAM 145

      SUBROUTINE BOUND(XMAG,YMAG,XSHOK,YSHOK,NTMAX,IBODY,HRO)           BOUND   2
C                                                                       BOUND   3
C     THIS ROUTINE DRAWS AND LABELS SHOCK WAVE AND                      BOUND   4
C          MAGNETOSPHERE OR IONOSPHERE BOUNDARY.                        BOUND   5
C          UCC PLOT SUBROUTINES USED ARE                                BOUND   6
C               VECTOR,CHAR,POLAR.                                      BOUND   7
C                                                                       BOUND   8
      COMMON /SCALE/ XSF,YSF,XMAX,YMAX,XLNGTH,YLNGTH                    SCALE   2
      DIMENSION R(21),A(21)                                             BOUND  10
      DIMENSION LABMG(2)                                                BOUND  11
      DIMENSION XMAG(NTMAX),YMAG(NTMAX),XSHOK(NTMAX),YSHOK(NTMAX)       BOUND  12
      DATA LABMG(1),LABMG(2),LAB10/10HMAGNETOPAU,2HSE,9HIONOPAUSE/      BOUND  13
      DATA LABSH/10HSHOCK WAVE/                                         BOUND  14
      DATA A/0.,.15708,.31416,.47124,.62832,.78540,                     BOUND  15
     1       .94248,1.09956,1.25664,1.41372,1.57080,                    BOUND  16
     2       1.72788,1.88496,2.04204,2.19911,2.35619,                   BOUND  17
     3       2.51327,2.67035,2.82743,2.98451,3.14159/                   BOUND  18
      DATA R/21*0.1/                                                    BOUND  19
      DATA NP/21/                                                       BOUND  20
C                                                                       BOUND  21
C     DRAW AND LABEL BODY BOUNDARY.                                     BOUND  22
C                                                                       BOUND  23
      CALL VECTOR(XMAG,YMAG,NTMAX,1,0,1H )                              BOUND  24
      XLABL=-1.0+1.0/XSF                                                BOUND  25
      YLABL=0.6                                                         BOUND  26
      IF(IBODY.EQ.1) GO TO 5                                            BOUND  27
      CALL CHAR(XLABL,YLABL,0.0,.12,LABMG,12)                           BOUND  28
      GO TO 10                                                          BOUND  29
C                                                                       BOUND  30
C     IONOPAUSE - ADD HRO TO LABEL                                      BOUND  31
C                                                                       BOUND  32
    5 CALL CHAR(XLABL,YLABL,0.0,.12,LAB10,9)                            BOUND  33
      XCH=-1.5+3.4/XSF                                                  BOUND  34
      YCH=YMAX-0.3/YSF                                                  BOUND  35
      CALL CHAR(XCH,YCH,0.0,.2,3HH/R,3)                                 BOUND  36
      CALL CHAR(XCH+.6/XSF,YCH,0.0,.075,2HO,1)                          BOUND  37
      CALL CHAR(XCH+.75/XSF,YCH,0.0,.2,1H=,1)                           BOUND  38
      CALL NUMPLT(XCH+1./XSF,YCH,0.0,.2,HRO,2)                          BOUND  39
C                                                                       BOUND  40
C     DRAW AND LABEL SHOCK WAVE.                                        BOUND  41
C                                                                       BOUND  42
   10 CONTINUE                                                          BOUND  43
      CALL VECTOR(XSHOK,YSHOK,NTMAX,1,0,1H )                            BOUND  44
      XLABL=-1.0-1.0/XSF                                                BOUND  45
      YLABL=1.4                                                         BOUND  46
      CALL CHAR(XLABL,YLABL,0.0,.12,LABSH,10)                           BOUND  47
C                                                                       BOUND  48
C     DRAW AND LABEL PLANET FOR MAGNETOSPHERE                           BOUND  49
C                                                                       BOUND  50
      IF (IBODY.EQ.1) RETURN                                            BOUND  51
      RMAX=.1*XSF                                                       BOUND  52
      CALL POLAR(R,A,NP,1.0,1H ,-RMAX,1.)                               BOUND  53
      CALL CHAR(.1,-.08,0.0,.12,6HPLANET,6)                             BOUND  54
C                                                                       BOUND  55
      RETURN                                                            BOUND  56
      END                                                               BOUND  57

      SUBROUTINE BOUT                                                   BOUT    2
C                                                                       BOUT    3
C     THIS SUBROUTINE PRINTS OUT THE MAGNETIC FIELD                     BOUT    4
C                                                                       BOUT    5
      COMMON /BIN/ ANGP,ANGN,K8CON,BCON(20)                             BIN     2
      COMMON/ONSTRM/ ZPLOT,NBLUNT,NZEND,NZADD                           ONSTRM  2
      COMMON /DKD/ ANG(20,100),DTH(100),DRS(100),R1,JX,DEG              BOUT    6
      LEVEL 2, BPARA,BPERP,BNORM,BMAG,BANG                              BCOMPS  2
      COMMON /BCOMPS/ BPARA(20,100),BPERP(20,100),BNORM(20,100),        BCOMPS  3
     *  BMAG(20,100),BANG(20,100)                                       BCOMPS  4
      COMMON/CONT/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100), TRANSF1  2
     *  VF(20,100),RHOF(20,100)                                         TRANSF1  3
      COMMON /BLUNT/ THETA(25),RP(2),25)                                TRANSF1  4
      LEVEL 2, VX,VY                                                    TRANSF1  5
      COMMON/VCOMP/ VX(20,100),VY(20,100)                               TRANSF1  6
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,NZMAX, BOUT   11
     1  AMACH,GAMMA,HRO,NHINDX                                          BOUT   12
C                                                                       BOUT   13
      ANGP=ANGP/DEG                                                     BOUT   14
      SANGP=SIN(ANGP)                                                   BOUT   15
      CANGP=COS(ANGP)                                                   BOUT   16
      ANGN=ANGN/DEG                                                     BOUT   17
      SANGN=SIN(ANGN)                                                   BOUT   18
      CANGN=COS(ANGN)                                                   BOUT   19
      WRITE(6,100)                                                      BOUT   20
C                                                                       BOUT   21
C     PRINT MAGNETIC FIELD FOR NOSE REGION                              BOUT   22
C                                                                       BOUT   23
      DO 10 J=1,NBLUNT                                                  BOUT   24
      IF (NHINDX .EQ. 1) GO TO 3                                        BOUT   25
      WRITE(6,110) J,THETA(J)                                           BOUT   26
      GO TO 5                                                           BOUT   27
    3 WRITE(6,115) J,THETA(J)                                           BOUT   28
    5 I=1                                                               BOUT   29
      WRITE(6,120) I,RP(I,J),BPARA(I,J)                                 BOUT   30
```

```
      DO 10 I=2,NRMAX                                                   BOUT   31
      B1=SANGP*BPERP(I,J)                                               BOUT   32
      B2=CANGP*BPARA(I,J)                                               BOUT   33
      F1=COS(ANG(I,J))*B2+COS(BANG(I,J))*B1                             BOUT   34
      F2=SIN(ANG(I,J))*B2+SIN(BANG(I,J))*B1                             BOUT   35
      BMAG(I,J)=SQRT(F1*F1+F2*F2)                                       BOUT   36
      BANG(I,J)=ATAN2(F2,F1)*DEG                                        BOUT   37
      BX=F1*CANGN                                                       BOUT   38
      BY=F2*CANGN                                                       BOUT   39
      BZ=BNORM(I,J)*SANGN                                               BOUT   40
      WRITE(6,120) I,RP(I,J),BPARA(I,J),BPERP(I,J),BMAG(I,J),BANG(I,J),  BOUT   41
     *  BNORM(I,J),BX,BY,BZ                                             BOUT   42
   10 CONTINUE                                                          BOUT   43
C                                                                       BOUT   44
C     PRINT MAGNETIC FIELD DOWNSTREAM                                   BOUT   45
C                                                                       BOUT   46
      DO 20 JZ=1,NZADD                                                  BOUT   47
      J=J2+NBLUNT                                                       BOUT   48
      Z=-YC(1,J)                                                        BOUT   49
      IF (NHINDX .EQ. 1) GO TO 13                                       BOUT   50
      WRITE(6,130) JZ,Z                                                 BOUT   51
      GO TO 15                                                          BOUT   52
   13 WRITE(6,135) JZ,Z                                                 BOUT   53
   15 I=1                                                               BOUT   54
      WRITE(6,120) I,YC(1,J),BPARA(I,J)                                 BOUT   55
      DO 20 I=2,NRMAX                                                   BOUT   56
      B1=SANGP*BPERP(I,J)                                               BOUT   57
      B2=CANGP*BPARA(I,J)                                               BOUT   58
      F1=COS(ANG(I,J))*B2+COS(BANG(I,J))*B1                             BOUT   59
      F2=SIN(ANG(I,J))*B2+SIN(BANG(I,J))*B1                             BOUT   60
      BMAG(I,J)=SQRT(F1*F1+F2*F2)                                       BOUT   61
      BANG(I,J)=ATAN2(F2,F1)*DEG                                        BOUT   62
      BX=F1*CANGN                                                       BOUT   63
      BY=F2*CANGN                                                       BOUT   64
      BZ=BNORM(I,J)*SANGN                                               BOUT   65
      WRITE(6,120) I,YC(I,J),BPARA(I,J),BPERP(I,J),BMAG(I,J),BANG(I,J),  BOUT   66
     *  BNORM(I,J),BX,BY,BZ                                             BOUT   67
   20 CONTINUE                                                          BOUT   68
      RETURN                                                            BOUT   69
C                                                                       BOUT   70
  100 FORMAT(1H1//52X,25HMAGNETIC FIELD COMPONENTS/52X,25(1H*)/)        BOUT   71
  110 FORMAT(//21H ANGULAR LOCATION NO.,I2,12H, AT THETA =,F8.4,        BOUT   72
     *  8H DEGREES//                                                    BOUT   73
     *  4X,1HI,6X,4HRP/D,8X,2(6HB/BINF,6X),6HB/BINF,5X,7HB-ANGLE,5X,    BOUT   74
     *  7H B/BINF,5X,7HBX/BINF,5X,7HBY/BINF,5X,7HBZ/BINF/              BOUT   75
     *  21X,1GH(PARALLEL),4X,6H(PERP),4X,2(10H(IN-PLANE),2X),1X,       BOUT   76
     *  6H(NORMAL),1X,3(1X,11H(RESULTANT)))                            BOUT   77
  115 FORMAT(//21H ANGULAR LOCATION NO.,I2,12H, AT THETA =,F8.4,        BOUT   78
     *  8H DEGREES//                                                    BOUT   79
     *  4X,1HI,6X,5HRP/RO,7X,2(6HB/BINF,6X),6HB/BINF,5X,7HB-ANGLE,5X,   BOUT   80
     *  7H B/BINF,5X,7HBX/BINF,5X,7HBY/BINF,5X,7HBZ/BINF/              BOUT   81
     *  21X,1GH(PARALLEL),4X,6H(PERP),4X,2(10H(IN-PLANE),2X),1X,       BOUT   82
     *  6H(NORMAL),1X,3(1X,11H(RESULTANT)))                            BOUT   83
  120 FORMAT(I5,5(2X,F10.4))                                           BOUT   84
  130 FORMAT(//20H ADDITIONAL AXIAL LOCATION NO.,I2,10H, AT X/D =,F8.4// BOUT   85
     *  4X,1HI,7X,3HR/D,8X,2(6HB/BINF,6X),6HB/BINF,5X,7HB-ANGLE,5X,     BOUT   86
     *  7H B/BINF,5X,7HBX/BINF,5X,7HBY/BINF,5X,7HBZ/BINF/              BOUT   87
     *  21X,1GH(PARALLEL),4X,6H(PERP),4X,2(10H(IN-PLANE),2X),1X,       BOUT   88
     *  6H(NORMAL),1X,3(1X,11H(RESULTANT)))                            BOUT   89
  135 FORMAT(//30H ADDITIONAL AXIAL LOCATION NO.,I2,11H, AT X/RO =,F8.4/ BOUT   90
     *  4X,1HI,7X,4HR/RO,7X,2(6HB/BINF,6X),6HB/BINF,5X,7HB-ANGLE,5X,    BOUT   91
     *  7H B/BINF,5X,7HBX/BINF,5X,7HBY/BINF,5X,7HBZ/BINF/              BOUT   92
     *  21X,1GH(PARALLEL),4X,6H(PERP),4X,2(10H(IN-PLANE),2X),1X,       BOUT   93
     *  6H(NORMAL),1X,3(1X,11H(RESULTANT)))                            BOUT   94
      END                                                               BOUT   95


      SUBROUTINE BSHK1 (X,R,ANG,EL)                                     BSHK1    2
C                                                                       BSHK1    3
C     THIS SUBROUTINE CALCULATES THE MAGNITUDE AND DIRECTION OF         BSHK1    4
C     EL/ELINF AT THE SHOCK, AND THE R-LOCATION, GIVEN THE              BSHK1    5
C     X-LOCATION OF THE POINT                                           BSHK1    6
C                                                                       BSHK1    7
      COMMON /BOUND/ XBOD(103),YBOD(100),XSHK(100),YSHK(100),NRMAX,     BSHK1    8
     *               NZMAX,AMACH,GAM,HRO,NHINDX                         BSHK1    9
      COMMON /SHOCKS/ DRSDX(100),DST(50)                                SHOCKS   2
      DO 10 J=1,NZMAX                                                   BSHK1   11
      IF (X .LT. XSHK(J)) GO TO 20                                      BSHK1   12
   10 CONTINUE                                                          BSHK1   13
      J=NZMAX                                                           BSHK1   14
   20 R=YSHK(J-1)+(X-XSHK(J-1))*(YSHK(J)-YSHK(J-1))/(XSHK(J)-XSHK(J-1)) BSHK1   15
      THET1=ATAN(DRSDX(J-1))                                            BSHK1   16
      THET2=ATAN(DRSDX(J))                                              BSHK1   17
      THET=THET1+(X-XSHK(J-1))*(THET2-THET1)/(XSHK(J)-XSHK(J-1))        BSHK1   18
C                                                                       BSHK1   19
C     CALCULATE MAGNITUDE OF VERTICAL FIELD COMPONENT OF SHOCK          BSHK1   20
C                                                                       BSHK1   21
      S2=SIN(THET)**2                                                   BSHK1   22
      EM2=AMACH**2                                                      BSHK1   23
      GAM2=(GAM+1.0)*0.5*EM2                                            BSHK1   24
      DD2=1.0-(EM2*S2-1.0)*(GAM*EM2*S2+1.0)/(GAM2*GAM2*S2)              BSHK1   25
      DD=SQRT(DD2)                                                      BSHK1   26
      COTH=1.0/TAN(THET)                                                BSHK1   27
      TD=GAM2/(EM2*S2-1.0)-1.0                                          BSHK1   28
      DELT=ATAN(COTH/TD)                                                BSHK1   29
      EL=SQRT(1.0+COTH*COTH*(1.0+DD2)-2.0*DD*(TD+1.0)*COTH*SIN(DELT))   BSHK1   30
C                                                                       BSHK1   31
C     CALCULATE ANGLE OF VERTICAL FIELD COMPONENT                       BSHK1   32
C                                                                       BSHK1   33
      ALPHA=ASIN(DD*COTH*SIN(THET-DELT)/EL)                             BSHK1   34
      ANG=THET+ALPHA                                                    BSHK1   35
      RETURN                                                            BSHK1   36
      END                                                               BSHK1   37


      SUBROUTINE BSTEP                                                  BSTEP    2
C                                                                       BSTEP    3
C     THIS SUBROUTINE CALCULATES THE VERTICAL FIELD LINES,              BSTEP    4
C        BY INTEGRATING ALONG EACH STREAMLINE TO LOCATE POSITIONS        BSTEP    5
C        AT EQUAL TIME INTERVALS                                         BSTEP    6
C                                                                       BSTEP    7
      COMMON/CON1/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100),  TRANSF1   2
     *  VF(20,100),RHOF(20,100)                                         TRANSF1   3
      COMMON /BLUNT/ THETAI(25),RP(2),2)                                TRANSF1   4
      LEVEL 2, VX,VY                                                    TRANSF1   5
      COMMON/VCOMP/ VX(20,100),VY(20,100)                               TRANSF1   6
      COMMON/DNSTRM/ ZPLOT,NBLUNT,NZEND,NZADD                           DNSTRM    2
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,NZMAX, BSTEP   10
     1  AMACH,GAMMA,HRO,NHINDX                                          BSTEP   11
      LEVEL 2,NB,NBF,XBF,RBF,ELBF,GAMBF                                 BVAL      2
      COMMON /BVAL/ NB,NBF(51),XBF(51,100),RBF(51,100),ELBF(51,100),    BVAL      3
     *  GAMBF(51,100)                                                   BVAL      4
      LEVEL 2, XST,RST,NUM,NST                                          STREAM    2
      COMMON /STREAM/ XST(50,150),RST(50,150),NUM(50),NST               STREAM    3
C                                                                       BSTEP   14
C     SECOND FIELD LINE IS TANGENT TO SHOCK NOSE                        BSTEP   15
C                                                                       BSTEP   16
      JBFMAX=99                                                         BSTEP   17
      NB=MIN1(ZPLOT*15.0+14.0,30.0)                                     BSTEP   18
      XSHKZ=XC(NRMAX,1)                                                 BSTEP   19
      DSINF=(XST(NST,1)-XSHKZ)/(FLOAT(NB)-1.5)                          BSTEP   20
      DELT=DSINF                                                        BSTEP   21
C                                                                       BSTEP   22
C     CALCULATE WHERE VERTICAL FIELD LINES CROSS SYMMETRY AXIS          BSTEP   23
C                                                                       BSTEP   24
      DO 600 J=1,JBFMAX                                                 BSTEP   25
      RBF(1,J)=0.0                                                      BSTEP   26
  600 CONTINUE                                                          BSTEP   27
      XBF(1,1)=XSHKZ-DSINF                                              BSTEP   28
      XBF(1,2)=XSHKZ                                                    BSTEP   29
C                                                                       BSTEP   30
C     ASSUME CONSTANT DECELERATION BETWEEN GRID POINTS                  BSTEP   31
C                                                                       BSTEP   32
      J=2                                                               BSTEP   33
      TOLD=0.0                                                          BSTEP   34
      IR=NRMAX                                                          BSTEP   35
      V2=VF(IR,1)                                                       BSTEP   36
  610 V1=V2                                                             BSTEP   37
      IR=IR-1                                                           BSTEP   38
      IF (IR .LE. 0) GO TO 640                                          BSTEP   39
      V2=VF(IR,1)                                                       BSTEP   40
      DT=2.0*(XC(IR,1)-XC(IR+1,1))/(V2+V1)                              BSTEP   41
      A=(V2-V1)/DT                                                      BSTEP   42
      TNEW=TOLD+DT                                                      BSTEP   43
      IF (TNEW .GE. DELT) GO TO 620                                     BSTEP   44
      TOLD=TNEW                                                         BSTEP   45
      GO TO 610                                                         BSTEP   46
  620 DT=DELT-TOLD                                                      BSTEP   47
      DELS=V1*DT+0.5*A*DT*DT                                            BSTEP   48
      J=J+1                                                             BSTEP   49
      XBF(1,J)=XC(IR+1,1)+DELS                                          BSTEP   50
      TOLD=TNEW-DELT                                                    BSTEP   51
  630 IF (J .GE. JBFMAX) GO TO 640                                      BSTEP   52
      IF (TOLD .LT. DELT) GO TO 610                                     BSTEP   53
      V1=V1+A*DT                                                        BSTEP   54
      DELS=V1*DELT+0.5*A*DELT**2                                        BSTEP   55
      TOLD=TOLD-DELT                                                    BSTEP   56
      DT=DELT                                                           BSTEP   57
```

```
      J=J+1                                                             BSTEP   56
      XBF(1,J)=XBF(1,J-1)+DELS                                          BSTEP   57
      GO TO 630                                                         BSTEP   58
  640 CONTINUE                                                          BSTEP   59
      NBF(I)=J                                                          BSTEP   60
      I=1                                                               BSTEP   61
C                                                                       BSTEP   62
C     CALCULATE WHERE VERTICAL FIELD LINES CROSS STREAMLINES            BSTEP   63
C                                                                       BSTEP   64
      DO 700 IM=1,NST                                                   BSTEP   65
      I=IM+1                                                            BSTEP   66
      XBF(1,I)=XSHKZ-DSINF                                              BSTEP   67
      RBF(I,1)=RST(IM,1)                                                BSTEP   68
      XBF(I,2)=XSHKZ                                                    BSTEP   69
      RBF(1,2)=RST(IM,1)                                                BSTEP   70
C                                                                       BSTEP   71
C     LOCATE POINTS BEFORE SHOCK WAVE                                   BSTEP   72
C                                                                       BSTEP   73
      DO 710 J=3,ND                                                     BSTEP   74
      IF (XBF(I,J-1)+DSINF .GT. XST(IM,1)) GO TO 720                    BSTEP   75
      XBF(I,J)=XBF(I,J-1)+DSINF                                         BSTEP   76
      RBF(I,J)=RST(IM,1)                                                BSTEP   77
  710 CONTINUE                                                          BSTEP   78
C                                                                       BSTEP   79
C     LOCATE POINTS WITHIN THE MAGNETOSPHERE                            BSTEP   80
C                                                                       BSTEP   81
  720 TOLD=(XST(IM,1)-XBF(I,J-1))                                       BSTEP   82
      J=J-1                                                             BSTEP   83
      KST=1                                                             BSTEP   84
      IF (IM .GE. NBLUNT) GO TO 810                                     BSTEP   85
      V2=VF(NRMAX,1)                                                    BSTEP   86
      GO TO 730                                                         BSTEP   87
  810 DO 820 JJ=NBLUNT,NZMAX                                            BSTEP   88
      IF (XST(IM,1) .GT. XC(NRMAX,JJ)) GO TO 820                        BSTEP   89
      V2=VF(NRMAX,JJ-1)+(VF(NRMAX,JJ)-VF(NRMAX,JJ-1))                   BSTEP   90
     *    *(XST(IM,1)-XC(NRMAX,JJ-1))/(XC(NRMAX,JJ)-XC(NRMAX,JJ-1))     BSTEP   91
      GO TO 730                                                         BSTEP   92
  820 CONTINUE                                                          BSTEP   93
  730 V1=V2                                                             BSTEP   94
      X1=XST(IM,KST+1)                                                  BSTEP   95
      Y1=RST(IM,KST+1)                                                  BSTEP   96
      V2=VINTPP(X1,Y1)                                                  BSTEP   97
      DST=SQRT((XST(IM,KST+1)-XST(IM,KST))**2+(RST(IM,KST+1)            BSTEP   98
     *    -RST(IM,KST))**2)                                             BSTEP   99
      DT=2.0*DST/(V1+V2)                                                BSTEP  100
      A=(V2-V1)/DT                                                      BSTEP  101
      TNEW=TOLD+DT                                                      BSTEP  102
      IF(TNEW.GE.DELT) GO TO 740                                        BSTEP  103
      KST=KST+1                                                         BSTEP  104
      TOLD=TNEW                                                         BSTEP  105
      IF (KST .GE. NUM(IM)) GO TO 760                                   BSTEP  106
      GO TO 730                                                         BSTEP  107
  740 J=J+1                                                             BSTEP  108
      DT=DELT-TOLD                                                      BSTEP  109
      DELS=V1*DT+0.5*A*DT*DT                                            BSTEP  110
      XBF(I,J)=XST(IM,KST)+DELS*(XST(IM,KST+1)-XST(IM,KST))/DST         BSTEP  111
      RBF(I,J)=RST(IM,KST)+DELS*(RST(IM,KST+1)-RST(IM,KST))/DST         BSTEP  112
      TOLD=TNEW-DELT                                                    BSTEP  113
      KST=KST+1                                                         BSTEP  114
      IF (KST .GE. NUM(IM)) GO TO 760                                   BSTEP  115
  750 IF (J .GE. JBFMAX) GO TO 760                                      BSTEP  116
      IF (TOLD .LT. DELT) GO TO 730                                     BSTEP  117
      J=J+1                                                             BSTEP  118
      V1=V1+A*DT                                                        BSTEP  119
      DELS=V1*0.5*T+0.5*A*DELT*DELT                                     BSTEP  120
      XBF(I,J)=XBF(I,J-1)+DELS*(XST(IM,KST)-XST(IM,KST-1))/DST          BSTEP  121
      RBF(I,J)=RBF(I,J-1)+DELS*(RST(IM,KST)-RST(IM,KST-1))/DST          BSTEP  122
      DT=DELT                                                           BSTEP  123
      TOLD=TOLD-DELT                                                    BSTEP  124
      GO TO 750                                                         BSTEP  125
  760 CONTINUE                                                          BSTEP  126
      NBF(I)=J                                                          BSTEP  127
      IF (J .GT. NBF(I-1)) NBF(I)=NBF(I-1)                             BSTEP  128
  700 CONTINUE                                                          BSTEP  129
      RETURN                                                            BSTEP  130
      END                                                               BSTEP  131
                                                                        BSTEP  132
                                                                        BSTEP  133

      SUBROUTINE BUBBL                                                  BUBBL    2
C                                                                       BUBBL    3
C     USING A BUBBLE TECHNIQUE, THIS ROUTINE SORTS THE REAL             BUBBL    4
C          ARRAY S INTO ASCENDING ORDER AND CHANGES                     BUBBL    5
C          THE ORDER OF ARRAYS A AND B IN A CORRESPONDING               BUBBL    6
C          MANNER.  K IS THE NUMBER OF DATA POINTS TO                   BUBBL    7
C          BE SORTED.                                                   BUBBL    8
C                                                                       BUBBL    9
      COMMON /LABELS/ A(30),S(30),B(30),K,ILB(30),NLAB                  BUBBL   10
C                                                                       BUBBL   11
      IF(K.EQ.1) RETURN                                                 BUBBL   12
      K1=K-1                                                            BUBBL   13
      DO 100 I=1,K1                                                     BUBBL   14
      L=I+1                                                             BUBBL   15
      DO 100 J=L,K                                                      BUBBL   16
      IF(S(J).GT.S(I)) GO TO 100                                        BUBBL   17
C                                                                       BUBBL   18
C     INTERCHANGE ARRAYS                                                BUBBL   19
C                                                                       BUBBL   20
      TEMP=S(I)                                                         BUBBL   21
      S(I)=S(J)                                                         BUBBL   22
      S(J)=TEMP                                                         BUBBL   23
      TEMP=A(I)                                                         BUBBL   24
      A(I)=A(J)                                                         BUBBL   25
      A(J)=TEMP                                                         BUBBL   26
      TEMP=B(I)                                                         BUBBL   27
      B(I)=B(J)                                                         BUBBL   28
      B(J)=TEMP                                                         BUBBL   29
  100 CONTINUE                                                          BUBBL   30
C                                                                       BUBBL   31
      RETURN                                                            BUBBL   32
      END                                                               BUBBL   33


      SUBROUTINE CHECK(ICHK,NXY,KOD2,J,K,NVAL,KOD9)                     CHECK    2
C                                                                       CHECK    3
C        CONTOUR PROGRAMS MAP, WALK, SERCH, ENTER, AND CHECK            CHECK    4
C        WRITTEN BY REESE SORENSON, NASA-AMES RES. CTR., AUG., 1974.    CHECK    5
C          (MODIFIED VERSION)                                           CHECK    6
C                                                                       CHECK    7
C        GIVEN THAT A LINE PASSES THROUGH AN INTERVAL UNDER             CHECK    8
C        INVESTIGATION.  QUESTION: IS IT A NEW LINE, OR IS IT           CHECK    9
C        PART OF A LINE ALREADY RECORDED?                               CHECK   10
C                                                                       CHECK   11
C        KOD9    =1      OK, NEW POINT.                                 CHECK   12
C        KOD9    =2      BAD, OLD POINT.                                CHECK   13
C                                                                       CHECK   14
      DIMENSION ICHK(4,1)                                               CHECK   15
C                                                                       CHECK   16
      DO 1 L=1,NXY                                                      CHECK   17
      IF(KOD2.NE.ICHK(1,L))  GO TO 1                                    CHECK   18
      IF(J.NE.ICHK(2,L))  GO TO 1                                       CHECK   19
      IF(K.NE.ICHK(3,L))  GO TO 1                                       CHECK   20
      IF(NVAL.EQ.ICHK(4,L))  GO TO 2                                    CHECK   21
    1 CONTINUE                                                          CHECK   22
      KOD9=1                                                            CHECK   23
      GO TO 3                                                           CHECK   24
    2 KOD9=2                                                            CHECK   25
    3 RETURN                                                            CHECK   26
      END                                                               CHECK   27


      SUBROUTINE CONOUT (ACONT,FACT,NHINDX)                             CONOUT   2
C                                                                       CONOUT   3
C          THIS ROUTINE WRITES OUT THE CONTOUR LINES FOUND BY           CONOUT   4
C          SUBROUTINE MAP                                               CONOUT   5
C                                                                       CONOUT   6
      COMMON /PLOTC/ X(1000),Y(1000),CVAL(30),NAD(30),IPLOT             CONOUT   7
      COMMON /COM2/ ICHK(4,1000),DUM(500)                               CONOUT   8
      DIMENSION ACONT(1)                                                CONOUT   9
C                                                                       CONOUT  10
C     WRITE HEADING FOR THIS SET OF CONTOURS                            CONOUT  11
C                                                                       CONOUT  12
      NMAX=NAD(1)                                                       CONOUT  13
      IF(IPLOT.EQ.2) GO TO 10                                           CONOUT  14
      IF (IPLOT .GT. 4) GO TO 15                                        CONOUT  15
      WRITE(6,600) NMAX                                                 CONOUT  16
      GO TO 20                                                          CONOUT  17
   10 WRITE(6,500) NMAX                                                 CONOUT  18
      GO TO 20                                                          CONOUT  19
   15 IF (IPLOT .EQ. 5) WRITE(6,600) NMAX                               CONOUT  20
      IF (IPLOT .EQ. 6) WRITE(6,610) NMAX                               CONOUT  21
   20 CONTINUE                                                          CONOUT  22
C                                                                       CONOUT  23
C     REVERSE SIGN OF X FOR OUTPUT                                      CONOUT  24
C                                                                       CONOUT  25
      JMAX=NAD(NMAX+1)-1                                                CONOUT  26
      DO 2 J=1,JMAX                                                     CONOUT  27
    2 X(J)=-X(J)                                                        CONOUT  28
C                                                                       CONOUT  29
C     PRINT CONTOUR LINE FOR EACH VALUE                                 CONOUT  30
```

```
C
      NAD(1)=1
      DO 1 K=1,KMAX
      NP=NAD(N+1)-NAD(N)
      NCONT=NAD(N)
      L=IC+K(4,NCONT)
      CVAL(N)=ACONT(L)
      M1=NAD(N)
      M2=NAD(N+1)-1
      IF(IPLOT.EQ.2) GO TO 30
      IF (IPLOT .NE. 4) GO TO 35
      TVAL=1.0+FACT*(1.0-CVAL(N)**2)
      WRITE(6,41C) NP,CVAL(N),TVAL
      GO TO 40
   30 WRITE(6,,21C) NP,CVAL(N)
      GO TO 40
   35 IF (IPLOT .EQ. 5) WRITE(6,620) NP,CVAL(N)
      IF (IPLOT .EQ. 6) WRITE(6,630) NP,CVAL(N)
   40 IF(NMINDX.EQ.1) GO TO 45
      WRITE(6,42C)
      GO TO 50
   45 WRITE(6,425)
   50 CONTINUE
      WRITE(6,435) (X(I),Y(I),I=M1,M2)
    1 CONTINUE
      NAD(1)=NMAX
C
C     RESTORE SIGN OF X
C
      DO 3 J=1,JMAX
    3 X(J)=-X(J)
      RETURN
C
  4CC FORMAT(1H1///49X,33HVELOCITY AND TEMPERATURE CONTOURS/49X,33(1H*)///
     *       10X,I3,45H VELOCITY (TEMPERATURE) CONTOUR LINES FOUND)
  41C FORMAT(///15X,I3,35H POINTS IN CONTOUR LINE OF V/VINF =,F7.3,
     *  11H,  T/TINF =,F7.3///)
  42C FORMAT(14X,3HX/D,17X,3HR/D/)
  425 FORMAT(14X,4HX/RO,16X,4HR/RO/)
  43C FORMAT(9X,F10.4,10X,F10.4)
  5CC FORMAT(1H1///57X,16HDENSITY CONTOURS/57X,16(1H*)///
     *       10X,I3,26H DENSITY CONTOUR LINES FOUND)
  51C FORMAT(///15X,I3,34H POINTS IN CONTOUR LINE OF RHO/RHOINF =,F7.3///)
  6CC FORMAT(1H1///53X,23HMAGNETIC FIELD CONTOURS/53X,23(1H*)///
     *       10X,I3,35H MAGNETIC FIELD CONTOUR LINES FOUND/
     *       14X,33H(FOR COMPONENT ALONG FIELD LINES ,
     *       31HPARALLEL TO FLOW IN FREESTREAM))
  61C FORMAT(1H1///10X,I3,35H MAGNETIC FIELD CONTOUR LINES FOUND/
     *       14X,33H(FOR COMPONENT ALONG FIELD LINES ,
     *       36HPERPENDICULAR TO FLOW IN FREESTREAM))
  62C FORMAT(///15X,I3,34H POINTS IN CONTOUR LINE OF B/BINF ,
     1     12H(PARALLEL) =,F7.3///)
  63C FORMAT(///15X,I3,34H POINTS IN CONTOUR LINE OF B/BINF ,
     1     14H(PERPENDICULAR) =,F7.3///)
      END


      SUBROUTINE CONTR
C
C     THIS SUBROUTINE PLOTS AND LABELS CONTOUR LINES.
C     ALSO DRAWS FIELD LINES FOR MAGNETIC FIELD STRENGTH PLOTS.
C          UCC PLOT SUBROUTINES USED ARE
C               DUTLN,NUMPLT,VECTOR.
C
      DIMENSION XS(150),YS(150)
      LEVEL 2, XSTRM,YSTRM,NUM,NST
      COMMON /LABELS/ XLAB(30),YLAB(30),CV(30),NCL,ILB(30),NLAB
      COMMON /SCALE/ XSF,YSF,XMAX,YMAX,XLNGTH,YLNGTH
      COMMON/STREAM/ XSTRM(50,150),YSTRM(50,150),NUM(50),NST
      LEVEL 2,NB,NBF,XBF,RBF,ELBF,GAMBF
      COMMON /BVAL/ NB,NBF(51),XBF(51,100),RBF(51,100),ELBF(51,100),
     *  GAMBF(51,100)
      COMMON/ONSTRM/ IPLOT,NBLUNT,NZERO,NZADD
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NKMAX,
     1              NTMAX,AMACH,GAM,RHO,NHINDX
      COMMON /PLOTC/ X(100C),Y(100C),CVAL(30),NAD(30),IPLOT
C
      IF(IPLOT.EQ.3) GO TO 70
C
C     DRAW CONTOUR LINES.
C
      IB=1
      NCL=NAD(1)
      DO 50 N=1,NCL
      IC=NAD(N+1)
      NPTS=IC-IB
      CALL VECTOR(X(IB),Y(IB),NPTS,1,0,1H )
C
C     LABEL CONTOURS.
C
C     DETERMINE WHICH END OF CONTOUR LINE IS CLOSER
C          TO BOUNDARY.
C          LABEL CONTOUR AT THAT END.
C
C
C     FIND MINIMUM DISTANCE FROM CONTOUR END POINTS
C          TO BOUNDARY.
C
      K=1
      XC=X(IB)
      YC=Y(IB)
    5 DMIN=999.
      DO 10 I=1,NTMAX
      D=SQRT((XBOD(I)-XC)**2+(YBOD(I)-YC)**2)
      IF(D.GE.DMIN) GO TO 10
      DMIN=D
   10 CONTINUE
      IF(K.EQ.2) GO TO 15
      DMIN1=DMIN
      XC=X(IE-1)
      YC=Y(IE-1)
      K=2
      GO TO 5
   15 CONTINUE
      DMIN2=DMIN
C
C     DETERMINE WHICH END OF CONTOUR LINE IS
C          CLOSER TO BOUNDARY.
C
      IF(DMIN1.LT.DMIN2) GO TO 20
      XLAB(N)=X(IE-1)+.1/XSF
      YLAB(N)=Y(IE-1)-.1/YSF
      GO TO 30
   20 XLAB(N)=X(IB)+.1/XSF
      YLAB(N)=Y(IB)-.1/YSF
   30 CONTINUE
      CV(N)=CVAL(N)
      IB=NAD(N+1)
   50 CONTINUE
C
C     SUBROUTINE LABEL CHECKS THE ARRAYS OF LABEL
C          COORDINATES FOR OVERLAPS.  IF A LABEL OVERLAPS,
C          IT IS NOT SAVED UNLESS IT DEFINES A MINIMUM
C          OR MAXIMUM CONTOUR VALUE.
C          ARRAY ILB CONTAINS INDICES OF LABELS TO BE
C          PLOTTED.
C
      CALL LABEL
C
C     LABEL CONTOUR LINES
C
      DO 60 I=1,NLAB
      N=ILB(I)
      CALL NUMPLT(XLAB(N),YLAB(N),0.07,.1,CV(N),2)
   60 CONTINUE
      IF (IPLOT .EQ. 5) GO TO 100
      IF (IPLOT .EQ. 6) GO TO 150
      RETURN
C
C     DRAW STREAMLINES
C
   70 CONTINUE
      DO 75 K=1,NST
      NN=NUM(K)
      XS(1)=XSTRM(K,1)-0.2
      YS(1)=YSTRM(K,1)
      DO 5C N=1,NN
      XS(N+1)=XSTRM(K,N)
      YS(N+1)=YSTRM(K,N)
   80 CONTINUE
      CALL VECTOR(XS,YS,NN+1,1,0,1H )
   75 CONTINUE
      RETURN
C
C     DRAW STREAMLINES FOR MAGNETIC FIELD PLOT
C
  100 DO 110 K=2,NST,3
```

```
      NN=NUM(K)                                                         CONTR 110
      XS(1)=XSTRM(K,1)-0.2                                              CONTR 111
      YS(1)=YSTRM(K,1)                                                  CONTR 112
      DO 120 N=1,NN                                                     CONTR 113
      XS(N+1)=XSTRM(K,N)                                                CONTR 114
      YS(N+1)=YSTRM(K,N)                                                CONTR 115
  120 CONTINUE                                                          CONTR 116
      DO 130 N=1,NN                                                     CONTR 117
      CALL DUTLN(XS(N),YS(N),XS(N+1),YS(N+1),20,0)                      CONTR 118
  130 CONTINUE                                                          CONTR 119
  110 CONTINUE                                                          CONTR 120
      RETURN                                                            CONTR 121
C                                                                       CONTR 122
C     DRAW PERPENDICULAR FIELD LINES FOR MAGNETIC FIELD PLOT            CONTR 123
C                                                                       CONTR 124
  150 CONTINUE                                                          CONTR 125
      XS(1)=XBF(2,1)                                                    CONTR 126
      XS(2)=XBF(2,1)                                                    CONTR 127
      YS(1)=0.0                                                         CONTR 128
      YS(2)=0.4                                                         CONTR 129
      CALL DUTLN(XS(1),YS(1),XS(2),YS(2),20,0)                          CONTR 130
      NS=NBF(1)-7                                                       CONTR 131
      DO 160 K=2,NB,2                                                   CONTR 132
      DO 170 J=1,NTMAX                                                  CONTR 133
      IF (XSHK(J) .LT. XBF(NST+2,K)) GO TO 170                          CONTR 134
      GO TO 180                                                         CONTR 135
  170 CONTINUE                                                          CONTR 136
      J=NTMAX                                                           CONTR 137
  180 CONTINUE                                                          CONTR 138
      RSH=YSHK(J-1)+(XBF(NST,K)-XSHK(J-1))*(YSHK(J)-YSHK(J-1))          CONTR 139
     *      /(XSHK(J)-XSHK(J-1))                                        CONTR 140
      DO 190 J=1,NST                                                    CONTR 141
      IF (XBF(J,K) .GT. RSH) GO TO 200                                  CONTR 142
      XS(J)=XBF(J,K)                                                    CONTR 143
      YS(J)=RBF(J,K)                                                    CONTR 144
  190 CONTINUE                                                          CONTR 145
  200 XS(J)=XBF(NST+1,K)                                                CONTR 146
      YS(J)=RSH                                                         CONTR 147
      XS(J+1)=XS(J)                                                     CONTR 148
      YS(J+1)=AMIN1(RSH+0.2,YMAX)                                       CONTR 149
      NN=J                                                              CONTR 150
      DO 210 N=1,NN                                                     CONTR 151
      CALL DUTLN(XS(N),YS(N),XS(N+1),YS(N+1),20,0)                      CONTR 152
  210 CONTINUE                                                          CONTR 153
      IF (NS .LT. K) RETURN                                             CONTR 154
  160 CONTINUE                                                          CONTR 155
      NS=NB+1+MOD(NB,2)                                                 CONTR 156
      NF=MIN0(NB+10,NBF(1)-5)                                           CONTR 157
      DO 220 K=NS,NF,2                                                  CONTR 158
      DO 230 J=1,NST                                                    CONTR 159
      XS(J)=XBF(J,K)                                                    CONTR 160
      YS(J)=RBF(J,K)                                                    CONTR 161
      IF (NBF(J+1) .LT. K) GO TO 240                                    CONTR 162
  230 CONTINUE                                                          CONTR 163
  240 NN=J                                                              CONTR 164
      XS(NN+1)=ZPLOT                                                    CONTR 165
      YS(NN+1)=YS(NN)+(XS(NN+1)-XS(NN))*(YS(NN)-YS(NN-1))               CONTR 166
     *       /(XS(NN)-XS(NN-1))                                         CONTR 167
      DO 250 N=1,NN                                                     CONTR 168
      CALL DUTLN(XS(N),YS(N),XS(N+1),YS(N+1),20,0)                      CONTR 169
  250 CONTINUE                                                          CONTR 170
  220 CONTINUE                                                          CONTR 171
      RETURN                                                            CONTR 172
C                                                                       CONTR 173
      END                                                               CONTR 174
```

```
      SUBROUTINE CONTUR                                                 CONTUR   2
C                                                                       CONTUR   3
C     SUBROUTINE CONTUR CONTROLS CALCULATING AND PRINTING THE CONTOURS  CONTUR   4
C                    AND CREATING THE PLOTS                             CONTUR   5
C                                                                       CONTUR   6
      COMMON /BIN/ ANGP,ANGN,KBCON,BCON(20)                             BIN      2
      LEVEL 2, BPARA,BPERP,BNORM,BMAG,BANG                              BCOMPS   2
      COMMON /BCOMPS/ BPARA(20,100),BPERP(20,100),BNORM(20,100),        BCOMPS   3
     *  BMAG(20,100),BANG(20,100)                                        BCOMPS   4
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,      CONTUR   9
     *  NZMAX,AMACH,GAM,HRD,NHINDX                                      CONTUR  10
      LEVEL 2,NB,NBF,XBF,RBF,ELBF,GAMBF                                 BVAL     2
      COMMON /BVAL/ NB,NBF(51),XBF(51,100),RBF(51,100),ELBF(51,100),    BVAL     3
     *  GAMBF(51,100)                                                   BVAL     4
      COMMON/CONT/ KVCON,VCON(20),KRCON,KCON(25),XC(20,100),YC(20,100), CONTUR  12
     *  VF(20,100),RHOF(20,100)                                         CONTUR  13
      COMMON/UNSTRM/ ZPLOT,NBLUNT,NZEND,NZADD                           UNSTRM   2
      COMMON/PLOTC/ CONTX(1000),CONTY(1000),CVAL(30),NAD(30),IPLOT      CONTUR  15
      LOGICAL LPLRUN,LPRFL,LPRST,LPRCON,LPRB,LPLOT                      PROPT    2
      COMMON /PROPT/ LRERUN,LPRFL,LPRST,LPRCON,LPRB,LPLOT               PROPT    3
      COMMON /COM2/ ICHK(4,1000),DUM(500)                               CONTUR  17
      COMMON /COM4/ BSCM(20,100),DUM1(100)                              CONTUR  18
C                                                                       CONTUR  19
C     PLOT STREAMLINES                                                  CONTUR  20
C                                                                       CONTUR  21
      CALL SFCALC(XSHK,YSHK,NZMAX)                                      CONTUR  22
      IPLOT=3                                                           CONTUR  23
      IF (LPLOT) CALL PLOTCN                                            CONTUR  24
C                                                                       CONTUR  25
C                                                                       CONTUR  26
C     CALCULATE VELOCITY CONTOUR LINES                                  CONTUR  27
C         (GRID IS NRMAX BY NZMAX)                                      CONTUR  28
C                                                                       CONTUR  29
      IF (KVCON .LE. 0) GO TO 10                                        CONTUR  30
      CALL MAP(VF,20,CONTX,CONTY,KVCON,2,VCON,NAD,1000,30,ICHK,         CONTUR  31
     1         2,1,NRMAX,NZMAX)                                         CONTUR  32
C                                                                       CONTUR  33
C     PLOT VELOCITY AND TEMPERATURE CONTOURS                            CONTUR  34
C                                                                       CONTUR  35
      IPLOT=1                                                           CONTUR  36
      FACT=0.5*(GAM-1.0)*AMACH*AMACH                                    CONTUR  37
      IF (LPRCON) CALL CONOUT(VCON,FACT,NHINDX)                         CONTUR  38
      IF (LPLOT) CALL PLOTCN                                            CONTUR  39
C                                                                       CONTUR  40
C     CALCULATE DENSITY CONTOUR LINES                                   CONTUR  41
C                                                                       CONTUR  42
   10 IF (KRCON .LE. 0) GO TO 20                                        CONTUR  43
      CALL MAP(RHOF,20,CONTX,CONTY,KRCON,2,RCON,NAD,1000,30,ICHK,       CONTUR  44
     1         2,1,NRMAX,NZMAX)                                         CONTUR  45
C                                                                       CONTUR  46
C     PLOT DENSITY CONTOURS                                             CONTUR  47
C                                                                       CONTUR  48
      IPLOT=2                                                           CONTUR  49
      IF (LPRCON) CALL CONOUT(RCON,FACT,NHINDX)                         CONTUR  50
      IF (LPLOT) CALL PLOTCN                                            CONTUR  51
C                                                                       CONTUR  52
C     CALCULATE, PRINT, AND PLOT CONTOUR LINES FOR PARALLEL             CONTUR  53
C     AND PERPENDICULAR MAGNETIC FIELD COMPONENTS                       CONTUR  54
C                                                                       CONTUR  55
   20 IF (KBCON .LE. 0) RETURN                                          CONTUR  56
      DO 100 J=1,NZMAX                                                  CONTUR  57
      DO 100 I=1,NRMAX                                                  CONTUR  58
      BSCM(I,J)=BPARA(I,J)                                              CONTUR  59
  100 CONTINUE                                                          CONTUR  60
      CALL MAP(BSCM,20,CONTX,CONTY,KBCON,2,BCON,NAD,1000,30,ICHK,       CONTUR  61
     1         2,1,NRMAX,NZMAX)                                         CONTUR  62
      IPLOT=5                                                           CONTUR  63
      IF (LPRCON) CALL CONOUT(BCON,FACT,NHINDX)                         CONTUR  64
      IF (LPLOT) CALL PLOTCN                                            CONTUR  65
      DO 110 J=1,NZMAX                                                  CONTUR  66
      DO 110 I=2,NRMAX                                                  CONTUR  67
      BSCM(I,J)=BPERP(I,J)                                              CONTUR  68
  110 CONTINUE                                                          CONTUR  69
      CALL MAP(BSCM,20,CONTX,CONTY,KBCON,2,BCON,NAD,1000,30,ICHK,       CONTUR  70
     1         2,1,NRMAX,NZMAX)                                         CONTUR  71
      IPLOT=6                                                           CONTUR  72
      IF (LPRCON) CALL CONOUT(BCON,FACT,NHINDX)                         CONTUR  73
      IF (LPLOT) CALL PLOTCN                                            CONTUR  74
      RETURN                                                            CONTUR  75
      END                                                               CONTUR  76
```

```
      FUNCTION DRDX(X,R)                                                DRDX     2
C                                                                       DRDX     3
C     THIS ROUTINE DETERMINES THE SLOPE OF THE FLOW ANGLE AT            DRDX     4
C     POINT (X,R) IN THE BLUNT BODY REGION                              DRDX     5
C                                                                       DRDX     6
      COMMON /BLUNT/ THETA(25),RP(20,25)                                DRDX     7
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,      DRDX     8
     1              NTHETA,AMACH,GAM,HRD,NHINDX                        DRDX     9
      COMMON /DRD/ ANG(20,100),DTH(100),DRS(100),R1,JS,DEG              DRDX    10
C                                                                       DRDX    11
C     DETERMINE RP AND THETA FROM R AND X                               DRDX    12
C                                                                       DRDX    13
      THTA=DEG*ATAN2(R,R1-X)                                            DRDX    14
      RPP=SQRT((R1-X)**2+R**2)                                          DRDX    15
C                                                                       DRDX    16
C     USE BIVARIATE LINEAR INTERPOLATION IN THETA THEN RP               DRDX    17
C                                                                       DRDX    18
      DO 3 J=JS,NTHETA                                                  DRDX    19
```

```
      IF (THETA(J) .GT. THTA) GO TO 4                                   DRDX   20
    3 CONTINUE                                                          DRDX   21
      J=NTHETA+1                                                        DRDX   22
    4 JR=J-1                                                            DRDX   23
      IF (JR .LT. JS) JR=JS                                             DRDX   24
      TSLOPE=(THTA-THETA(JR))/DTHE(JR)                                  DRDX   25
      RP2=RP(I,JR)+(RP(I,JR+1)-RP(I,JR))*TSLOPE                         DRDX   26
      DO 5 I=2,NRMAX                                                    DRDX   27
      RP1=RP2                                                           DRDX   28
      RP2=RP(I,JR)+(RP(I,JR+1)-RP(I,JR))*TSLOPE                         DRDX   29
      IF (RP2 .GT. RPP) GO TO 6                                         DRDX   30
    5 CONTINUE                                                          DRDX   31
      I=NRMAX                                                           DRDX   32
    6 CONTINUE                                                          DRDX   33
      DR1=ANG(I-1,JR)+(ANG(I-1,JR+1)-ANG(I-1,JR))*TSLOPE                DRDX   34
      DR2=ANG(I,JR)+(ANG(I,JR+1)-ANG(I,JR))*TSLOPE                      DRDX   35
      DRDX=DR1+(DR2-DR1)*(RPP-RP1)/(RP2-RP1)                            DRDX   36
      RETURN                                                            DRDX   37
      END                                                               DRDX   38


      FUNCTION DRDXM(X,R)                                               DRDXM   2.
C                                                                       DRDXM   3
C     THIS ROUTINE DETERMINES THE SLOPE OF THE FLOW ANGLE AT            DRDXM   4
C     THE DOWNSTREAM POINT (X,R).                                       DRDXM   5
C                                                                       DRDXM   6
      COMMON /BLOUNT/ THETA(25),RP(20,25)                               DRDXM   7
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,      DRDXM   8
     1              NZMAX,AMACH,GAM,RHO,NHINDX                          DRDXM   9
      COMMON/CONT/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100), DRDXM  10
     *  VF(20,100),RHOF(20,100)                                         DRDXM  11
      COMMON /DRD/ ANGM(20,100),DXCM(100),DRCM(100),RP,JX,DEG           DRDXM  12
C                                                                       DRDXM  13
C     USE BIVARIATE LINEAR INTERPOLATION IN XC THEN YC                  DRDXM  14
C                                                                       DRDXM  15
      DO 1 J=JX,NZMAX                                                   DRDXM  16
      IF (XC(1,J) .GT. X) GO TO 2                                       DRDXM  17
    1 CONTINUE                                                          DRDXM  18
      J=NZMAX                                                           DRDXM  19
    2 JR=J-1                                                            DRDXM  20
      IF (JR .LT. JX) JR=JX                                             DRDXM  21
      ZSLOPE=(X-XC(1,JR))/DXCM(JR)                                      DRDXM  22
      R2=YC(1,JR)+(YC(1,JR+1)-YC(1,JR))*ZSLOPE                          DRDXM  23
      DO 3 I=2,NRMAX                                                    DRDXM  24
      R1=R2                                                             DRDXM  25
      R2=YC(I,JR)+(YC(I,JR+1)-YC(I,JR))*ZSLOPE                          DRDXM  26
      IF (R2 .GT. R) GO TO 4                                            DRDXM  27
    3 CONTINUE                                                          DRDXM  28
      I=NRMAX                                                           DRDXM  29
    4 CONTINUE                                                          DRDXM  30
      DR1=ANGM(I-1,JR)+(ANGM(I-1,JR+1)-ANGM(I-1,JR))*ZSLOPE             DRDXM  31
      DR2=ANGM(I,JR)+(ANGM(I,JR+1)-ANGM(I,JR))*ZSLOPE                   DRDXM  32
      DRDXM=DR1+(DR2-DR1)*(R-R1)/(R2-R1)                                DRDXM  33
      RETURN                                                            DRDXM  34
      END                                                               DRDXM  35


      S2=SIN(THET)**2                                                   ELLINF  25
      EM2=AMACH**2                                                      ELLINF  26
      GAM2=(GAM+1.0)*0.5*EM2                                            ELLINF  27
      DD2=1.0-(EM2*S2-1.0)*(GAM*EM2*S2+1.0)/(GAM2*GAM2*S2)              ELLINF  28
      DD=SQRT(DD2)                                                      ELLINF  29
      COTH=1.0/DRSDX(J)                                                 ELLINF  30
      TD=GAM2/(EM2*S2-1.0)-1.0                                          ELLINF  31
      DELT=ATAN(COTH/TD)                                                ELLINF  32
      EL=SQRT(1.0+COTH*COTH*(1.0+DD2)-2.0*DD*(TD+1.0)*COTH*SIN(DELT))   ELLINF  33
      ANG=THET-ASIN(DD*COTH*SIN(THET-DELT)/EL)                          ELLINF  34
   40 ELLINF=EL                                                         ELLINF  35
      GAMEL=ANG                                                         ELLINF  36
      RETURN                                                            ELLINF  37
C                                                                       ELLINF  38
C     IF POINT IS ON AXIS USE LINEAR INTERPOLATION                      ELLINF  39
C                                                                       ELLINF  40
  100 X=XC(1,J)                                                         ELLINF  41
      Y=YC(1,J)                                                         ELLINF  42
      IF (J .GT. 1) GO TO 200                                           ELLINF  43
      N=NBF(1)                                                          ELLINF  44
      DO 110 K=1,N                                                      ELLINF  45
      IF (X .LT. XBF(1,K)) GO TO 120                                    ELLINF  46
  110 CONTINUE                                                          ELLINF  47
  120 ELLINF=ELBF(1,K-1)+(X-XBF(1,K-1))*(ELBF(1,K)-ELBF(1,K-1))         ELLINF  48
     *          /(XBF(1,K)-XBF(1,K-1))                                 ELLINF  49
      GAMEL=ANG                                                         ELLINF  50
      RETURN                                                            ELLINF  51
C                                                                       ELLINF  52
C     IF POINT IS AT X=ZPLOT, USE LINEAR INTERPOLATION                  ELLINF  53
C                                                                       ELLINF  54
  200 IF (J .LT. NZMAX) GO TO 300                                       ELLINF  55
      FUNCTION ELLINF(I,J)                                              ELLINF   2
C                                                                       ELLINF   3
C     THIS FUNCTION INTERPOLATES FOR EL/ELINF AT THE (I,J) GRID POINT   ELLINF   4
C     FROM THE ARRAY ELBF                                               ELLINF   5
C                                                                       ELLINF   6
      LEVEL 2,NB,NBF,XBF,RBF,ELBF,GAMBF                                 BVAL    2
      COMMON /BVAL/ NB,NBF(51),XBF(51,100),RBF(51,100),ELBF(51,100),    BVAL    3
     *  GAMBF(51,100)                                                   BVAL    4
      COMMON /SHOCKS/ DRSDX(100),DST(50)                                SHOCKS  2
      LEVEL 2, XST,PST,NUM,NST                                          STREAM  2
      COMMON /STREAM/ XST(50,150),RST(50,150),NUM(50),NST               STREAM  3
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,      ELLINF  10
     1              NZMAX,AMACH,GAM,RHO,NHINDX                          ELLINF  11
      COMMON/CONT/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100), ELLINF  12
     *  VF(20,100),RHOF(20,100)                                         ELLINF  13
      COMMON/UNSTRM/ ZPLOT,NBLUNT,NZEND,NZADD                           UNSTRM  2
      DIMENSION XQ(4),RQ(4),ELQ(4),GAMQ(4)                              ELLINF  15
C                                                                       ELLINF  16
C     IF POINT IS ON SHOCK, USE FORMULAE AND SAVE RESULTS               ELLINF  17
C                                                                       ELLINF  18
      IF (I .LT. NRMAX) GO TO 100                                       ELLINF  19
      IF (J .GT. 1) GO TO 10                                            ELLINF  20
      EL=ELBF(1,1)                                                      ELLINF  21
      ANG=GAMBF(1,1)                                                    ELLINF  22
      GO TO 40                                                          ELLINF  23
   10 THET=ATAN(DRSDX(J))                                               ELLINF  24
      NIM=NBF(3)+1                                                      ELLINF  56
      DO 210 IM=3,NST                                                   ELLINF  57
      IN=IM+1                                                           ELLINF  58
      N=NBF(IN)+1                                                       ELLINF  59
      IF (Y .LT. RBF(IM,N)) GO TO 220                                   ELLINF  60
      NIM=N                                                             ELLINF  61
  210 CONTINUE                                                          ELLINF  62
      ELLINF=ELBF(IN,N)+(Y-RBF(IN,N))*(EL-ELBF(IN,N))                   ELLINF  63
     *          /(YC(NRMAX,NZMAX)-RBF(IN,N))                           ELLINF  64
      GAMEL=GAMBF(IN,N)+(Y-RBF(IN,N))*(ANG-GAMBF(IN,N))                 ELLINF  65
     *          /(YC(NRMAX,NZMAX)-RBF(IN,N))                           ELLINF  66
      RETURN                                                            ELLINF  67
  220 ELLINF=ELBF(IM,NIM)+(Y-RBF(IM,NIM))*(ELBF(IN,N)-ELBF(IM,NIM))     ELLINF  68
     *          /(RBF(IN,N)-RBF(IM,NIM))                               ELLINF  69
      GAMEL=GAMBF(IM,NIM)+(Y-RBF(IM,NIM))*(GAMBF(IN,N)-GAMBF(IM,NIM))   ELLINF  70
     *          /(RBF(IN,N)-RBF(IM,NIM))                               ELLINF  71
      RETURN                                                            ELLINF  72
C                                                                       ELLINF  73
C     INTERIOR POINT - USE QUADRILATERAL INTERPOLATION                  ELLINF  74
C                                                                       ELLINF  75
  300 N1=NBF(1)                                                         ELLINF  76
      NMAX=NBF(3)+1                                                     ELLINF  77
      DO 320 IST=1,NST                                                  ELLINF  78
      N2=NBF(IST+1)+1                                                   ELLINF  79
      IF (N2 .GT. NMAX) N2=NMAX                                         ELLINF  80
      N=MIN0(N1,N2)                                                     ELLINF  81
      DO 310 JJ=1,N                                                     ELLINF  82
      IF (X .LT. XBF(IST+1,JJ)) GO TO 340                               ELLINF  83
  310 CONTINUE                                                          ELLINF  84
      GO TO 490                                                         ELLINF  85
C                                                                       ELLINF  86
C     FIND QUADRILATERAL WHICH CONTAINS POINT                           ELLINF  87
C                                                                       ELLINF  88
  320 IF (Y .GT. RBF(IST+1,JJ)) GO TO 490                               ELLINF  89
      IF (XBF(IST+1,JJ) .LT. XST(IST,1)) GO TO 370                      ELLINF  90
      SLOPE=(RBF(IST+1,JJ)-Y)/(XBF(IST+1,JJ)-X)                         ELLINF  91
      SLOPE1=(RBF(IST+1,JJ)-RBF(IST+1,JJ-1))                            ELLINF  92
     *          /(XBF(IST+1,JJ)-XBF(IST+1,JJ-1))                       ELLINF  93
      IF (SLOPE1 .GT. SLOPE) GO TO 490                                  ELLINF  94
      IF (JJ .EQ. N) GO TO 420                                          ELLINF  95
      SLOPE2=(RBF(IST+1,JJ)-RBF(IST,JJ))/(XBF(IST+1,JJ)-XBF(IST,JJ))    ELLINF  96
      IF (SLOPE2 .LT. SLOPE) GO TO 350                                  ELLINF  97
  420 CONTINUE                                                          ELLINF  98
      XQ(1)=XBF(IST+1,JJ)                                               ELLINF  99
      RQ(1)=RBF(IST+1,JJ)                                               ELLINF 100
      ELQ(1)=ELBF(IST+1,JJ)                                             ELLINF 101
      GAMQ(1)=GAMBF(IST+1,JJ)                                           ELLINF 102
      IF (XBF(IST+1,JJ-1) .LT. XST(IST,1)) GO TO 390                    ELLINF 103
      XQ(4)=XBF(IST+1,JJ-1)                                             ELLINF 104
      RQ(4)=RBF(IST+1,JJ-1)                                             ELLINF 105
      ELQ(4)=ELBF(IST+1,JJ-1)                                           ELLINF 106
      GAMQ(4)=GAMBF(IST+1,JJ-1)                                         ELLINF 107
      IF (JJ .EQ. N) GO TO 430                                          ELLINF 108
  350 IF (IST .EQ. 1) GO TO 410                                         ELLINF 109
      IF (XBF(IST,JJ-1) .LT. XST(IST-1,1)) GO TO 380                    ELLINF 110
  410 XQ(3)=XBF(IST,JJ-1)                                               ELLINF 111
      RQ(3)=RBF(IST,JJ-1)                                               ELLINF 112
```

```
      XLO(3)=XLBF(IST,JJ-1)                                              ELLINF 113
      GAMO(3)=GAMBF(IST,JJ-1)                                            ELLINF 114
  400 XO(2)=XBF(IST,JJ)                                                  ELLINF 115
      RO(2)=RBF(IST,JJ)                                                  ELLINF 116
      ELO(2)=ELBF(IST,JJ)                                                ELLINF 117
      GAMO(2)=GAMBF(IST,JJ)                                              ELLINF 118
      ELLINF=QUAD(XO,RO,ELO,X,Y)                                         ELLINF 119
      GAMEL=QUAD(XO,RO,GAMO,X,Y)                                         ELLINF 120
      RETURN                                                             ELLINF 121
  350 CONTINUE                                                           ELLINF 122
      IF (SLOPE2 .LT. 0.0) GO TO 420                                     ELLINF 123
      IF (XBF(IST,JJ-1) .GT. X) GO TO 450                                ELLINF 124
      JJ=JJ+1                                                            ELLINF 125
      IF (JJ .LE. N) GO TO 340                                           ELLINF 126
C                                                                        ELLINF 127
C     CONSTRAM BOUNDARY CUTS QUADRILATERAL CONTAINING POINT              ELLINF 128
C                                                                        ELLINF 129
      IF (XBF(IST+1,JJ-1) .LT. XST(IST+1)) GO TO 320                     ELLINF 130
      XO(4)=XBF(IST+1,JJ-1)                                              ELLINF 131
      RO(4)=RBF(IST+1,JJ-1)                                              ELLINF 132
      ELO(4)=ELBF(IST+1,JJ-1)                                            ELLINF 133
      GAMO(4)=GAMBF(IST+1,JJ-1)                                          ELLINF 134
  330 IF (IST .GT. NST) GO TO 470                                        ELLINF 135
      XO(1)=XBF(IST+1,JJ)                                                ELLINF 136
      RO(1)=RBF(IST+1,JJ)                                                ELLINF 137
      ELO(1)=ELBF(IST+1,JJ)                                              ELLINF 138
      GAMO(1)=GAMBF(IST+1,JJ)                                            ELLINF 139
  430 IF (XBF(IST,JJ-1) .LT. XST(IST-1,1)) GO TO 440                     ELLINF 140
      XO(3)=XBF(IST,JJ-1)                                                ELLINF 141
      RO(3)=RBF(IST,JJ-1)                                                ELLINF 142
      ELO(3)=ELBF(IST,JJ-1)                                              ELLINF 143
      GAMO(3)=GAMBF(IST,JJ-1)                                            ELLINF 144
  460 NS=NBF(IST)+1                                                      ELLINF 145
      XO(2)=XBF(IST,NS)                                                  ELLINF 146
      RO(2)=RBF(IST,NS)                                                  ELLINF 147
      ELO(2)=ELBF(IST,NS)                                                ELLINF 148
      GAMO(2)=GAMBF(IST,NS)                                              ELLINF 149
      ELLINF=QUAD(XO,RO,ELO,X,Y)                                         ELLINF 150
      GAMEL=QUAD(XO,RO,GAMO,X,Y)                                         ELLINF 151
      RETURN                                                             ELLINF 152
  320 XO(4)=XSHK(J)                                                      ELLINF 153
      RO(4)=YSHK(J)                                                      ELLINF 154
      ELO(4)=EL                                                          ELLINF 155
      GAMO(4)=ANG                                                        ELLINF 156
      GO TO 330                                                          ELLINF 157
  470 XO(1)=ZPLOT                                                        ELLINF 158
      CALL BSHK1(XO(1),RO(1),GAMO(1),ELO(1))                             ELLINF 159
      GO TO 430                                                          ELLINF 160
  440 XO(3)=XST(IST-1,1)                                                 ELLINF 161
      CALL BSHK1(XO(3),RO(3),GAMO(3),ELO(3))                             ELLINF 162
      GO TO 460                                                          ELLINF 163
C                                                                        ELLINF 164
C     QUADRILATERAL CONTAINING POINT IS REFLEXIVE                        ELLINF 165
C                                                                        ELLINF 166
  450 JJ=JJ-1                                                            ELLINF 167
      SLOPE=(Y-RBF(IST,JJ-1))/(X-XBF(IST,JJ-1))                         ELLINF 168
      SLOPE1=(RBF(IST+1,JJ-1)-RBF(IST,JJ-1))                            ELLINF 169
     *       /(XBF(IST+1,JJ-1)-XBF(IST,JJ-1))                           ELLINF 170
      IF (SLOPE1 .LT. SLOPE) GO TO 450                                   ELLINF 171
      SLOPE2=(RBF(IST,JJ)-RBF(IST,JJ-1))/(XBF(IST,JJ)-XBF(IST,JJ-1))     ELLINF 172
      IF (SLOPE2 .GT. SLOPE) GO TO 450                                   ELLINF 173
      GO TO 420                                                          ELLINF 174
C                                                                        ELLINF 175
C     POINT IS CLOSE TO SHOCK - NEED VALUES ON SHOCK                     ELLINF 176
C                                                                        ELLINF 177
  370 XO(1)=XBF(IST+1,JJ)                                                ELLINF 178
      CALL BSHK1(XO(1),RO(1),GAMO(1),ELO(1))                             ELLINF 179
      SLOPE=(RO(1)-Y)/(XO(1)-X)                                          ELLINF 180
      SLOPE2=(RBF(I-1)-RBF(IST,JJ))/(XBF(I-1)-XBF(IST,JJ))               ELLINF 181
      IF (SLOPE2 .LT. SLOPE) GO TO 350                                   ELLINF 182
  390 XO(4)=XSHK(J)                                                      ELLINF 183
      RO(4)=YSHK(J)                                                      ELLINF 184
      ELO(4)=EL                                                          ELLINF 185
      GAMO(4)=ANG                                                        ELLINF 186
      GO TO 360                                                          ELLINF 187
  350 XO(3)=XST(IST-1,1)                                                 ELLINF 188
      CALL BSHK1(XO(3),RO(3),GAMO(3),ELO(3))                             ELLINF 189
      GO TO 400                                                          ELLINF 190
  490 N1=N2                                                              ELLINF 191
  510 CONTINUE                                                           ELLINF 192
C                                                                        ELLINF 193
C     SLOPEL DETERMINES THE DIRECTION OF EL/ELINF                        ELLINF 194
C                                                                        ELLINF 195
      ENTRY SLOPEL                                                       ELLINF 196
      ELLINF=GAMEL                                                       ELLINF 197
      RETURN                                                             ELLINF 198
      END                                                                ELLINF 199


      SUBROUTINE ENTER(KODE,J,X,NVAL,A1,A2,JMIN,KMIN,ICHK,KOD4,X,Y,NXY,  ENTER   2
     *  ACONT,ISIZE1)                                                    ENTER   3
C                                                                        ENTER   4
C       CONTOUR PROGRAMS MAP, WALK, SEARCH, ENTER, AND CHECK             ENTER   5
C       WRITTEN BY R. E. SORENSON, NASA-AMES RES. CTR., AUG., 1974.      ENTER   6
C         (MODIFIED VERSION)                                             ENTER   7
C                                                                        ENTER   8
C       ASSUMING THAT A POINT ON A CONTOUR LINE HAS BEEN FOUND,          ENTER   9
C       THIS SUBROUTINE RECORDS THAT POINT IN THE BOOKKEEPING ARRAYS.    ENTER  10
C                                                                        ENTER  11
      COMMON/CONT/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100),  ENTER  12
     1             V(20,100),RHO(20,100)                                 ENTER  13
      DIMENSION ICHK(4,1),X(1),Y(1),ACONT(1)                             ENTER  14
C                                                                        ENTER  15
      NXY=NXY+1                                                          ENTER  16
      IF(NXY.GT.ISIZE1) GO TO 1                                          ENTER  17
C                                                                        ENTER  18
      ICHK(1,NXY)=KODE                                                   ENTER  19
      ICHK(2,NXY)=J                                                      ENTER  20
      ICHK(3,NXY)=K                                                      ENTER  21
      ICHK(4,NXY)=NVAL                                                   ENTER  22
C                                                                        ENTER  23
C     IF ENDPOINTS ARE EQUAL, ENTER MIDPOINT                             ENTER  24
C                                                                        ENTER  25
      IF(( A2-A1).EQ.0.0) GO TO 6                                        ENTER  26
      DIF=(ACONT(NVAL)-A1)/(A2-A1)                                       ENTER  27
      GO TO (2,3),KODE                                                   ENTER  28
C     DIF=0.5                                                            ENTER  29
      GO TO (2,3),KODE                                                   ENTER  30
C                                                                        ENTER  31
C       INTERPOLATE FOR CONTOUR POSITION                                 ENTER  32
C                                                                        ENTER  33
    2 X2=XC(J,K+1)                                                       ENTER  34
      Y2=YC(J,K+1)                                                       ENTER  35
      GO TO 4                                                            ENTER  36
    3 X2=XC(J+1,K)                                                       ENTER  37
      Y2=YC(J+1,K)                                                       ENTER  38
    4 X1=XC(J,K)                                                         ENTER  39
      Y1=YC(J,K)                                                         ENTER  40
      X(NXY)=X1+DIF*(X2-X1)                                              ENTER  41
      Y(NXY)=Y1+DIF*(Y2-Y1)                                              ENTER  42
      KOD4=1                                                             ENTER  43
      GO TO 5                                                            ENTER  44
C                                                                        ENTER  45
    1 WRITE(6,101)                                                       ENTER  46
  101 FORMAT(49H CONTOUR SEARCH ABORTED - TABLE OVERFLOW IN (X,Y))       ENTER  47
      KOD4=2                                                             ENTER  48
C                                                                        ENTER  49
    5 CONTINUE                                                           ENTER  50
      RETURN                                                             ENTER  51
      END                                                                ENTER  52


      SUBROUTINE EXTRAP                                                  EXTRAP   2
C                                                                        EXTRAP   3
C     THIS ROUTINE CALCULATES EXTRAPOLATED VALUES OF                     EXTRAP   4
C          RHO AND V/V AT POINTS ALONG THE BOUNDARY THETA=0 USING        EXTRAP   5
C          A LAGRANGIAN INTERPOLATING POLYNOMIAL OVER                    EXTRAP   6
C          THREE UNEQUALLY SPACED POINTS ON EACH RADIAL CURVE.           EXTRAP   7
C                                                                        EXTRAP   8
      COMMON /BLUNT/ THETA(25),RP(2,25)                                  EXTRAP   9
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NKMAX,       EXTRAP  10
     1              NZMAX,AMACH,GAM,RRO,NMINDX                           EXTRAP  11
      COMMON/CONT/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100),  EXTRAP  12
     1             V(20,100),RHO(20,100)                                 EXTRAP  13
      COMMON /DRO/ ANG(20,100),DT(100),DRS(100),RI,JX,DEG                EXTRAP  14
      COMMON /SHOCKS/ DRSDX(100),DST(50)                                 SHOCKS   2
C                                                                        EXTRAP  16
C     CALCULATE LAGRANGIAN COEFFICIENTS                                  EXTRAP  17
C                                                                        EXTRAP  18
      TH23=THETA(2)-THETA(3)                                             EXTRAP  19
      TH24=THETA(2)-THETA(4)                                             EXTRAP  20
      TH34=THETA(3)-THETA(4)                                             EXTRAP  21
      E2=THETA(3)*THETA(4)/(TH23*TH24)                                   EXTRAP  22
      F3=THETA(2)*THETA(4)/(-TH23*TH34)                                  EXTRAP  23
      G4=THETA(2)*THETA(3)/(TH24*TH34)                                   EXTRAP  24
C                                                                        EXTRAP  25
C     CALCULATE XC, RHO,AND V AT THETA=0.                                EXTRAP  26
```

```
      AM2=AMACH**2
      DO 10 I=2,IMAX
      YC(I,1)=0.0
      XC(I,1)=-(.25*PP(I,2)+.5*PP(I,3)+.25*PP(I,4))
      PP(I,1)=-XC(I,1)
      IF (I.LT.NRMAX) GO TO 10
      RHO(I,2)=.25*RHO(I,2)+.5*RHO(I,3)+.25*RHO(I,4)
      V(I,2)=.25*V(I,2)+.5*V(I,3)+.25*V(I,4)
      IF(V(I,2).LT.0.0) V(I,2)=0.
   10 CONTINUE
C
C     CALCULATE EXACT VALUES AT SHOCK WAVE
C
      RHO(NRMAX,1)=(GAM+1.0)*AM2/((GAM-1.0)*AM2+2.0)
      V(NRMAX,1)=1.0/RHO(NRMAX,1)
C
C     EXACT VALUES AT BODY
C
      YC(1,1)=0.0
      XC(1,1)=-1.0
      PP(1,1)=1.0
      V(1,1)=0.0
      PHOF(1,1)=RHO(NRMAX,1)*((GAM+1.0)**2*AM2*0.5/(2.0*GAM*AM2-
     *  (GAM-1.0)))**(1.0/(GAM-1.0))
      DPSDX(1)=99999.0
C
C     DEFINE BOUNDARY ARRAYS FOR LONGPAUSE/PAGNETJPAUSE AND SHOCK
C
      DO 20 J=1,NZMAX
      XBOD(J)=XC(1,J)
      YBOD(J)=YC(1,J)
      XSHK(J)=XC(NRMAX,J)
      YSHK(J)=YC(NRMAX,J)
   20 CONTINUE
C
C     CONVERT ANG ARRAY TO FLOW FIELD ANGLE
C
      DO 30 I=1,NRMAX
      ANG(I,1)=0.0
   30 CONTINUE
      DO 40 J=2,NZMAX
      DO 40 I=1,NRMAX
      ANG(I,J)=ATAN(ANG(I,J))
   40 CONTINUE
      RETURN
      END


      SUBROUTINE FLOUT
C
C     THIS ROUTINE PRINTS THE FLOW FIELD VALUES WHICH WILL BE USED
C        TO CALCULATE THE STREAMLINES AND CONTOURS
C
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,NZMAX,
     *  AMACH,GAM,HRO,NHINDX
      COMMON/DROP/ ANG(20,100),DZ(100),DRS(100),RJ,JS,DEG
      COMMON/CONST/ PVCON,VCON(20),RRCON,RCON(20),XC(20,100),YC(20,100),
     *  VF(20,100),RHOF(20,100)
      COMMON /BCON/ THETA(20),PP(20,20)
      LEVEL 2, VX,VY
      COMMON/VCOMP/ VX(20,100),VY(20,100)
      COMMON/UNSTEM/ ZPLOT,NBLUNT,NZEND,NZADD
C
C     REVERSE SIGN OF XC FOR OUTPUT
C
      JMAX=NBLUNT*NZEND
      DO 2 J=1,JMAX
      DO 2 I=1,NRMAX
    2 XC(I,J)=-XC(I,J)
C
C     PRINT VALUES ALONG SYMMETRY AXIS
C
      WRITE(6,210)
      FACT=0.5*(GAM-1.0)*AMACH*AMACH
      IF(NHINDX.EQ.1) GO TO 10
      WRITE(6,220)
      GO TO 20
   10 WRITE(6,230)
   20 CONTINUE
      DO 30 I=1,NRMAX
      T=1.0+FACT*(1.0-VF(I,1)**2)
      P=RHOF(I,1)*T
      WRITE(6,240) I,XC(I,1),VF(I,1),RHOF(I,1),T,P
   30 CONTINUE
```

```
EXTRAP  27
EXTRAP  28
EXTRAP  29
EXTRAP  30
EXTRAP  31
EXTRAP  32
EXTRAP  33
EXTRAP  34
EXTRAP  35
EXTRAP  36
EXTRAP  37
EXTRAP  38
EXTRAP  39
EXTRAP  40
EXTRAP  41
EXTRAP  42
EXTRAP  43
EXTRAP  44
EXTRAP  45
EXTRAP  46
EXTRAP  47
EXTRAP  48
EXTRAP  49
EXTRAP  50
EXTRAP  51
EXTRAP  52
EXTRAP  53
EXTRAP  54
EXTRAP  55
EXTRAP  56
EXTRAP  57
EXTRAP  58
EXTRAP  59
EXTRAP  60
EXTRAP  61
EXTRAP  62
EXTRAP  63
EXTRAP  64
EXTRAP  65
EXTRAP  66
EXTRAP  67
EXTRAP  68
EXTRAP  69
EXTRAP  70
EXTRAP  71
EXTRAP  72
EXTRAP  73


FLOUT    2
FLOUT    3
FLOUT    4
FLOUT    5
FLOUT    6
FLOUT    7
FLOUT    8
FLOUT    9
TRANSF1  2
TRANSF1  3
TRANSF1  4
TRANSF1  5
TRANSF1  6
DNSTRM   2
FLOUT   12
FLOUT   13
FLOUT   14
FLOUT   15
FLOUT   16
FLOUT   17
FLOUT   18
FLOUT   19
FLOUT   20
FLOUT   21
FLOUT   22
FLOUT   23
FLOUT   24
FLOUT   25
FLOUT   26
FLOUT   27
FLOUT   28
FLOUT   29
FLOUT   30
FLOUT   31
FLOUT   32
FLOUT   33
```

```
C
C     PRINT VALUES OVER BLUNT BODY
C
      WRITE(6,240)
      DO 4 J=2,NBLUNT
      WRITE(6,250) J,RHOF(1,J)
      IF(NHINDX.EQ.0) GO TO 3
      WRITE(6,260)
      GO TO 4
    3 WRITE(6,270)
    4 CONTINUE
      DO 40 I=1,NRMAX
      T=1.0+FACT*(1.0-VF(I,J)**2)
      P=RHOF(I,J)*T
      ALPHA=DEG*ANG(I,J)
      WRITE(6,290) I,XC(I,J),YC(I,J),VX(I,J),VY(I,J),ALPHA,
     *  VF(I,J),RHOF(I,J),T,P
   40 CONTINUE
C
C     PRINT VALUES FOR MARCHING CODE REGION
C
      WRITE(6,280)
      DO 70 JZ=1,NZEND
      J=JZ+NBLUNT
      IF(NHINDX.EQ.1) GO TO 60
      WRITE(6,250) JZ,XC(1,J)
      WRITE(6,310)
      GO TO 65
   60 WRITE(6,311) JZ,XC(1,J)
      WRITE(6,320)
   65 CONTINUE
      DO 70 I=1,NRMAX
      T=1.0+FACT*(1.0-VF(I,J)**2)
      P=RHOF(I,J)*T
      ALPHA=DEG*ANG(I,J)
      WRITE(6,290) I,YC(I,J),VY(I,J),VX(I,J),ALPHA,VF(I,J),RHOF(I,J),T,P
   70 CONTINUE
C
C     RESTORE SIGN OF XC
C
      DO 3 J=NBLUN1,JMAX
      DO 3 I=1,NRMAX
    3 XC(I,J)=-XC(I,J)
      RETURN
C
  200 FORMAT(2X,I3,2X,F8.4,2X,F8.4)
  210 FORMAT(1H1,///51X,26HDETAILED FLOW FIELD OUTPUT//5X,22(1H*)/////
     *       5X,'FLOW FIELD VALUES EXTRAPOLATED TO SYMMETRY AXIS',/
     *       5X,'AT THETA = 0.00 DEGREES')
  220 FORMAT(4X,1HI,7X,3HX/D,5X,3HR/D,5X,6HV/VINF,3X,10HRHO/RHOINF,2X,6HT/TINF,
     *  6X,6HP/PINF)
  230 FORMAT(4X,1HI,7X,4HX/RO,7X,6HV/VINF,3X,10HRHO/RHOINF,5X,6HT/TINF,
     *  6X,6HP/PINF)
  240 FORMAT(1H1,41X,45HFLOW FIELD VALUES FROM BLUNT BODY CALCULATION)
  250 FORMAT(//,14X,'ANGULAR LOCATION NO.',I2,12H, AT THETA =,F8.4,
     *  6H DEGREES)
  260 FORMAT(4X,1HI,7X,4HR/D,5X,3HR/D,5X,3HX/D,7X,7HVX/VINF,5X,
     *  7HVX/VINF,3X,3HFLOW ANGLE,5X,6HV/VINF,3X,10HRHO/RHOINF,5X,
     *  6HT/TINF,6X,6HP/PINF)
  270 FORMAT(4X,1HI,6X,5HR/D2,5X,4HX/RO,6X,4HR/D,6X,7HVX/VINF,5X,
     *  7HVX/VINF,3X,10HFLOW ANGLE,5X,6HV/VINF,3X,10HRHO/RHOINF,5X,
     *  6HT/TINF,6X,6HP/PINF)
  280 FORMAT(1H1,41X,43HFLOW FIELD VALUES FROM MARCHING CALCULATION)
  290 FORMAT(//5CH ADDITIONAL AXIAL LOCATION NO.,I2,1CH, AT X/D =,F8.4/)
  300 FORMAT(4X,1HI,7X,3HR/D,7X,7HVR/VINF,5X,7HVX/VINF,3X,
     *  10HFLOW ANGLE,5X,6HV/VINF,3X,10HRHO/RHOINF,5X,6HT/TINF,6X,
     *  6HP/PINF)
  310 FORMAT(//3CH ADDITIONAL AXIAL LOCATION NO.,I2,1CH, AT X/RO =,
     *  F8.4/)
  320 FORMAT(4X,1HI,7X,4HR/RO,6X,7HVR/VINF,5X,7HVX/VINF,3X,
     *  10HFLOW ANGLE,5X,6HV/VINF,3X,10HRHO/RHOINF,5X,6HT/TINF,6X,
     *  6HP/PINF)
      END


      SUBROUTINE FLOWB
C
C     THIS ROUTINE CONVERTS THE DATA FROM THE BLUNT BODY CALCULATION
C     AND CALCULATES STREAMLINES AROUND THE BLUNT BODY, USING THE
C     COMPUTATIONAL GRID POINTS ON THE SHOCK AS STARTING POINTS
C
      DIMENSION W(20),S(20,6),AR(5),AV(5)
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,
     1              NTHETA,AMACH,GAM,HRO,NHINDX
```

```
      COMMON/CON1/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100),
     *  VF(20,100),RHOF(20,100)                                          TRANSF1  2
      COMMON /BLUNT/ THETA(25),RP(20,25)                               TRANSF1  3
      LEVEL 2, VX,VY                                                   TRANSF1  4
      COMMON/VCOMP/ VX(20,100),VY(20,100)                              TRANSF1  5
      COMMON /ORD/ ANG(20,100),DTH(100),DRS(100),R1,JX,DEG             TRANSF1  6
      COMMON/DNSTRM/ ZPLOT,NBLUNT,NZEND,NZADD                          FLOWB   12
      COMMON/STREAM/ XXX(50,150),RRR(50,150),NUM(50),NST               DNSTRM   2
      LEVEL 2, XXX,RRR,NUM,NST                                         FLOWB   14
      DATA W/20*1.0/                                                   FLOWB   15
C                                                                      FLOWB   16
C     INPUT CYLINDRICAL VELOCITY COMPONENTS AND POLAR GRID POINTS      FLOWB   17
C                                                                      FLOWB   18
      R1=0.                                                            FLOWB   19
      PI=3.1415927                                                     FLOWB   20
      DEG=180./PI                                                      FLOWB   21
      FAC=1./(AMACH*SQRT(GAM))                                         FLOWB   22
      THETA(1)=0.0                                                     FLOWB   23
      NTP=NTHETA+1                                                     FLOWB   24
      DO 2 J=2,NTP                                                     FLOWB   25
      THETA(J)=DEG*THETA(J)                                            FLOWB   26
      DO 1 I=1,NRMAX                                                   FLOWB   27
      VX(I,J)=FAC*VX(I,J)                                              FLOWB   28
      VY(I,J)=FAC*VY(I,J)                                              FLOWB   29
      VF(I,J)=SQRT(VX(I,J)**2+VY(I,J)**2)                              FLOWB   30
      IF(ABS(VX(I,J)).LT.0.0001) GO TO 6                               FLOWB   31
      ANG(I,J)=VY(I,J)/VX(I,J)                                         FLOWB   32
      GO TO 1                                                          FLOWB   33
    6 ANG(I,J)=999.9999                                                FLOWB   34
    1 CONTINUE                                                         FLOWB   35
      DTH(J-1)=THETA(J)-THETA(J-1)                                     FLOWB   36
      DRS(J)=RP(2,J)-RP(1,J)                                           FLOWB   37
    2 CONTINUE                                                         FLOWB   38
C                                                                      FLOWB   39
C     SMOOTH RHO AND V ALONG CONSTANT-THETA LINES.                     FLOWB   40
C                                                                      FLOWB   41
      DO 3 J=2,NTP                                                     FLOWB   42
      CALL FLSOFY(NRMAX,3,RP(1,J),RHOF(1,J),W,25,S,AR,IER)             FLOWB   43
      CALL FLSOFY(NRMAX,3,RP(1,J),VF(1,J),W,25,S,AV,IER)               FLOWB   44
      DO 3 I=1,NRMAX                                                   FLOWB   45
      XRP=RP(I,J)                                                      FLOWB   46
      RHOF(I,J)=((AR(4)*XRP+AR(3))*XRP+AR(2))*XRP+AR(1)                FLOWB   47
      VF(I,J)=((AV(4)*XRP+AV(3))*XRP+AV(2))*XRP+AV(1)                  FLOWB   48
    3 CONTINUE                                                         FLOWB   49
C                                                                      FLOWB   50
C     TRAJECTORY STREAMLINE CALCULATION                                FLOWB   51
C                                                                      FLOWB   52
      DX=AMIN1(DRS(2)/2.,.005)                                         FLOWB   53
      JX=2                                                             FLOWB   54
      NST=NTHETA-1                                                     FLOWB   55
      XF=0.                                                            FLOWB   56
      NLIM=-XC(NRMAX,2)/DX+10.                                         FLOWB   57
      DO 12 JST=1,NST                                                  FLOWB   58
      J=JST+1                                                          FLOWB   59
      RS=YC(NRMAX,J)                                                   FLOWB   60
      XS=XC(NRMAX,J)                                                   FLOWB   61
      DWRX=-XC(NRMAX,2)*.02                                            FLOWB   62
      XWRT=XS+DWRX                                                     FLOWB   63
      IF(JST.EQ.4) DX=AMIN1(DRS(2)/2.,DWRX)                            FLOWB   64
      DO 10 N=1,NLIM                                                   FLOWB   65
      X=XS                                                             FLOWB   66
      R=RS                                                             FLOWB   67
C                                                                      FLOWB   68
C     THIRD-ORDER MODIFIED EULER INTEGRATION PROCEDURE                 FLOWB   69
C                                                                      FLOWB   70
      DRX=ORDX(XS,RS)                                                  FLOWB   71
      RS1=RS+DRX*DX                                                    FLOWB   72
      XS=XS+DX                                                         FLOWB   73
      SLOPE=.5*(DRX+ORDX(XS,RS1))                                      FLOWB   74
      RS=RS+SLOPE*DX                                                   FLOWB   75
      IF(N.EQ.1) GO TO 13                                              FLOWB   76
      IF(XS.GT.XF) GO TO 11                                            FLOWB   77
      IF(XS.LT.XWRT) GO TO 10                                          FLOWB   78
      NN=NN+1                                                          FLOWB   79
      XXX(JST,NN)=XS                                                   FLOWB   80
      RRR(JST,NN)=RS                                                   FLOWB   81
      XWRT=XWRT+DWRX                                                   FLOWB   82
      GO TO 10                                                         FLOWB   83
   13 NN=1                                                             FLOWB   84
      XXX(JST,NN)=X                                                    FLOWB   85
      RRR(JST,NN)=R                                                    FLOWB   86
   10 CONTINUE                                                         FLOWB   87
   11 CONTINUE                                                         FLOWB   88
      NN=NN+1                                                          FLOWB   89
      XXX(JST,NN)=XS                                                   FLOWB   90
      RRR(JST,NN)=RS                                                   FLOWB   91
      NUM(JST)=NN                                                      FLOWB   92
   12 CONTINUE                                                         FLOWB   93
      JX=NTHETA+1                                                      FLOWB   94
      RETURN                                                           FLOWB   95
      END                                                              FLOWB   96
                                                                       FLOWB   97

      SUBROUTINE FLOWM                                                 FLOWM    2
C                                                                      FLOWM    3
C     THIS ROUTINE CONVERTS THE DATA FROM THE DOWNSTREAM MARCHING      FLOWM    4
C     CALCULATION, AND CONTINUES THE CALCULATION OF THE STREAMLINES.   FLOWM    5
C                                                                      FLOWM    6
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,     FLOWM    7
     1              NZMAX,AMACH,GAM,NRO,NHINDX                         FLOWM    8
      COMMON/CON1/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100),TRANSF1  2
     *  VF(20,100),RHOF(20,100)                                        TRANSF1  3
      COMMON /BLUNT/ THETA(25),RP(20,25)                               TRANSF1  4
      LEVEL 2, VX,VY                                                   TRANSF1  5
      COMMON/VCOMP/ VX(20,100),VY(20,100)                              TRANSF1  6
      COMMON /ORD/ ANGM(20,100),DXCM(100),DRCM(100),R1,JX,DEG          FLOWM   10
      COMMON/DNSTRM/ ZPLOT,NBLUNT,NZEND,NZADD                          DNSTRM   2
      LEVEL 2, XXX,RRR,NUM,NST                                         FLOWM   12
      COMMON /STREAM/ XXX(50,150),RRR(50,150),NUM(50),NST              FLOWM   13
C                                                                      FLOWM   14
C     CONVERT FLOW FIELD INFORMATION                                   FLOWM   15
C                                                                      FLOWM   16
      FACT=0.5*AMACH*AMACH*(GAM-1.0)                                   FLOWM   17
      FAC=FACT+1.0                                                     FLOWM   18
      FACM=FAC**(1.0/(GAM-1.))                                         FLOWM   19
      FACV=SQRT(FAC/FACT)                                              FLOWM   20
      NZMAX=NBLUNT+NZEND                                               FLOWM   21
      NR=NRMAX                                                         FLOWM   22
      JZ=1                                                             FLOWM   23
      DO 1 J=JX,NZMAX                                                  FLOWM   24
      DO 4 I=1,NR                                                      FLOWM   25
      VX(I,J)=FACV*VX(I,J)                                             FLOWM   26
      VY(I,J)=FACV*VY(I,J)                                             FLOWM   27
      VF(I,J)=SQRT(VX(I,J)**2+VY(I,J)**2)                              FLOWM   28
      RHOF(I,J)=FACM*RHOF(I,J)                                         FLOWM   29
      IF(ABS(VX(I,J)).LT.0.0001) GO TO 3                               FLOWM   30
      ANGM(I,J)=VY(I,J)/VX(I,J)                                        FLOWM   31
      GO TO 2                                                          FLOWM   32
    3 ANGM(I,J)=999.9999                                               FLOWM   33
    2 CONTINUE                                                         FLOWM   34
    4 CONTINUE                                                         FLOWM   35
      DXCM(J-1)=XC(1,J)-XC(1,J-1)                                      FLOWM   36
      DRCM(J)=YC(2,J)-YC(1,J)                                          FLOWM   37
    1 CONTINUE                                                         FLOWM   38
C                                                                      FLOWM   39
C     CALCULATE ADDITIONAL STARTING POINTS FOR STREAMLINES             FLOWM   40
C                                                                      FLOWM   41
      NZMAX=NBLUNT+NZADD                                               FLOWM   42
      RSB=RRR(NST,1)                                                   FLOWM   43
      KZ=NBLUNT                                                        FLOWM   44
      ADDST=51-NST                                                     FLOWM   45
      RSF=YC(NR,NZMAX)                                                 FLOWM   46
      DRS=AMAX1(RRR(NST,1)-RRR(NST-1,1),(RSF-RRR(NST,1))/ADDST)        FLOWM   47
      NSM1=NST+1                                                       FLOWM   48
      DO 8 K=NSM1,50                                                   FLOWM   49
      RSB=RSB+DRS                                                      FLOWM   50
      IF (RSB.GE.RSF) GO TO 6                                          FLOWM   51
      NST=NST+1                                                        FLOWM   52
      KZZ=KZ                                                           FLOWM   53
      DO 9 JZ=KZZ,NZMAX                                                FLOWM   54
      IF (RSB .GT. YC(NR,JZ)) GO TO 9                                  FLOWM   55
      RRR(NST,1)=RSB                                                   FLOWM   56
      XXX(NST,1)=XC(NR,JZ-1)+(RSB-YC(NR,JZ-1))*(XC(NR,JZ)-XC(NR,JZ-1))FLOWM   57
     *          /(YC(NR,JZ)-YC(NR,JZ-1))                               FLOWM   58
      NUM(NST)=1                                                       FLOWM   59
      KZ=JZ                                                            FLOWM   60
      GO TO 8                                                          FLOWM   61
    9 CONTINUE                                                         FLOWM   62
    8 CONTINUE                                                         FLOWM   63
C                                                                      FLOWM   64
C     STREAMLINE TRAJECTORY CALCULATION                                FLOWM   65
C                                                                      FLOWM   66
      DX=DXCM(JX)                                                      FLOWM   67
      XF=XC(1,NZMAX)                                                   FLOWM   68
      ZPLOT=XF                                                         FLOWM   69
      DWRX=.2                                                          FLOWM   70
      NLIM=XF/DX+10.                                                   FLOWM   71
      DO 12 I=1,NST                                                    FLOWM   72
      NN=NUM(I)                                                        FLOWM   73
```

```
      RS=RRR(I,NN)                                                      FLOWM   74
      XS=XXX(I,NN)                                                      FLOWM   75
      XWRT=XS+DWRX                                                      FLOWM   76
C                                                                       FLOWM   77
C     THIRD ORDER MODIFIED EULER INTEGRATION PROCEDURE                  FLOWM   78
C                                                                       FLOWM   79
      DO 10 N=1,NLIM                                                    FLOWM   80
      DRX=DRDXM(XS,RS)                                                  FLOWM   81
      IF (XS+DX .GE. XF) GO TO 11                                       FLOWM   82
      RS1=RS+DRX*DX                                                     FLOWM   83
      XS=XS+DX                                                          FLOWM   84
      SLOPE=.5*(DRX+DRDXM(XS,RS1))                                      FLOWM   85
      RS=RS+SLOPE*DX                                                    FLOWM   86
      IF(XS.LT.XWRT) GO TO 10                                           FLOWM   87
      NN=NN+1                                                           FLOWM   88
      XXX(I,NN)=XS                                                      FLOWM   89
      RRR(I,NN)=RS                                                      FLOWM   90
      XWRT=XWRT+DWRX                                                    FLOWM   91
   10 CONTINUE                                                          FLOWM   92
   11 CONTINUE                                                          FLOWM   93
      RS1=RS+DRX*(XF-XS)                                                FLOWM   94
      SLOPE=.5*(DRX+DRDXM(XF,RS1))                                      FLOWM   95
      RS=RS+SLOPE*(XF-XS)                                               FLOWM   96
      NN=NN+1                                                           FLOWM   97
      XXX(I,NN)=XF                                                      FLOWM   98
      RRR(I,NN)=RS                                                      FLOWM   99
      NUM(I)=NN                                                         FLOWM  100
   12 CONTINUE                                                          FLOWM  101
      RETURN                                                            FLOWM  102
      END                                                               FLOWM  103


      SUBROUTINE FLSQFY (M,N,X,Y,W,MN1,S1,A,IER)                        FLSQFY   2
C******                                                                 FLSQFY   3
C        PURPOSE                                                        FLSQFY   4
C           FLSQFY CONSTRUCTS A LEAST SQUARES POLYNOMIAL APPROXIMATION  FLSQFY   5
C           OF SPECIFIED DEGREE TO A GIVEN SET OF DATA POINTS WITH      FLSQFY   6
C           GIVEN WEIGHTS USING ORTHOGONAL POLYNOMIALS.                 FLSQFY   7
C        USAGE                                                          FLSQFY   8
C           DIMENSION X(M),Y(M),W(M),S1(MN1,6),A(N+1)                   FLSQFY   9
C           CALL FLSQFY (M,N,X,Y,W,MN1,S1,A,IER)                        FLSQFY  10
C        INPUT PARAMETERS                                               FLSQFY  11
C           M   - NUMBER OF DATA POINTS                                 FLSQFY  12
C           N   - DEGREE OF POLYNOMIAL DESIRED, N.LT.M                  FLSQFY  13
C           X   - ARRAY OF INDEPENDENT VARIABLE                         FLSQFY  14
C           Y   - ARRAY OF DEPENDENT VARIABLE                           FLSQFY  15
C           W   - ARRAY OF POSITIVE WEIGHTS                             FLSQFY  16
C           MN1 - ROW DIMENSION OF SCRATCH ARRAY S1, MN1.GE.M           FLSQFY  17
C           S1  - SCRATCH ARRAY                                         FLSQFY  18
C                 IN THE COLUMN DEFINITIONS BELOW,                      FLSQFY  19
C                    I   REFERS TO POLYNOMIAL ORDER                     FLSQFY  20
C                    J   REFERS TO ROW INDEXING WITHIN A COLUMN         FLSQFY  21
C                 COL 1  POLYNOMIAL P(I-1), VALUE AT EACH X(J) THRU 240 FLSQFY  22
C                        COEFFICIENT OF EACH X**(J-1) TERM AFTER 240    FLSQFY  23
C                 COL 2  POLYNOMIAL P(I), SAME FORM AS COL 1            FLSQFY  24
C                 COL 3  ALPHA(I+1), WHERE I+1=J                        FLSQFY  25
C                 COL 4  BETA(I), WHERE I=J-1                           FLSQFY  26
C                 COL 5  S(I), WHERE I=J-1                              FLSQFY  27
C                 COL 6  SIGMA**2, WHERE I=J-1                          FLSQFY  28
C        OUTPUT PARAMETERS                                              FLSQFY  29
C           A   - ARRAY OF COMPUTED COEFFICIENTS                        FLSQFY  30
C                 A(1) THRU A(N+1) CONTAIN COMPUTED POLYNOMIAL          FLSQFY  31
C                 COEFFICIENTS, IN ORDER OF INCREASING DEGREE.          FLSQFY  32
C           IER - ERROR INDICATOR                                       FLSQFY  33
C                 .EQ.0   SUCCESS                                       FLSQFY  34
C                 .EQ.1   (N.GE.M).OR.(N.LE.0)                          FLSQFY  35
C                 .EQ.2   MN1.LT.M                                      FLSQFY  36
C                 .EQ.3   M.LE.1                                        FLSQFY  37
C                 .EQ.4   W(I).LE.0                                     FLSQFY  38
C        REFERENCE                                                      FLSQFY  39
C           FORSYTHE, G. E., GENERATION AND USE OF ORTHOGONAL           FLSQFY  40
C           POLYNOMIALS FOR DATA-FITTING WITH A DIGITAL COMPUTER,       FLSQFY  41
C           J. SIAM, VOL. 5, NO. 2, (JUNE 1957) PP. 74-88.              FLSQFY  42
C           NOTE - MOST NOTATION AND MOST LOCAL VARIABLE NAMES ARE      FLSQFY  43
C           BASED ON THE REFERENCE, IE WPP IS W(I,I)-(P(I),P(I)),       FLSQFY  44
C           BUT W, WT AND W(I) REFER TO THE WEIGHTS ARRAY.              FLSQFY  45
C******                                                                 FLSQFY  46
      DIMENSION X(1),Y(1),W(1),S1(MN1,1),A(1)                           FLSQFY  47
C                                                                       FLSQFY  48
C  ***  INITIAL SET-UP  ***                                            FLSQFY  49
C       TEST FOR USER ERRORS                                            FLSQFY  50
  100 IER = 4                                                           FLSQFY  51
      IF (N.GE.M).OR.(N.LE.0)) IER=1                                    FLSQFY  52
      IF (MN1.LT.M) IER=2                                               FLSQFY  53
      IF (M.LE.1) IER=3                                                 FLSQFY  54
      IF (IER.NE.4) GO TO 990                                           FLSQFY  55
C         BETA(1)=0, P(0)=1, P(-1)=0, W(0,0)=SUM(W(I))                  FLSQFY  56
C         DELTA**2(-1)=(WF,F)                                           FLSQFY  57
      N1 = N+1                                                          FLSQFY  58
      S1(1,4)=0.                                                        FLSQFY  59
      DSQ=0.                                                            FLSQFY  60
      WPP=0.                                                            FLSQFY  61
      DO 110 J=1,M                                                      FLSQFY  62
      S1(J,2)=1.                                                        FLSQFY  63
      S1(J,1)=0.                                                        FLSQFY  64
      WT = W(J)                                                         FLSQFY  65
      IF (WT.LE.0.) GO TO 990                                           FLSQFY  66
      WPP = WPP+WT                                                      FLSQFY  67
  110 DSQ = DSQ+WT*Y(J)*Y(J)                                            FLSQFY  68
C                                                                       FLSQFY  69
C  ***  GENERATE ORTHOGONAL POLYNOMIALS - THRU 240  ***                 FLSQFY  70
      DO 240 I=1,N1                                                     FLSQFY  71
      WYP=0.                                                            FLSQFY  72
      WXPP=0.                                                           FLSQFY  73
C         COMPUTE (WP(I),XP(I)), AND ONLGA(I)=(F,P(I))                  FLSQFY  74
      DO 200 J=1,M                                                      FLSQFY  75
      TEMP=W(J)*S1(J,2)                                                 FLSQFY  76
      IF (I.LT.N1) WXPP=WXPP+TEMP*X(J)*S1(J,2)                          FLSQFY  77
  200 WYP=WYP+TEMP*Y(J)                                                 FLSQFY  78
C         S(I)=ONLGA(I)/W(I,I)                                          FLSQFY  79
      S1(I,5)=WYP/WPP                                                   FLSQFY  80
C         DELTA**2(I)=DELTA**2(I-1)-S(I)**2W(I,I)                       FLSQFY  81
      DSQ=DSQ-S1(I,5)*S1(I,5)*WPP                                       FLSQFY  82
C         SIGMA**2(I)=DELTA**2(I)/(DEGREES OF FREEDOM)                  FLSQFY  83
C         SIGMA**2(M)=0                                                 FLSQFY  84
      BK = M-I                                                          FLSQFY  85
      IF (BK.NE.0.) BK = DSQ/BK                                         FLSQFY  86
      S1(I,6) = BK                                                      FLSQFY  87
C         THRU 240 - CALCULATIONS FOR NEXT I, SKIP WHEN I=N+1           FLSQFY  88
C         ALPHA(I+1)=(WP(I),XP(I))/(I,I)                                FLSQFY  89
      IF (I.GE.N1) GO TO 240                                            FLSQFY  90
      S1(I,3)=WXPP/WPP                                                  FLSQFY  91
      WPPO=WPP                                                          FLSQFY  92
      WPP=0.                                                            FLSQFY  93
      BT=S1(I,4)                                                        FLSQFY  94
      AL=S1(I,3)                                                        FLSQFY  95
C         P(I+1)=(X-ALPHA(I+1))P(I)-BETA(I)P(I-1)                       FLSQFY  96
C         W(I+1,I+1)=(WP(I+1),P(I+1)), BI(I+1)=W(I+1,I+1)/W(I,I)        FLSQFY  97
      DO 230 J=1,M                                                      FLSQFY  98
      TEMP=(X(J)-AL)*S1(J,2)-BT*S1(J,2)                                 FLSQFY  99
      WPP=WPP+W(J)*TEMP**2                                              FLSQFY 100
      S1(J,1)=S1(J,2)                                                   FLSQFY 101
  230 S1(J,2)=TEMP                                                      FLSQFY 102
      S1(I+1,4)=WPP/WPPO                                                FLSQFY 103
  240 CONTINUE                                                          FLSQFY 104
C                                                                       FLSQFY 105
C  ***  COMPUTE COEFFICIENTS OF LEAST SQUARES POLYNOMIAL               FLSQFY 106
C       A = S(0)P(0)+...+S(N)P(N)  ***                                  FLSQFY 107
C         P(0)=1, P(-1)=0, A(INITIAL)=S(0)P(0)=S(0)                     FLSQFY 108
      DO 300 I8=1,N1                                                    FLSQFY 109
      A(I8) = 0.                                                        FLSQFY 110
      S1(I9,1)=0.                                                       FLSQFY 111
  300 S1(I8,2)=0.                                                       FLSQFY 112
      S1(1,2)=1.                                                        FLSQFY 113
      A(1)=S1(1,5)                                                      FLSQFY 114
C         SUM LOOP THRU 310, I = ORDER OF POLYNOMIAL FORMED             FLSQFY 115
      DO 310 I=1,N                                                      FLSQFY 116
      AL=S1(I,3)                                                        FLSQFY 117
      BT=S1(I,4)                                                        FLSQFY 118
      T2=0.                                                             FLSQFY 119
      I1 = I+1                                                          FLSQFY 120
C         FORM P(I+1)=XP(I)-ALPHA(I+1)P(I)-BETA(I)P(I-1)                FLSQFY 121
C         AND ADD TO POLYNOMIAL SUM IN A                                FLSQFY 122
      DO 310 I8=1,I1                                                    FLSQFY 123
      T1=T2-AL*S1(I8,2)-BT*S1(I8,1)                                     FLSQFY 124
      T2=S1(I8,2)                                                       FLSQFY 125
      S1(I8,1)=S1(I8,2)                                                 FLSQFY 126
      S1(I8,2)=T1                                                       FLSQFY 127
  310 A(I8)=A(I8)+T1*S1(I+1,5)                                          FLSQFY 128
      IER = 0                                                           FLSQFY 129
  990 RETURN                                                            FLSQFY 130
      END                                                               FLSQFY 131


      SUBROUTINE INPUT                                                  INPUT   2
C                                                                       INPUT   3
C     THIS ROUTINE READS ALL DATA REQUIRED FOR ONE CASE,                INPUT   4
C          EXCEPT FLOW FIELD DATA FOR RERUN                             INPUT   5
C                                                                       INPUT   6
      COMMON /BIN/ ANGP,ANGN,KBCON,BCON(20)                             BIN     2
```

```
      COMMON/CON1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT,
     > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NL,NL1,IT,TAU,ITER,ICNT,PTURT,PINF,
     <KINF,QINF,CINF,JCS,TM,CLUS,PT,HORN,RNOS,NCASE,NPUNCH
      COMMON/CON2/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100),
     *  VF(20,100),RHOF(20,100)
      COMMON /BLUNT/ THETA(25),RP(20,25)
      LEVEL 2, VX,VY
      COMMON/VCOMP/ VX(20,100),VY(20,100)
      COMMON/DNSTRM/ ZPLOT,NBLUNT,NZEND,NZADD
      COMMON/JOB/ZLI,CFI,CFE,ZLE,ZTRAN,DZTRAN
      COMMON /NOBOD/ XX(100),RR(100),NBOD
      COMMON /BCOND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,
     *  NTMAX,AMACH,GAMMA,HRO,NHINDX
      LOGICAL LRERUN,LPRFL,LPRST,LPRCON,LPRB,LPLOT
      COMMON /PROPT/ LRERUN,LPRFL,LPRST,LPRCON,LPRB,LPLOT
      DIMENSION TITLE(6)

      READ(5,100) TITLE
      READ(5,110) AMACH,GAMMA,HRO,XCALC,NR,NBLUNT,CN,ITER
      READ(5,120) LRERUN,LPRFL,LPRST,LPRCON,LPRB,LPLOT
      READ(5,130) XPLOT,ANGP,ANGN
      READ(5,140) KVCON
      IF (KVCON .GT. 0)  READ(5,160) (VCON(I),I=1,KVCON)
      READ(5,140) KRCON
      IF (KRCON .GT. 0)  READ(5,160) (RCON(I),I=1,KRCON)
      READ(5,140) KBCON
      IF (KBCON .GT. 0)  READ(5,160) (BCON(I),I=1,KBCON)
      HORN=HRO
      IF (XPLOT .GE. 0.0) XPLOT=-1.
      ZPLOT=-XPLOT
       IF (LRERUN) GO TO 10
      IF (HORN .LT. 9.0) READ(5,150) NBOD,(XX(I),RR(I),I=1,NBOD)
      IF (XCALC .GE. 0.0) XCALC=-1.0
      ZLE=-XCALC
      IF (NR .LE. 0) NR=14
      IF (NBLUNT .LE. 0) NBLUNT=24
      IF (CN .LE. 0.0) CN=3.0
      IF(ITER.LT.1) ITER=300
      NHINDX=0
      IF(HORN.GT.0.0) NHINDX=1
      JMAX=NBLUNT+1
      NTMAX=NBLUNT
      NRMAX=NR
      KMAX=NR
      NZEND=0
      NZADD=1
      XMACH=AMACH
      GAM=GAMMA
C
C     WRITE OUT INPUT AND DEFAULT VALUES TO BE USED
C

   10 CONTINUE
      WRITE(6,200) TITLE
      IF (LRERUN) RETURN
      WRITE(6,220) AMACH,GAMMA
      WRITE(9) NBLUNT,NR,AMACH,GAMMA,HORN
      IF (HORN) 14,16,18
   14 WRITE(6,240) NBOD,(XX(I),RR(I),I=1,NBOD)
      WRITE(9) NBOD,(XX(I),RR(I),I=1,NBOD)
      GO TO 20
   16 WRITE(6,216)
      GO TO 20
   18 WRITE(6,218) HORN
   20 CONTINUE
      WRITE(6,220) NR,NBLUNT,CN,ITER,XCALC
      WRITE(6,230) XPLOT,ANGP,ANGN
      WRITE(6,235) LRERUN,LPRFL,LPRST,LPRCON,LPRB,LPLOT
      WRITE(6,240)
      WRITE(6,242) KVCON
      IF (KVCON .GT. 0)  WRITE(6,250) (VCON(I),I=1,KVCON)
      WRITE(6,244) KRCON
      IF (KRCON .GT. 0)  WRITE(6,250) (RCON(I),I=1,KRCON)
      WRITE(6,246) KBCON
      IF (KBCON .GT. 0)  WRITE(6,250) (BCON(I),I=1,KBCON)
      RETURN
C
  100 FORMAT(6A10)
  110 FORMAT(4F10.0,2I10,F10.0,I10)
  120 FORMAT(6L10)
  130 FORMAT(3F10.0)
  140 FORMAT(I10)
  150 FORMAT(I10/(2F10.0))
  160 FORMAT(8F10.0)
C
  200 FORMAT(1H1//25X,8A10////55X,15HINPUT VARIABLES/55X,15(1H*))
```

```
      210 FORMAT(///20X,21HFREESTREAM MACH NO. =,F6.2
     *       ///20X,21HSPECIFIC HEAT RATIO =,F6.4)
      214 FORMAT(///2CX,46HOBSTACLE GEOMETRY: USER-SUPPLIED COORDINATES =,
     *       14,7H POINTS//33X,3HX/D,11X,3HR/D/
     *       (30X,F8.4,6X,F8.4))
      216 FORMAT(///20X,49HOBSTACLE GEOMETRY: DEFAULT MAGNETOPAUSE COORDINAT,
     *       21HES - EQUATORIAL TRACE)
      218 FORMAT(///2CX,44HOBSTACLE G-OMETRY: DEFAULT IONOPAUSE COORDINATES,
     *       16H FOR H/RO=,F4.2)
      22C FORMAT(///2CX,37HPARAMETERS FOR BLUNT BODY CALCULATION//
     *       25X,28HNO. OF RADIAL MESH POINTS  =,I4/
     *       25X,28HNO. OF ANGULAR MESH POINTS =,I4/
     *       25X,14HCOURANT NUMBER,13X,1H=,F5.2/
     *       25X,17HNO. OF ITERATIONS,10X,1H=,I4/
     *      //2CX,42HTERMINAL DOWNSTREAM LOCATION FOR MARCHING ,
     *       13HCALCULATION =,F6.2)
      230 FORMAT(///2CX,43HTERMINAL DOWNSTREAM LOCATION FOR PLOTTING =,F6.2,
     *       ///20X,48HIN-PLANE DEVIATION OF MAGNETIC FIELD FROM FLOW =,
     *       F7.2,6H DEGREES
     *       ///2CX,48HDEVIATION OF MAGNETIC FIELD FROM PLANE OF FLOW =,
     *       F7.2,6H DEGREES)
      235 FORMAT(///2CX,6HLRERUN =,L2//20X,6HLPRFL  =,L2//20X,6HLPRST  =,L2,
     *       ///20X,6HLPRCON =,L2//20X,6HLPRB   =,L2//20X,6HLPLOT  =,L2)
      240 FORMAT(1H1//45X,42HVALUES SPECIFIED FOR CONTOUR CALCULATION/
     *       45X,40(1H*))
      242 FORMAT(////20X,12,29H CONTOUR LEVELS FOR VELOCITY=)
      244 FORMAT(////20X,12,28H CONTOUR LEVELS FOR DENSITY=)
      246 FORMAT(////20X,12,44H CONTOUR LEVELS FOR MAGNETIC FIELD STRENGTH=)
      250 FORMAT(/(10X,8F10.3))
          END
```

```
          SUBROUTINE LABEL

C         THIS ROUTINE ELIMINATES OVERLAP IN PLOTTING A
C             SET OF CONTOUR LABELS.
C
          COMMON /LABELS/ XLAB(30),YLAB(30),CV(30),NCL,ILB(30),NLAB
          COMMON /SCALE/ XSF,YSF,XMAX,YMAX,XLNGTH,YLNGTH
C
C         SORT THE ARRAY YLAB INTO ASCENDING ORDER.  CHANGE
C             THE ORDER OF XLAB AND CV TO CORRESPOND.
C
          CALL BUBBL
C
C
C         FIND FIRST LABEL FROM SORTED ARRAY THAT IS ABOVE X-AXIS.
C
          DO 5 K=1,NCL
          IF(YLAB(K).LT.0.) GO TO 5
          KMIN=K
          GO TO 10
        5 CONTINUE
C
C         NO LABELS FOUND ABOVE X-AXIS.  BAD SET OF CONTOUR VALUES
C             SPECIFIED.  STOP PROGRAM.
C
          WRITE(6,200)
      200 FORMAT( 1H1,10H**********,10X,48HUNABLE TO LABEL CONTOUR LINES BEC
         1AUSE OF OVERLAP,10X,10H**********)
          STOP
C
       10 ILB(1)=KMIN
          NLAB=1
          IF(NCL.EQ.1) RETURN
C
C         EXAMINE ALL LABELS BEYOND LABEL KMIN FOR OVERLAP.
C
          K=2
          KMINP=KMIN+1
          DO 15 I=KMINP,NCL
          IF(YLAB(I).LT.YLAB(KMIN)+.1/YSF.AND.XLAB(I).LT.XLAB(KMIN)
         1   +.4/XSF) GO TO 15
C
C         SAVE ARRAY INDEX FOR LABEL TO BE PLOTTED.
C
          KMIN=I
          ILB(K)=I
          NLAB=K
          K=K+1
       15 CONTINUE
C
C         INSURE THAT THE LAST LABEL IS PLOTTED.
C
          ILB(NLAB)=NCL
```

```
C
C
      RETURN
      END


      SUBROUTINE MAP  (A,JDIM,X,Y,NLIN,KOD,ACONT,NAD,ISIZ1,ISIZ2,ICHK,
     *                 JMIN,KMIN,JMAX,KMAX)
C
C        CONTOUR PROGRAMS MAP, WALK, SEARCH, ENTER, AND CHECK
C        WRITTEN BY REESE SORENSON, NASA-AMES RES. CTR., AUG., 1974.
C          (MODIFIED VERSION)
C
C        THIS SUBROUTINE AND THOSE WHICH IT CALLS PREPARE DATA
C        FOR CONTOUR PLOTTING.
C
C          CALLING PARAMETERS:
C
C        A       TWO-DIMENSIONAL ARRAY TO BE CONTOUR PLOTTED.
C        JDIM    FIRST-POSITION DIMENSION-SIZE OF A.
C        X,Y     ARRAYS TO CONTAIN CONTOUR LINE DATA UPON RETURN.
C                EACH PAIR OF X(I) AND Y(I) REPRESENTS A POINT ON A
C                CONTOUR LINE.
C        NLIN    IS THE NUMBER OF CONSTANT-A CONTOUR LEVELS.
C        KOD     =1 IF CONTOUR LEVELS ARE TO BE COMPUTED INTERNALLY
C                BY LINEARLY INTERPOLATING BETWEEN THE MAXIMUM AND
C                MINIMUM VALUES OF A.
C                =2 IF CONTOUR LEVELS ARE TO BE GIVEN BY THE USER.
C        ACONT   ARRAY OF CONTOUR LEVELS.  SCRATCH IF KOD=1.
C                IF KOD=2, ACONT MUST BE FILLED WITH MONOTONICALLY
C                INCREASING CONTOUR LEVELS.  SHOULD BE DIMENSIONED
C                AT LEAST NLIN.
C        NAD     IT SHOULD BE RECOGNIZED THAT MANY CONTOUR LINES
C                MIGHT RESULT FROM EACH CONTOUR LEVEL.
C                UPON RETURN, NAD(1) IS THE TOTAL NUMBER OF CONTOUR
C                LINES.  ALL THE LINES ARE RUN-TOGETHER, HEAD TO TAIL
C                IN ARRAYS X AND Y.  THE FIRST LINE STARTS AT X(1)
C                AND Y(1) AND ENDS AT X(I) AND Y(I) FOR I=NAD(2)-1.
C                THE N-TH CONTOUR LINE STARTS AT X(I) AND Y(I) FOR
C                I=NAD(N) AND ENDS AT X(I) AND Y(I) FOR I=NAD(N+1)-1.
C                NAD(N) FOR N.GE.2 IS THE STARTING ADDRESS IN X AND Y
C                OF THE N-TH CONTOUR LINE.
C        ISIZ1   DIMENSION SIZE OF X AND Y.
C        ISIZ2   DIMENSION SIZE OF NAD.  IT IS DIFFICULT TO SAY
C                HOW LARGE X, Y, AND NAD SHOULD BE DIMENSIONED.
C                THIS DEPENDS ON THE DIMENSION SIZE OF ARRAY A,
C                THE VALUE OF NLIN, AND THE DEGREE OF ECCENTRICITY
C                OF THE SURFACE REPRESENTED BY ARRAY A.  ERROR MESSAGES
C                WILL RESULT IF THESE ARRAYS ARE TOO SMALL.
C        ICHK    AN ARRAY TO BE USED AS SCRATCH BY THE CONTOUR PROGRAMS.
C                IT SHOULD HAVE AT LEAST 4*ISIZ1 CELLS.
C        JMIN,KMIN,JMAX,KMAX    IT IS RECOGNIZED THAT ONE MIGHT
C                WISH TO CONTOUR-PLOT AN ARRAY A FOR SUBSCRIPTS
C                STARTING AT SOME VALUE LARGER THAN 1.  THIS SUBROUTINE
C                WILL PROCESS ARRAY A(J,K) FOR JMIN.LE.J.LE.JMAX,
C                AND KMIN.LE.K.LE.KMAX.  JMIN, KMIN, JMAX, AND KMAX
C                ARE THE LIMITS ON THE SUBSCRIPTS.  JMIN AND KMIN MAY
C                BE 1.
C
C
      DIMENSION A(1),X(1),Y(1),ACONT(1),NAD(1),ICHK(4,1)
C
      IDEX(J,K)=J+(K-1)*JDIM
C
C     FIND AMAX AND AMIN
      I=IDEX(JMIN,KMIN)
      AMAX=A(I)
      AMIN=AMAX
      DO 1 K=KMIN,KMAX
      DO 1 J=JMIN,JMAX
      I=IDEX(J,K)
      VAL=A(I)
      IF(VAL-AMAX)2,1,3
3     AMAX=VAL
      GO TO 1
2     IF(VAL-AMIN)4,1,1
4     AMIN=VAL
1     CONTINUE
C
C        FIND ACONT BY LINEARLY INTERPOLATING
      GO TO (5,6),KOD
5     DIF=(AMAX-AMIN)/FLOAT(NLIN+1)
      DO 7 N=1,NLIN
7     ACONT(N)=AMIN+FLOAT(N)*DIF
      NMIN=1
      NMAX=NLIN
      GO TO 8
C
C        CHECK ACONT IF GIVEN BY USER
6     NMIN=0
10    NMIN=NMIN+1
      IF(NMIN.GT.NLIN)  GO TO 9
      IF(AMIN.GT.ACONT(NMIN))  GO TO 10
C
      NMAX=NLIN+1
11    NMAX=NMAX-1
      IF(NMAX.LT.1)  GO TO 9
      IF(AMAX.LT.ACONT(NMAX))  GO TO 11
      IF(NMIN.GT.NMAX)  GO TO 9
C
      IF (NMIN.GT.NMAX) GO TO 12
      NST=NMIN+1
      DO 13 N=NST,NMAX
      IF(ACONT(N).LE.ACONT(N-1))  GO TO 9
13    CONTINUE
12    CONTINUE
C
C        PART 1:  GO AROUND THE BOUNDARY LOOKING FOR LINES WHICH
C        INTERSECT THE BOUNDARY.
      NAD(1)=0
      J=JMIN
      K=KMIN
      NINT=2*(JMAX-JMIN)+2*(KMAX-KMIN)
      KOD7=2
      KOD2=1
      I=IDEX(JMIN,KMIN)
      AA=A(I)
      NXY=0
C
      DO 29 N=1,NINT
      GO TO (21,22,23,24),KOD7
C
C        ORIENTATION 1: UPWARDS
21    J=J-1
      I=IDEX(J,K)
      IF(J.GE.JMIN)  GO TO 25
      J=JMIN
      I=IDEX(J,K+1)
      KOD7=2
      KOD2=1
      GO TO 25
C
C        ORIENTATION 2: TO THE RIGHT
22    K=K+1
      I=IDEX(J,K+1)
      IF(K.LT.KMAX)  GO TO 25
      I=IDEX(J+1,K)
      KOD7=3
      KOD2=2
      GO TO 25
C
C        ORIENTATION 3:  DOWNWARD
23    J=J+1
      I=IDEX(J+1,K)
      IF(J.LT.JMAX)  GO TO 25
      K=KMAX-1
      I=IDEX(J,K)
      KOD7=4
      KOD2=1
      GO TO 25
C
C        ORIENTATION 4:  TO THE LEFT
24    K=K-1
      I=IDEX(J,K)
C
      IF(K.GE.KMIN) GO TO 25
      K=KMIN
      I=IDEX(J,K)
      KOD7=1
      KOD2=2
C        FIND AH AND AL
25    AB=AA
      AA=A(I)
      IF(AB-AA)26,26,27
26    AH=AA
      AL=AB
      GO TO 28
27    AH=AB
      AL=AA
C
C        CHECK TO SEE IF A CONTOUR LINE PASSES THROUGH THE INTERVAL
C        UNDER CONSIDERATION.
```

```
      KODE=1                                                                    MAP   168
      DO 29 NVAL=NMIN,NMAX                                                      MAP   169
      VAL=ACONT(NVAL)                                                           MAP   170
      GO TO (37,39),KOD5                                                        MAP   171
37    IF(VAL.LT.AL)  GO TO 29                                                   MAP   172
      KODE=2                                                                    MAP   173
30    IF(VAL.GT.AH)  GO TO 29                                                   MAP   174
C                                                                               MAP   175
C        CHECK TO SEE IF THE CONTOUR LINE POINT JUST FOUND IS A                 MAP   176
C        NEW ONE, OR THE TAIL END OF ONE ALREADY FOUND.                         MAP   177
      CALL CHECK(ICHK,NXY,KOD2,J,K,NVAL,KOD9)                                   MAP   178
      IF(KOD5.EQ.2)  GO TO 29                                                   MAP   179
C                                                                               MAP   180
C        DETERMINE A1 AND A2                                                    MAP   181
      IF(KOD7.EQ.2.OR.KOD7.EQ.4)  GO TO 32                                      MAP   182
      A2=AA                                                                     MAP   183
      A1=AB                                                                     MAP   184
      GO TO 33                                                                  MAP   185
32    A2=AB                                                                     MAP   186
      A1=AA                                                                     MAP   187
C                                                                               MAP   188
C        ENTER IN THE TABLES THE POINT JUST FOUND.                              MAP   189
33    NDUM=NAD(1)+1                                                             MAP   190
      IF(NDUM.GT.ISIZ2)  GO TO 35                                               MAP   191
      NAD(1)=NDUM                                                               MAP   192
      CALL ENTER(KOD2,J,K,NVAL,A1,A2,JMIN,KMIN,ICHK,KOD4,X,Y,NXY,ACONT, -       MAP   193
     *           ISIZ1)                                                         MAP   194
      IF(KOD4.EQ.2)  GO TO 70                                                   MAP   195
      NAD(NDUM)=NXY                                                             MAP   196
C                                                                               MAP   197
C        NOW FOLLOW UP THIS LEAD,  WALK THE LENGTH OF THE LINE.                 MAP   198
      KODE=KOD7                                                                 MAP   199
      KOD6=2                                                                    MAP   200
      JS=J                                                                      MAP   201
      KS=K                                                                      MAP   202
36    CALL WALK(KODE,JS,KS,A,JDIM,ICHK,JMIN,JMAX,KMIN,KMAX,KOD4,NXY,    -      MAP   203
     *          ACONT,KOD6,X,Y,NVAL,ISIZ1)                                      MAP   204
      IF(KOD4.EQ.2)  GO TO 70                                                   MAP   205
      IF(KOD5.GT.0)  GO TO 36                                                   MAP   206
C                                                                               MAP   207
29    CONTINUE                                                                  MAP   208
28    CONTINUE                                                                  MAP   209
C                                                                               MAP   210
C                                                                               MAP   211
C        PART 2: SCAN THE WHOLE FIELD FOR LINES THAT DON'T INTERSECT            MAP   212
C           THE BOUNDARY                                                        MAP   213
      JI=JMAX-1                                                                 MAP   214
      KI=KMAX-1                                                                 MAP   215
      DO 45 K=KMIN,KI                                                           MAP   216
      I=IDEX(JMIN,K)                                                            MAP   217
      A2=A(I)                                                                   MAP   218
      DO 40 J=JMIN,JI                                                           MAP   219
      A1=A2                                                                     MAP   220
      DO 44 KOD2=1,2                                                            MAP   221
      GO TO (41,42),KOD2                                                        MAP   222
41    IF(J.EQ.JMIN)  GO TO 44                                                   MAP   223
      I=IDEX(J,K+1)                                                             MAP   224
      A2=A(I)                                                                   MAP   225
      GO TO 43                                                                  MAP   226
42    I=IDEX(J+1,K)                                                             MAP   227
      A2=A(I)                                                                   MAP   228
      IF(K.EQ.KMIN)  GO TO 44                                                   MAP   229
43    CONTINUE                                                                  MAP   230
C                                                                               MAP   231
C        FIND AH AND AL                                                         MAP   232
      IF(A1-A2)45,45,46                                                         MAP   233
45    AH=A2                                                                     MAP   234
      AL=A1                                                                     MAP   235
      GO TO 47                                                                  MAP   236
46    AH=A1                                                                     MAP   237
      AL=A2                                                                     MAP   238
C                                                                               MAP   239
C        CHECK TO SEE IF A CONTOUR LINE PASSES THROUGH THE INTERVAL             MAP   240
C           UNDER CONSIDERATION.                                                MAP   241
47    KODE=1                                                                    MAP   242
      DO 48 NVAL=NMIN,NMAX                                                      MAP   243
      VAL=ACONT(NVAL)                                                           MAP   244
      GO TO (49,50),KOD5                                                        MAP   245
49    IF(VAL.LT.AL)  GO TO 48                                                   MAP   246
      KODE=2                                                                    MAP   247
50    IF(VAL.GT.AH)  GO TO 44                                                   MAP   248
C                                                                               MAP   249
C        WE HAVE FOUND A POINT ON A CONTOUR LINE.  CHECK TO SEE IF              MAP   250
C        IT IS A NEW ONE, OR IF WE HAVE ALREADY DONE THIS LINE.                 MAP   251
      CALL CHECK(ICHK,NXY,KOD2,J,K,NVAL,KOD9)                                   MAP   252
      IF(KOD9.EQ.2)  GO TO 48                                                   MAP   253
C                                                                               MAP   254
C        WE HAVE A POINT.  ENTER IT IN THE TABLES.                              MAP   255
      NDUM=NAD(1)+1                                                             MAP   256
      IF(NDUM.GT.ISIZ2)  GO TO 35                                               MAP   257
      NAD(1)=NDUM                                                               MAP   258
      CALL ENTER(KOD2,J,K,NVAL,A1,A2,JMIN,KMIN,ICHK,KOD4,X,Y,NXY,ACONT, -       MAP   259
     *           ISIZ1)                                                         MAP   260
      IF(KOD4.EQ.2)  GO TO 70                                                   MAP   261
      NAD(NDUM)=NXY                                                             MAP   262
C                                                                               MAP   263
C        NOW WALK THE LENGTH OF THE LINE.                                       MAP   264
      KODE=KOD2                                                                 MAP   265
      KOD6=2                                                                    MAP   266
      NXYST=NXY                                                                 MAP   267
      JS=J                                                                      MAP   268
      KS=K                                                                      MAP   269
60    CALL WALK(KODE,JS,KS,A,JDIM,ICHK,JMIN,JMAX,KMIN,KMAX,KOD4,NXY,    -      MAP   270
     *          ACONT,KOD6,X,Y,NVAL,ISIZ1)                                      MAP   271
      IF(KOD4.EQ.2)  GO TO 70                                                   MAP   272
      IF(KOD5.GT.0)  GO TO 96                                                   MAP   273
C                                                                               MAP   274
C        IF THIS LINE IS A CLOSED CURVE, CLOSE IT BY ENTERING FIRST             MAP   275
C           POINT AGAIN.                                                        MAP   276
      DIST=SQRT((X(NXYST)-X(NXY))**2+(Y(NXYST)-Y(NXY))**2)                      MAP   277
      IF(DIST.GT.1.E-42)  GO TO 48                                              MAP   278
      NXY=NXY+1                                                                 MAP   279
      IF(NXY.GT.ISIZ1)  GO TO 60                                                MAP   280
      X(NXY)=X(NXYST)                                                           MAP   281
      Y(NXY)=Y(NXYST)                                                           MAP   282
      ICHK(1,NXY)=ICHK(1,NXYST)                                                 MAP   283
      ICHK(2,NXY)=ICHK(2,NXYST)                                                 MAP   284
      ICHK(3,NXY)=ICHK(3,NXYST)                                                 MAP   285
      ICHK(4,NXY)=ICHK(4,NXYST)                                                 MAP   286
C                                                                               MAP   287
48    CONTINUE                                                                  MAP   288
44    CONTINUE                                                                  MAP   289
40    CONTINUE                                                                  MAP   290
C                                                                               MAP   291
C        ENTER END OF LAST LINE IN BOOKKEEPPING ARRAYS.                         MAP   292
      NXY=NXY+1                                                                 MAP   293
70    NDUM=NAD(1)+1                                                             MAP   294
      IF(NDUM.GT.ISIZ2)  GO TO 35                                               MAP   295
      NAD(NDUM)=NXY                                                             MAP   296
      GO TO 34                                                                  MAP   297
C                                                                               MAP   298
C        WRITE ERROR MESSAGES.                                                  MAP   299
35    WRITE(6,100)                                                              MAP   300
 100  FORMAT(47H1CONTOUR SEARCH ABORTED - TABLE OVERFLOW IN NAD)                MAP   301
      NAD(1)=ISIZ2-1                                                            MAP   302
      GO TO 34                                                                  MAP   303
C                                                                               MAP   304
96    WRITE(6,103)                                                              MAP   305
 103  FORMAT(1H1,1X,5(14H*),2X,20HEXECUTION TERMINATED,2X,5(1H*)///             MAP   306
     *   13X,44HARRAY OF CONTOUR VALUES IMPROPERLY SPECIFIED)                   MAP   307
      STOP                                                                      MAP   308
C                                                                               MAP   309
65    WRITE(6,104)                                                              MAP   310
      GO TO 70                                                                  MAP   311
 104  FORMAT(49H1CONTOUR SEARCH ABORTED - TABLE OVERFLOW IN (X,Y))              MAP   312
C                                                                               MAP   313
34    RETURN                                                                    MAP   314
      END                                                                       MAP   315


      SUBROUTINE PLOTCN                                                         PLOTCN   2
C                                                                               PLOTCN   3
C     THIS ROUTINE CONTROLS THE DRAWING OF THE PLOTS                            PLOTCN   4
C           USE PLOT SUBROUTINES CALLED ARE                                     PLOTCN   5
C              DASH,ENPLT,PLOT.                                                 PLOTCN   6
C                                                                               PLOTCN   7
C     IPLOT  = 1     FOR VELOCITY CONTOUR PLOT                                  PLOTCN   8
C     IPLOT  = 2     FOR DENSITY CONTOUR PLOT                                   PLOTCN   9
C     IPLOT  = 3     FOR STREAMLINE PLOT                                        PLOTCN  10
C                                                                               PLOTCN  11
      LEVEL 2,KST,YST,NUM,NST                                                   PLOTCN  12
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,              PLOTCN  13
     1             NTMAX,AMACH,GAM,RHO,NHINDX                                   PLOTCN  14
      COMMON/PLOTC/ X(1000),Y(1000),CVAL(30),NAD(30),IPLOT                      PLOTCN  15
      COMMON /SCALE/ XSF,YSF,XMAX,YMAX,XLNGTH,YLNGTH                            SCALE   2
      COMMON/STREAM/ XST(50,150),YST(50,150),NUM(50),NST                        PLOTCN  17
C                                                                               PLOTCN  18
C                                                                               PLOTCN  19
C     PLOT STREAMLINES                                                          PLOTCN  20
C           THE FIRST CALL TO PLOT INITIALIZES THE PLOT.                        PLOTCN  21
C           SCALE FACTORS ARE EQUAL TO 1.                                       PLOTCN  22
```

```
C     SUBROUTINE SETUP ESTABLISHES PERMANENT PLOT ORIGIN,                PLOTCN  23
C         DRAWS AXES, LABELS, AND TITLE.                                 PLOTCN  24
C     SUBROUTINE BOUND DRAWS AND LABELS SHOCK WAVE, PLANET,              PLOTCN  25
C          AND MAGNETOSPHERE (IBODY=0) OR IONOSPHERE                     PLOTCN  26
C          (IBODY=1) BOUNDARY.                                           PLOTCN  27
C                                                                        PLOTCN  28
      IF(IPLOT.NE.3) GO TO 10                                            PLOTCN  29
      CALL DASH(4.0,8.0,4.0,8.0,1)                                       PLOTCN  30
      CALL PLOT(0.0,-12.0,-3)                                            PLOTCN  31
      CALL SETUP(IPLOT,AMACH,GAM,NHINDX)                                 PLOTCN  32
      CALL BOUND(XBOD,YBOD,XSHK,YSHK,NTMAX,NHINDX,HRO)                   PLOTCN  33
      CALL CONTR                                                         PLOTCN  34
      CALL ENPLT(XMAX,0.0)                                               PLOTCN  35
      RETURN                                                             PLOTCN  36
C                                                                        PLOTCN  37
C                                                                        PLOTCN  38
C     DRAW CONTOUR PLOTS                                                 PLOTCN  39
C                                                                        PLOTCN  40
   10 CONTINUE                                                           PLOTCN  41
      CALL PLOT(0.0,-12.0,-3)                                            PLOTCN  42
      CALL SETUP(IPLOT,AMACH,GAM,NHINDX)                                 PLOTCN  43
      CALL BOUND(XBOD,YBOD,XSHK,YSHK,NTMAX,NHINDX,HRO)                   PLOTCN  44
      CALL CONTR                                                         PLOTCN  45
      CALL ENPLT(XMAX,0.0)                                               PLOTCN  46
      IF(IPLOT.EQ.1) GO TO 20                                            PLOTCN  47
      RETURN                                                             PLOTCN  48
C                                                                        PLOTCN  49
C     CHANGE VELOCITY CONTOUR VALUES TO TEMPERATURE                      PLOTCN  50
C                                                                        PLOTCN  51
   20 IPLOT=4                                                            PLOTCN  52
      FACT=0.5*(GAM-1.0)*AMACH*AMACH                                     PLOTCN  53
      MVAL=NAU(1)                                                        PLOTCN  54
      DO 25 I=1,MVAL                                                     PLOTCN  55
      CVAL(I)=1.0+FACT*(1.0-CVAL(I))**2)                                 PLOTCN  56
   25 CONTINUE                                                           PLOTCN  57
      GO TO 10                                                           PLOTCN  58
      END                                                                PLOTCN  59
```

```
      FUNCTION QUAD (X,Y,F,XP,YP)                                        QUAD    2
C                                                                        QUAD    3
C     THIS FUNCTION PERFORMS A GENERALIZED QUADRILATERAL INTERPOLATION,   QUAD    4
C     GIVEN THE COORDINATES OF THE VERTICES, THE VALUES OF THE FUNCTION  QUAD    5
C     AT THOSE POINTS, AND THE COORDINATES OF THE POINT AT WHICH A       QUAD    6
C     VALUE OF THE FUNCTION IS DESIRED, (XP,YP)                          QUAD    7
C    X AND Y ARE THE COORDINATE ARRAYS AND F IS THE FUNCTION ARRAY       QUAD    8
C                                                                        QUAD    9
      DIMENSION X(4),Y(4),F(4)                                           QUAD   10
C                                                                        QUAD   11
      S1=4.0*XP-(X(1)+X(2)+X(3)+X(4))                                    QUAD   12
      S2=X(1)+X(2)-X(3)-X(4)                                             QUAD   13
      S3=X(1)-X(2)+X(3)-X(4)                                             QUAD   14
      S4=-X(1)+X(2)+X(3)-X(4)                                            QUAD   15
      T1=4.0*YP-(Y(1)+Y(2)+Y(3)+Y(4))                                    QUAD   16
      T2=Y(1)+Y(2)-Y(3)-Y(4)                                             QUAD   17
      T3=Y(1)-Y(2)+Y(3)-Y(4)                                             QUAD   18
      T4=-Y(1)+Y(2)+Y(3)-Y(4)                                            QUAD   19
      A=S2*T3-S3*T2                                                      QUAD   20
      B=S1*T3+S2*T4-S3*T1-S4*T2                                          QUAD   21
      C=S1*T4-S4*T1                                                      QUAD   22
      D=SQRT(B*B-4.0*A*C)                                                QUAD   23
      ETA=-(B+D)/(A*2.0)                                                 QUAD   24
      IF (ABS(ETA) .GT. 1.0) ETA=ETA+D/A                                 QUAD   25
      XI=(S1+ETA*S2)/(S4+ETA*S3)                                         QUAD   26
      QUAD=0.25*(F(1)*(1.0-ETA)*(1.0-XI)+F(2)*(1.0-ETA)*(1.0+XI)         QUAD   27
     *          +F(3)*(1.0+ETA)*(1.0+XI)+F(4)*(1.0+ETA)*(1.0-XI))       QUAD   28
      RETURN                                                             QUAD   29
      END                                                                QUAD   30
```

```
      SUBROUTINE RERUN                                                   RERUN   2
      COMMON/CONT/ NVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100),  TRANSF1  2
     *  VF(20,100),RHOF(20,100)                                          TRANSF1  3
      COMMON /BLUNT/ THETA(25),RP(20,25)                                 TRANSF1  4
      LEVEL 2, VX,VY                                                     TRANSF1  5
      COMMON/VCOMP/ VX(20,100),VY(20,100)                                TRANSF1  6
      COMMON /BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,      RERUN   4
     *               NZMAX,AMACH,GAM,HRO,NHINDX                        RERUN   5
      COMMON /DNSTRM/ ZPLOT,NBLUNT,NZEND,NZADD                           RERUN   6
      COMMON /NUBOD/ XX(100),RR(100),NBOD                                NUBOD   2
      COMMON /BIN/ ANGP,ANGN,KBCON,BCON(20)                              BIN     2
      LOGICAL LRERUN,LPRFL,LPRST,LPRCON,LPRB,LPLOT                       PROPT   2
      COMMON /PROPT/ LRERUN,LPRFL,LPRST,LPRCUN,LPRB,LPLOT                PROPT   3
      COMMON /SHOCKS/ DRSDX(100),DST(50)                                 SHOCKS  2
```

```
C                                                                        RERUN  11
C     THIS SUBROUTINE READS DATA FROM TAPE4 (WRITTEN TO TAPE5 ON         RERUN  12
C     A PREVIOUS RUN) TO ALLOW RESTARTING OF THE PROGRAM USING           RERUN  13
C     DIFFERENT CONTOUR VALUES, PLOT SIZE, AND/OR MAGNETIC FIELD         RERUN  14
C     ANGLES, WITHOUT REPEATING THE CALCULATION OF THE VELOCITY          RERUN  15
C     AND DENSITY FIELDS.                                                RERUN  16
C                                                                        RERUN  17
      READ(5) NBLUNT,NRMAX,AMACH4,GAM4,HRO4                              RERUN  18
      IF (ABS(AMACH-AMACH4) .GT. 1.0E-2) GO TO 10                        RERUN  19
      IF (ABS(GAM-GAM4) .GT. 1.0E-5) GO TO 100                           RERUN  20
      IF (HRO .LT. 3.7  .AND.  HRO4 .LT. 3.0) GO TO 2                    RERUN  21
      IF (ABS(HRO-HRO4) .GT. 1.0E-5) GO TO 100                           RERUN  22
      NZMAX=NBLUNT                                                       RERUN  23
      GO TO 3                                                            RERUN  24
    2 READ(4) NBOD,(XX(I),RR(I),I=1,NBOD)                                RERUN  25
    3 CONTINUE                                                           RERUN  26
C                                                                        RERUN  27
C     PRINT INPUT DATA                                                   RERUN  28
C                                                                        RERUN  29
      WRITE(6,210) AMACH,GAM                                             RERUN  30
      IF (HRO) 4,6,6                                                     RERUN  31
    4 WRITE(6,214) NBOD,(XX(I),RR(I),I=1,NBOD)                           RERUN  32
      GO TO 9                                                            RERUN  33
    5 WRITE(6,216)                                                       RERUN  34
      GO TO 9                                                            RERUN  35
    6 WRITE(6,218) HRO                                                   RERUN  36
    9 CONTINUE                                                           RERUN  37
      XPLOT=-ZPLOT                                                       RERUN  38
      WRITE(6,230) XPLOT,ANGP,ANGN                                       RERUN  39
      WRITE(6,235) LRERUN,LPRFL,LPRST,LPRCON,LPRB,LPLOT                  RERUN  40
      WRITE(6,240)                                                       RERUN  41
      WRITE(6,242) NVCON                                                 RERUN  42
      IF (NVCON .GT. 0)  WRITE(6,250) (VCON(I),I=1,NVCON)                RERUN  43
      WRITE(6,244) KRCON                                                 RERUN  44
      IF (KRCON .GT. 0)  WRITE(6,250) (RCON(I),I=1,KRCON)                RERUN  45
      WRITE(6,246) KBCON                                                 RERUN  46
      IF (KBCON .GT. 0)  WRITE(6,250) (BCON(I),I=1,KBCON)                RERUN  47
C                                                                        RERUN  48
  210 FORMAT(///2X,21HFREESTREAM MACH NO. =,F6.2                         RERUN  49
     *       ///2X,21HSPECIFIC HEAT RATIO =,F6.3)                       RERUN  50
  214 FORMAT(///2X,45HOBSTACLE GEOMETRY: USER-SUPPLIED COORDINATES =,    RERUN  51
     *       14,7H POINTS//33X,3HX/D,11X,3HR/D/                         RERUN  52
     *       (30X,F6.4,5X,F6.4))                                        RERUN  53
  216 FORMAT(///2X,49HOBSTACLE GEOMETRY: DEFAULT MAGNETOPAUSE COORDINAT, RERUN  54
     *       21HES - EQUATORIAL TRACE)                                  RERUN  55
  218 FORMAT(///2X,46HOBSTACLE GEOMETRY: DEFAULT IONOPAUSE COORDINATES,  RERUN  56
     *       10H FOR H/RO=,F4.2)                                        RERUN  57
  230 FORMAT(///2X,43HTERMINAL DOWNSTREAM LOCATION FOR PLOTTING =,F6.2,  RERUN  58
     *       ///2X,46HIN-PLANE DEVIATION OF MAGNETIC FIELD FROM FLOW =, RERUN  59
     *       F7.2,8H DEGREES                                            RERUN  60
     *       ///2X,43HOUT-OF-PLANE DEVIATION OF MAGNETIC FIELD FROM PLANE OF FLOW =, RERUN  61
     *       F7.2,8H DEGREES)                                           RERUN  62
  235 FORMAT(///2X,6HLRERUN =,L2///2X,6HLPRFL  =,L2///2X,6HLPRST  =,L2, RERUN  63
     *       ///2X,6HLPRCON =,L2///2X,6HLPRB  =,L2///2X,6HLPLOT  =,L2)  RERUN  64
  240 FORMAT(1H1///45X,43HVALUES SPECIFIED FOR CONTOUR CALCULATION/      RERUN  65
     *       45X,43(1H*))                                               RERUN  66
  242 FORMAT(////2X,I2,29H CONTOUR LEVELS FOR VELOCITY:)                 RERUN  67
  244 FORMAT(////2X,I2,29H CONTOUR LEVELS FOR DENSITY:)                  RERUN  68
  246 FORMAT(////2X,I2,44H CONTOUR LEVELS FOR MAGNETIC FIELD STRENGTH:)  RERUN  69
  250 FORMAT(/(1X,6F10.3))                                               RERUN  70
C                                                                        RERUN  71
C                                                                        RERUN  72
C     READ DATA GENERATED BY BLUNT BODY CODE                             RERUN  73
C                                                                        RERUN  74
      DO 10 J=1,NBLUNT                                                   RERUN  75
      JB=J+1                                                             RERUN  76
      READ(4) J1,THETA(JB),DRSDX(JB)                                     RERUN  77
      DO 10 I=1,NRMAX                                                    RERUN  78
      READ(4) I1,XC(I,JB),YC(I,JB),RP(I,JB),VX(I,JB),VY(I,JB),RHOF(I,JB) RERUN  79
   10 CONTINUE                                                           RERUN  80
      CALL FLOWB                                                         RERUN  81
C                                                                        RERUN  82
C     READ DATA GENERATED BY MARCHING CODE                               RERUN  83
C                                                                        RERUN  84
      NZADD=0                                                            RERUN  85
      NZI=NBLUNT+1                                                       RERUN  86
      DO 20 J=NZI,100                                                    RERUN  87
      READ(4) J1,Z,DRSDX(J)                                              RERUN  88
      IF (INTEOF(5)) .NE. 0) GO TO 30                                    RERUN  89
      Z=Z-1.0                                                            RERUN  90
      IF (Z .GT. ZPLOT .AND. NZADD .GT. 0) GO TO 30                      RERUN  91
      NZADD=NZADD+1                                                      RERUN  92
      DO 20 I=1,NRMAX                                                    RERUN  93
      READ(4) I1,YC(I,J),VX(I,J),VY(I,J),RHOF(I,J)                      RERUN  94
      XC(I,J)=Z                                                          RERUN  95
   20 CONTINUE                                                           RERUN  96
```

```
   30 CONTINUE                                                          RERUN   97
      NZEND=NZADD                                                       RERUN   98
      CALL FLOWR                                                        RERUN   99
      RETURN                                                            RERUN  100
C                                                                       RERUN  101
C     PRINT ERROR MESSAGE IF TAPE4 IS NOT THE SAME CASE AS              RERUN  102
C     SPECIFIED IN CARD INPUT  -  PROGRAM IS STOPPED                    RERUN  103
C                                                                       RERUN  104
  100 WRITE(6,1000) AMACH,GAM,HRO,AMACH4,GAM4,HRO4                      RERUN  105
 1000 FORMAT(1H1,19X,5(1H*),2X,20HEXECUTION TERMINATED 2X,5(1H*)//26X,  RERUN  106
     *       34HRUN DATA ON TAPE4 DOES NOT AGREE/2(2X,10HWITH CASE ,    RERUN  107
     *       24HSPECIFIED ON CARD INPUT//2(2X,8HMACH NO.,6X,5HGAMMA,9X, RERUN  108
     *       4HRHO//12X,10HFROM CARD,,3(4X,F10.4)/12X,10HFROM TAPE4,    RERUN  109
     *       3(4X,F10.4))                                               RERUN  110
      STOP                                                              RERUN  111
      END                                                               RERUN  112


      FUNCTION RRINF(I,J)                                               RRINF   2
C                                                                       RRINF   3
C     THIS ROUTINE CALCULATES R/RINF AT THE (I,J) GRID POINT            RRINF   4
C                                                                       RRINF   5
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NKMAX,      RRINF   6
     1             NZMAX,AMACH,GAM,HRO,NKIND                            RRINF   7
      COMMON/CONT/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100),. RRINF   8
     *  VF(20,100),KAUF(20,100)                                        RRINF   9
      LEVEL 2, RST,RST,NUM,NST                                          STREAM   2
      COMMON /STREAM/ RST(50,150),RST(50,150),NUM(50),NST               STREAM   3
C                                                                       RRINF  11
C     IF POINT IS ON SHOCK BOUNDARY R/RINF=1.0                          RRINF  12
C                                                                       RRINF  13
      IF (I .LT. NKMAX  .AND.  J .GT. 1) GO TO 10                       RRINF  14
      RRINF=1.0                                                         RRINF  15
      RETURN                                                            RRINF  16
C                                                                       RRINF  17
C     BRACKET POINT BY TWO STREAMLINES                                  RRINF  18
C                                                                       RRINF  19
                                                                        RRINF  20
                                                                        RRINF  21
   10 CONTINUE                                                          RRINF  22
      X=XC(I,J)                                                         RRINF  23
      Y=YC(I,J)                                                         RRINF  24
      R2=0.0                                                            RRINF  25
      DO 20 JST=1,NST                                                   RRINF  26
      NN=NUM(JST)                                                       RRINF  27
      DO 30 K=1,NN                                                      RRINF  28
      IF (RST(JST,K) .GT. X) GO TO 40                                   RRINF  29
   30 CONTINUE                                                          RRINF  30
      K=NN                                                              RRINF  31
   40 CONTINUE                                                          RRINF  32
      J2=JST                                                            RRINF  33
      J1=JST-1                                                          RRINF  34
      R1=R2                                                             RRINF  35
      IF (K .EQ. 1) R2=RST(J2,1)                                        RRINF  36
      IF (K .GT. 1) R2=RST(J2,K-1)+(X-RST(J2,K-1))*(RST(J2,K)           RRINF  37
     *                -RST(J2,K-1))/(RST(J2,K)-RST(J2,K-1))              RRINF  38
      IF (R2 .GT. Y) GO TO 60                                           RRINF  39
   20 CONTINUE                                                          RRINF  40
   60 IF (J2 .EQ. 1) GO TO 100                                          RRINF  41
C                                                                       RRINF  42
C     INTERPOLATE FOR R/RINF                                            RRINF  43
C                                                                       RRINF  44
      RINF1=RST(J1,1)                                                   RRINF  45
   70 RINF2=RST(J2,1)                                                   RRINF  46
      RRINF=Y/(RINF)*(RINF2-RINF1)*(Y-R1)/(R2-R1))                      RRINF  47
      RETURN                                                            RRINF  48
C                                                                       RRINF  49
C     USE SYMMETRY AXIS AND BODY FOR ZERO-TH STREAMLINE                 RRINF  50
C                                                                       RRINF  51
  100 CONTINUE                                                          RRINF  52
      IF (X .GT. -1.0) GO TO 110                                        RRINF  53
      RRINF=1.0+Y/R2*(R2/RST(1,1)-1.0)                                  RRINF  54
      RETURN                                                            RRINF  55
  110 DO 120 K=2,NZMAX                                                  RRINF  56
      IF (X .LE. XC(1,K)) GO TO 130                                     RRINF  57
  120 CONTINUE                                                          RRINF  58
  130 R1=YC(1,K-1)+(X-XC(1,K-1))*(YC(1,K)-YC(1,K-1))/(XC(1,K)-XC(1,K-1))RRINF  59
      RINF1=0.0                                                         RRINF  60
      GO TO 70                                                          RRINF  61
      END                                                               RRINF  62


      SUBROUTINE SEARCH(J,K,KOD2,A,JDIM,ICHK,NXY,KOD3,NVAL,A1,A2,ACONT) SERCH   2
C                                                                       SERCH   3
C        CONTOUR PROGRAMS MAP, WALK, SERCH, ENTER, AND CHECK            SERCH   4
C        WRITTEN BY REESE SORENSON, NASA-AMES RES. CTR., AUG., 1974.    SERCH   5
C          (MODIFIED VERSION)                                           SERCH   6
C                                                                       SERCH   7
C        THIS SUBROUTINE CHECKS WHETHER A CONTOUR LINE AT LEVEL NVAL    SERCH   8
C        PASSES THROUGH AN INTERVAL HAVING INDICES J,K AT ITS           SERCH   9
C        LEFT(KOD2=2) OR BOTTOM(KOD2=1) POINT.                          SERCH  10
C                                                                       SERCH  11
      DIMENSION A(1),ACONT(1),ICHK(4,1)                                 SERCH  12
C                                                                       SERCH  13
      IDEX(J,K)=J+(K-1)*JDIM                                            SERCH  14
C                                                                       SERCH  15
      GO TO (2,2),KOD2                                                  SERCH  16
    1 I=IDEX(J,K+1)                                                     SERCH  17
      GO TO 3                                                           SERCH  18
    2 I=IDEX(J+1,K)                                                     SERCH  19
    3 A2=A(I)                                                           SERCH  20
      I=IDEX(J,K)                                                       SERCH  21
      A1=A(I)                                                           SERCH  22
C                                                                       SERCH  23
      IF(A1-A2)4,5,5                                                    SERCH  24
    4 AH=A2                                                             SERCH  25
      AL=A1                                                             SERCH  26
      GO TO 6                                                           SERCH  27
    5 AH=A1                                                             SERCH  28
      AL=A2                                                             SERCH  29
C                                                                       SERCH  30
C     VAL=ACONT(NVAL)                                                   SERCH  31
      IF(VAL.LT.AL)  GO TO 12                                           SERCH  32
      IF(VAL.GT.AH)  GO TO 12                                           SERCH  33
C                                                                       SERCH  34
      CALL CHECK(ICHK,NXY,KOD2,J,K,NVAL,KOD9)                           SERCH  35
      IF(KOD9.EQ.2)  GO TO 12                                           SERCH  36
C                                                                       SERCH  37
      KOD3=1                                                            SERCH  38
      GO TO 20                                                          SERCH  39
   12 KOD3=0                                                            SERCH  40
   20 RETURN                                                            SERCH  41
      END                                                               SERCH  42


      SUBROUTINE SETUP(IPLOT,TMACH,GAM,IBODY)                           SETUP   2
C                                                                       SETUP   3
C     THIS ROUTINE ESTABLISHES PLOT ORIGIN, DRAWS                       SETUP   4
C          AND LABELS AXES, AND WRITES TITLE.                           SETUP   5
C          USE PLOT SUBROUTINES USED ARE                                SETUP   6
C               PLOT,AXIS,NUMPLT,CHAR,SCALE,                            SETUP   7
C               CHECK,MATH,PLTLN,DOTLN.                                 SETUP   8
C                                                                       SETUP   9
      DIMENSION TITLE3(2),TITLE4(2)                                     SETUP  10
      DIMENSION TITLE5(2),TITLE6(3),TITLE5(2)                           SETUP  11
      DIMENSION TITLEF(2)                                               SETUP  12
      COMMON /SCALE/ XSF,YSF,XMAX,YMAX,XLNGTH,YLNGTH                    SCALE   2
      DATA PI2/1.5707963/                                               SETUP  14
      DATA TITLE1,TITLE2/9HVELOCITY,,8HDENSITY,/                        SETUP  15
      DATA TITLE3(1),TITLE3(2)/10HSTREAMLINE,1HS/                       SETUP  16
      DATA TITLE4(1),TITLE4(2)/10HTEMPERATUR,1HE/                       SETUP  17
      DATA TITLE5(1),TITLE5(2)/10H(PARALLEL ,10HCOMPONENT)/             SETUP  18
      DATA TITLE6(1),TITLE6(2),TITLE6(3)/10H(PERPENDIC,10HULAR COMPO,   SETUP  19
     1       5HNENT)/                                                   SETUP  20
      DATA TITLE6(1),TITLE6(2)/10HMAGNETIC F,5HIELD,/                   SETUP  21
      DATA TITLEC/9HCONTOURS/,TITLEF(1),TITLEF(2)/10HFIELD LINE,1HS/    SETUP  22
C                                                                       SETUP  23
C                                                                       SETUP  24
C     SET ORIGIN AT LEFT END OF X-AXIS                                  SETUP  25
C                                                                       SETUP  26
      CALL PLOT(1.0,1.0,-3)                                             SETUP  27
      XS=XSF                                                            SETUP  28
      YS=YSF                                                            SETUP  29
      CALL SCALE(XS,YS,1)                                               SETUP  30
C                                                                       SETUP  31
C     DRAW AND LABEL X-AXIS.  NOTE THAT SUBROUTINE AXIS                 SETUP  32
C          REQUIRES PARAMETERS IN INCHES, NOT USER UNITS.               SETUP  33
C                                                                       SETUP  34
      NTC=INT(2.4*XMAX+1.5)+1.0                                         SETUP  35
      CALL AXIS(0.0,0.0,1H ,0,XLNGTH,-NTC,1,0.0)                        SETUP  36
C                                                                       SETUP  37
C     SET PERMANENT ORIGIN AT X=0                                       SETUP  38
      CALL PLOT(1.5,0.0,-3)                                             SETUP  39
      XS=XSF                                                            SETUP  40
      YS=YSF                                                            SETUP  41
      CALL SCALE(XS,YS,1)                                               SETUP  42
C                                                                       SETUP  43
C     DRAW Y-AXIS                                                       SETUP  44
C                                                                       SETUP  45
      NTC=INT(YMAX)+1                                                   SETUP  46
```

```
      CALL AXIS(0.0,0.0,14HU,YLNGTH,-NTC,2,PI2)
C
C     LABEL X-AXIS
C
      YCH=-.4/YSF
      XCH=0.5*XMAX-.75-.21/XSF
      IF (IBODY.EQ.1) GO TO 2
      CALL CHAR(XCH,YCH,0.0,.14,3HX/D,3)
      GO TO 3
    2 CALL CHAR(XCH,YCH,0.0,.14,3HX/R,3)
      XCH=XCH+.42/XSF
      CALL CHAR(XCH,YCH,3.0,.05,1HO,1)
    3 CONTINUE
C
C     ANNOTATE X-AXIS = NOTE ANNOTATION IS POSITIVE LEFT.
C
      YY=-.15/YSF
      IX=0
      XX=-1.5
    5 DELX=FLOAT(IX)*0.5
      XDX=XX+DELX
      CALL NUMPLT(XDX,YY,0.0,-0.1,-XDX,1)
      IX=IX+1
      IF(XDX.LT.XMAX) GO TO 5
C
C     ANNOTATE Y-AXIS AT SIDE EDGE OF PLOT
C
      YY=0.0
   10 YY=YY+1.0
      XCH=-1.5-.25/XSF
      CALL NUMPLT(XCH,YY-.05/YSF,0.0,-0.1,YY,1)
      XTIC=XCH+0.16/XSF
      CALL CHAR(XTIC,YY-.05/YSF,0.0,.09,1H-,1)
      IF(YY.LT.YMAX) GO TO 10
C
C     LABEL Y-AXIS.
C
      XCH=-1.5-0.83/XSF
      YCH=AINT((YMAX-1.0)*0.5)+0.5-.07/YSF
      IF (IBODY.EQ.1) GO TO 12
      CALL CHAR(XCH,YCH,0.0,.14,3HR/D,3)
      GO TO 13
   12 CALL CHAR(XCH,YCH,0.0,.14,3HR/R,3)
      XCH=XCH+.42/XSF
      CALL CHAR(XCH,YCH,0.0,.05,1HO,1)
   13 CONTINUE
C
C     DRAW TITLE.  IPLOT DETERMINES WHICH TITLE
C          SHOULD BE DRAWN.
C
      YCH=YMAX-0.8/YSF
      XCH=-1.5+.4/XSF
      IF(IPLOT.EQ.2) GO TO 15
      IF(IPLOT.EQ.3) GO TO 20
      IF (IPLOT.EQ.4) GO TO 14
C
C     VELOCITY PLOT
      IF (IPLOT .GT. 4) GO TO 21
C
      CALL CHAR(XCH,YCH,0.0,.20,TITLE1,9)
      XCH=XCH+2./XSF
      CALL PLTLN(XCH,YCH,XCH,YCH+.2/YSF)
      CALL CHAR(XCH+.05/XSF,YCH,0.0,.2,1HV,1)
      XCH=XCH+.25/XSF
      CALL PLTLN(XCH,YCH,XCH,YCH+.2/YSF)
      CALL CHAR(XCH+.05/XSF,YCH,0.0,.2,2H/V,2)
      CALL MATH(XCH+.4/XSF,YCH-.05/YSF,.1,0.0,15)
      GO TO 25
C
C     TEMPERATURE PLOT
C
   14 CONTINUE
      CALL CHAR(XCH,YCH,0.0,.20,TITLE4,11)
      XCH=XCH+2.4/XSF
      CALL CHAR(XCH,YCH,0.0,.2,3HT/T,3)
      CALL MATH(XCH+.55/XSF,YCH-.05/YSF,.1,0.0,15)
      GO TO 25
C
C     DENSITY PLOT
C
   15 CONTINUE
      CALL CHAR(XCH,YCH,0.0,.2,TITLE2,8)
      XCH=XCH+2./XSF
      CALL GREEK(XCH,YCH,.2,0.0,17)
      CALL CHAR(XCH+.2/XSF,YCH,0.0,.2,1HP,1)
      CALL GREEK(XCH+.35/XSF,YCH,.2,0.0,17)
      CALL MATH(XCH+.4/XSF,YCH-.05/YSF,.1,0.0,15)
      GO TO 25
C
C     STREAMLINES PLOT
C
   20 CONTINUE
      CALL CHAR(XCH,YCH,0.0,.20,TITLE3,11)
      GO TO 25
C
C     MAGNETIC FIELD PLOTS
C
   21 CONTINUE
      CALL CHAR(XCH,YCH,0.0,0.2,TITL36,15)
      XCH=XCH+3.2/XSF
      CALL CHAR(XCH,YCH,0.0,0.2,4H(B/B,4)
      CALL MATH(XCH+0.75/XSF,YCH-0.03/YSF,0.1/YSF,0.0,15)
      CALL CHAR(XCH+0.8/XSF,YCH,0.0,0.2,1H),1)
      IF (IPLOT .EQ. 6) GO TO 22
      CALL MATH(XCH+1.15/XSF,YCH-0.05/YSF,0.1/YSF,90.0,4)
      XCH=-1.5+0.4/XSF
      YCH=YMAX-1.1/YSF
      CALL CHAR(XCH,YCH,0.0,0.10,TITLE5,23)
      GO TO 23
   22 CALL CHAR(XCH+1.1/XSF,YCH+0.05/YSF,3.1416,0.1,1HT,1)
      XCH=-1.5+.4/XSF
      YCH=YMAX-1.1/YSF
      CALL CHAR(XCH,YCH,0.0,0.10,TITLE6,25)
   23 YCH=YMAX-1.3/YSF
      XCH=-1.5+.6/XSF
      CALL CHAR(XCH,YCH,0.0,0.10,TITLE8,8)
      CALL PLTLN(XCH+1.1/XSF,YCH,XCH+1.5/XSF,YCH)
      YCH=YMAX-1.5/YSF
      CALL CHAR(XCH,YCH,0.0,0.10,TITLE7,1)
      CALL OUTLN(XCH+1.15/XSF,YCH,XCH+1.5/XSF,YCH,20,0)
C
C     DRAW MACH NUMBER (M) AND RATIO OF SPECIFIC
C          HEATS (GAMMA).
C
   25 CONTINUE
      YCH=YMAX-.3/YSF
      XCH=-1.5+.4/XSF
      CALL CHAR(XCH,YCH,0.0,.2,2HM=,2)
      CALL NUMPLT(XCH+.4/XSF,YCH,0.0,.2,TMACH,1)
      XCH=XCH+1.4/XSF
      CALL GREEK(XCH,YCH+.2,0.0,3)
      CALL CHAR(XCH+.2/XSF,YCH,0.0,.2,1H=,1)
      CALL NUMPLT(XCH+.4/XSF,YCH,0.0,.2,GAM,2)
C
      RETURN
      END

      SUBROUTINE SFCALC(XSHK,YSHK,NTMAX)
C
C     THIS ROUTINE DETERMINES SCALE FACTORS AND SIZE OF PLOT.
C
      COMMON /SCALE/ XSF,YSF,XMAX,YMAX,XLNGTH,YLNGTH
      DIMENSION XSHK(1),YSHK(1)
      DATA YSIZE/6.0/
C
C     CALCULATE XMAX AND YMAX BASED ON LAST POINT OF SHOCK WAVE.
C
      XMAX=AINT(XSHK(NTMAX))+0.5
      IF(XMAX.LT.XSHK(NTMAX)) XMAX=XMAX+.5
      YMAX=AINT(YSHK(NTMAX))+1.0
      IF(YMAX.LT.3.0)YMAX=3.0
      IF(XMAX.LT.6.5) XMAX=6.5
C
C     FIX PLOT SIZE IN Y-DIRECTION
C     ADJUST SCALE FACTORS FOR EQUAL X AND Y SCALE FACTORS
C
      YLNGTH=YSIZE
      YSF=YLNGTH/YMAX
      XSF=YSF
      XLNGTH=XSF*(XMAX+1.5)
      RETURN
      END

      SUBROUTINE STOUT
C
C     THIS SUBROUTINE PRINTS THE STREAMLINES CALCULATED
C          IN SUBROUTINES FLOWB AND FLOWM
C
```

```
      LEVEL 2, XST,YST,NUM,NST                                          STOUT    7
      COMMON /BLUNT/ THETA(25),RP(25,25)                                STOUT    8
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,      STOUT    9
     1              NZMAX,AMACH,GAM,HRO,NHINDX                         STOUT   10
      COMMON /STREAM/ XST(50,150),YST(50,150),NUM(50),NST               STOUT   11
C                                                                       STOUT   12
C     REVERSE SIGN OF XST FOR OUTPUT                                    STOUT   13
C                                                                       STOUT   14
      DO 2 K=1,NST                                                      STOUT   15
      NN=NUM(K)                                                         STOUT   16
      DO 2 J=1,NN                                                       STOUT   17
    2 XST(K,J)=-XST(K,J)                                                STOUT   18
C                                                                       STOUT   19
C     STREAMLINES FOR MAGNETOSPHERE                                     STOUT   20
C                                                                       STOUT   21
      WRITE(6,600) NST                                                  STOUT   22
      IF(NHINDX.EQ.1) GO TO 1                                           STOUT   23
      DO 10 K=1,NST                                                     STOUT   24
      WRITE(6,611) K,XST(K,1),YST(K,1)                                  STOUT   25
      IF(XST(K,1).GE.0.0) WRITE(6,621) THETA(K+1),RP(NRMAX,K+1)         STOUT   26
      NN=NUM(K)                                                         STOUT   27
      WRITE(6,630) (XST(K,J),YST(K,J),J=1,NN)                           STOUT   28
   10 CONTINUE                                                          STOUT   29
      GO TO 4                                                           STOUT   30
C                                                                       STOUT   31
C     STREAMLINES FOR IONOPAUSE SHAPE                                   STOUT   32
C                                                                       STOUT   33
    1 DO 11 K=1,NST                                                     STOUT   34
      WRITE(6,611) K,XST(K,1),YST(K,1)                                  STOUT   35
      IF (XST(K,1) .GE. 0.0) WRITE(6,621) THETA(K+1),RP(NRMAX,K+1)      STOUT   36
      NN=NUM(K)                                                         STOUT   37
      WRITE(6,631) (XST(K,J),YST(K,J),J=1,NN)                           STOUT   38
   11 CONTINUE                                                          STOUT   39
C                                                                       STOUT   40
C     RESTORE SIGN OF XST                                               STOUT   41
C                                                                       STOUT   42
    4 CONTINUE                                                          STOUT   43
      DO 3 K=1,NST                                                      STOUT   44
      NN=NUM(K)                                                         STOUT   45
      DO 3 J=1,NN                                                       STOUT   46
    3 XST(K,J)=-XST(K,J)                                                STOUT   47
      RETURN                                                            STOUT   48
C                                                                       STOUT   49
  600 FORMAT(1H1//49X,33HSTREAMLINE TRAJECTORY CALCULATION/49X,33(1H*)//STOUT   50
     *  //10X,I3,23H STREAMLINES CALCULATED)                           STOUT   51
  610 FORMAT(/////15H STREAMLINE NO.,I2,19H, STARTING AT X/D =,F8.4,    STOUT   52
     *  6H,  R/D =,F8.4)                                               STOUT   53
  611 FORMAT(/////15H STREAMLINE NO.,I2,20H, STARTING AT X/RO =,F8.4,   STOUT   54
     *  9H,  R/RO =,F8.4)                                              STOUT   55
  620 FORMAT(13X,23HICORRESPONDS TO THETA =,F6.2,9H DEGREES,,2X,        STOUT   56
     *       6HRP/D =,F8.4,1H))                                        STOUT   57
  621 FORMAT(13X,23HICORRESPONDS TO THETA =,F6.2,9H DEGREES,,2X,        STOUT   58
     *       7HRP/RO =,F8.4,1H))                                       STOUT   59
  630 FORMAT(/15X,3HX/D,12X,3HR/D/(9X,F10.4,5X,F10.4))                  STOUT   60
  631 FORMAT(/15X,4HX/RO,11X,4HR/RO/(9X,F10.4,5X,F10.4))                STOUT   61
      END                                                               STOUT   62

      FUNCTION VINTRP(X,Y)                                              VINTRP   2
C                                                                       VINTRP   3
C     THIS FUNCTION INTERPOLATES FOR V AT (X,Y) FROM THE GRID VALUES    VINTRP   4
C                                                                       VINTRP   5
      COMMON/CON1/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100),TRANSF1   2
     *  VF(20,100),RHOF(20,100)                                         TRANSF1   3
      COMMON /BLUNT/ THETA(25),RP(25,25)                                TRANSF1   4
      LEVEL 2, VX,VY                                                    TRANSF1   5
      COMMON/VCOMP/ VX(20,100),VY(20,100)                               TRANSF1   6
      COMMON/BOUND/ XBOD(100),YBOD(100),XSHK(100),YSHK(100),NRMAX,NZMAX,VINTRP   7
     1  AMACH,GAMMA,HRO,NHINDX                                          VINTRP   8
      COMMON /DNSTRM/ ZPLOT,NBLUNT,NZEND,NZADD                          VINTRP   9
      DATA DEG/57.29578/                                                VINTRP  10
      IF(X.GT.0.0) GO TO 100                                            VINTRP  11
      THET=ATAN(-Y/X)*DEG                                               VINTRP  12
      DO 10 N1=2,NBLUNT                                                 VINTRP  13
      IF(THET.LT.THETA(N1)) GO TO 20                                    VINTRP  14
   10 CONTINUE                                                          VINTRP  15
   20 R=SQRT(X*X+Y*Y)                                                   VINTRP  16
      DTHET=(THETA(N1)-THET)/(THETA(N1)-THETA(N1-1))                    VINTRP  17
      RP1=DTHET*RP(1,N1-1)+(1.0-DTHET)*RP(1,N1)                         VINTRP  18
      DO 30 NR=2,NRMAX                                                  VINTRP  19
      RP2=DTHET*RP(NR,N1-1)+(1.0-DTHET)*RP(NR,N1)                       VINTRP  20
      IF(R.LT.RP2) GO TO 40                                             VINTRP  21
      RP1=RP2                                                           VINTRP  22
   30 CONTINUE                                                          VINTRP  23
   40 DR=(RP2-R)/(RP2-RP1)                                              VINTRP  24
      VINTRP=(DTHET*VF(NR-1,N1-1)+(1.0-DTHET)*VF(NR-1,N1))*DR           VINTRP  25
     *  +(DTHET*VF(NR,N1-1)+(1.0-DTHET)*VF(NR,N1))*(1.0-DR)             VINTRP  26
      RETURN                                                            VINTRP  27
  100 R=Y                                                               VINTRP  28
      DO 110 N1=NBLUNT,NZMAX                                            VINTRP  29
      IF(X.LT.XC(1,N1)) GO TO 120                                       VINTRP  30
  110 CONTINUE                                                          VINTRP  31
      N1=NZMAX                                                          VINTRP  32
  120 DTHET=(XC(1,N1)-X)/(XC(1,N1)-XC(1,N1-1))                          VINTRP  33
      RP1=DTHET*YC(1,N1-1)+(1.0-DTHET)*YC(1,N1)                         VINTRP  34
      DO 130 NR=2,NRMAX                                                 VINTRP  35
      RP2=DTHET*YC(NR,N1-1)+(1.0-DTHET)*YC(NR,N1)                       VINTRP  36
      IF(R.LT.RP2) GO TO 40                                             VINTRP  37
      RP1=RP2                                                           VINTRP  38
  130 CONTINUE                                                          VINTRP  39
      RETURN                                                            VINTRP  40
      END                                                               VINTRP  41

      SUBROUTINE WALK(KOD,J,K,A,JDIM,ICHK,JMIN,JMAX,KMIN,KMAX,KOD4,     -  WALK    2
     *                NXY,ACONT,KOD5,X,Y,NVAL,IS1Z1)                    WALK    3
C                                                                       WALK    4
C        CONTOUR PROGRAMS MAP, WALK, SEARCH, SNTER, AND CHECK           WALK    5
C        WRITTEN BY REESE SORENSON, NASA-AMES RES. CTR., AUG., 1974.    WALK    6
C          (MODIFIED VERSION)                                           WALK    7
C                                                                       WALK    8
C        GIVEN ONE POINT ON A LINE, THIS SUBROUTINE WALKS AROUND        WALK    9
C        THE REST OF THE LINE, RECORDING THE REST OF THE LINE.          WALK   10
C                                                                       WALK   11
      DIMENSION A(1),ICHK(4,1),ACONT(1)                                 WALK   12
      DIMENSION K3(3),A2(3),A2P(3),JJ(3),KK(3),K2(3)                    WALK   13
C                                                                       WALK   14
      GO TO (10,11,12,13),KOD                                           WALK   15
C                                                                       WALK   16
C        ORIENTATION 1: UPWARD                                          WALK   17
   10 JJ(1)=J                                                           WALK   18
      JJ(2)=J                                                           WALK   19
      JJ(3)=J+1                                                         WALK   20
      KK(1)=K                                                           WALK   21
      KK(2)=K+1                                                         WALK   22
      KK(3)=K                                                           WALK   23
      GO TO 15                                                          WALK   24
C                                                                       WALK   25
C        ORIENTATION 2: TO THE RIGHT                                    WALK   26
   11 JJ(1)=J                                                           WALK   27
      JJ(2)=J+1                                                         WALK   28
      JJ(3)=J                                                           WALK   29
      KK(1)=K+1                                                         WALK   30
      KK(2)=K                                                           WALK   31
      KK(3)=K                                                           WALK   32
      GO TO 15                                                          WALK   33
C                                                                       WALK   34
C        ORIENTATION 3: DOWNWARD                                        WALK   35
   12 JJ(1)=J+1                                                         WALK   36
      JJ(2)=J                                                           WALK   37
      JJ(3)=J                                                           WALK   38
      KK(1)=K-1                                                         WALK   39
      KK(2)=K-1                                                         WALK   40
      KK(3)=K-1                                                         WALK   41
   15 K2(1)=1                                                           WALK   42
      K2(2)=2                                                           WALK   43
      K2(3)=1                                                           WALK   44
      GO TO 14                                                          WALK   45
C                                                                       WALK   46
C        ORIENTATION 4:  TO THE LEFT                                    WALK   47
   13 JJ(1)=J-1                                                         WALK   48
      JJ(2)=J-1                                                         WALK   49
      JJ(3)=J-1                                                         WALK   50
      KK(1)=K                                                           WALK   51
      KK(2)=K                                                           WALK   52
      KK(3)=K+1                                                         WALK   53
   14 K2(1)=2                                                           WALK   54
      K2(2)=1                                                           WALK   55
      K2(3)=2                                                           WALK   56
C                                                                       WALK   57
C        SEARCH THE 3 POSSIBLE DIRECTIONS.                              WALK   58
C                                                                       WALK   59
      DO 17 N=1,3                                                       WALK   60
   17 CALL SERCH(JJ(N),KK(N),K2(N),A,JDIM,ICHK,NXY,K3(N),NVAL,A1(N),    -  WALK   61
     *           A2(N),ACONT)                                           WALK   62
C                                                                       WALK   63
      K3S=K3(1)+K3(2)+K3(3)                                             WALK   64
      IF(K3S-111),2,18                                                  WALK   65
C                                                                       WALK   66
C        BRANCH POINT                                                   WALK   67
```

```
18    IF(K3(KOD6).NE.1)  GO TO 9                                        WALK    67
      K6=KOD6                                                           WALK    66
      GO TO 4                                                           WALK    69
C                                                                       WALK    70
C        ONE IN, ONE OUT.  FIND WHERE OUT.                              WALK    71
2     DO 3 K6=1,3                                                       WALK    72
      IF(K3(K6).EQ.1)  GO TO 4                                          WALK    73
3     CONTINUE                                                          WALK    74
C                                                                       WALK    75
C        RECORD THE POINT                                               WALK    76
4     CALL ENTER(K2(K5),JJ(K6),KK(K6),NVAL,A1(K6),A2(K5),JM1N,KM1N,ICHK,-  WALK    77
     *           KOD5,X,Y,NAX,ACONT,IS121)                              WALK    78
C                                                                       WALK    79
C        RESET J AND K                                                  WALK    80
      J=JJ(K6)                                                          WALK    81
      K=KK(K6)                                                          WALK    82
C                                                                       WALK    83
C        SEE IF WE'RE AT A BOUNDARY.  IF SO, QUIT.                      WALK    84
      IF(J.GT.JMIN)  GO TO 23                                           WALK    85
      GO TO (26,19,24,25),KOD5                                          WALK    86
23    IF(J.LT.JMAX)  GO TO 19                                           WALK    87
      GO TO (24,25,26,19),KOD5                                          WALK    88
24    IF(K6.EQ.3)  GO TO 1                                              WALK    89
      GO TO 19                                                          WALK    90
25    IF(K6.EQ.2)  GO TO 1                                              WALK    91
      GO TO 19                                                          WALK    92
26    IF(K6.EQ.1)  GO TO 1                                              WALK    93
C                                                                       WALK    94
19    IF(K.GT.KM1N)  GO TO 27                                           WALK    95
      GO TO (6,28,29,28),KOD5                                           WALK    96
27    IF(K.LT.KMAX)  GO TO 8                                            WALK    97
      GO TO (29,29,8,30),KOD5                                           WALK    98
28    IF(K6.EQ.1)  GO TO 1                                              WALK    99
      GO TO 8                                                           WALK   100
29    IF(K6.EQ.2)  GO TO 1                                              WALK   101
      GO TO 8                                                           WALK   102
30    IF(K6.EQ.3)  GO TO 1                                              WALK   103
C                                                                       WALK   104
C        PREPARE FOR NEXT STEP                                          WALK   105
8     KOD6=4-K6                                                         WALK   106
      KOD5=KOD5+K6-2                                                    WALK   107
      IF(KOD5.GT.4)  KOD5=KOD5-4                                        WALK   108
      IF(KOD5.LT.1)  KOD5=KOD5+4                                        WALK   109
      GO TO 31                                                          WALK   110
C                                                                       WALK   111
9     CONTINUE                                                          WALK   112
1     KOD5=0                                                            WALK   113
31    RETURN                                                            WALK   114
      END                                                               WALK   115
```

```
      SUBROUTINE BLUNT8                                                 BLUNT8   2
      COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT,  COM1    2
     > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTURT,PINF,    COM1    3
     <RINF,QINF,CINF,JCS,TM,CLUS,PT,HORN,RNOSE,NCASE,NPUNCH             COM1    4
      COMMON /COM2/ X(25,20),Y(25,20),XET(25,20,2),XEX(25,20,2),        COM2    2
     *  XEY(25,20,2),D(25,20)                                          COM2    3
      COMMON /COM3/ Q(25,20,4),EF(25,4),S(25,20,4),G(4),AB(4,4)         COM3    2
      COMMON/CONT/ KVCON,VCON(20),KRCUN,RCON(20),XC(20,100),YC(20,100),  TRANSF1  2
     *  VF(20,100),RHOF(20,100)                                        TRANSF1  3
      COMMON /BLUNT/ THETA(25),RP(20,25)                                TRANSF1  4
C     LEVEL 2, VX,VY                                                    TRANSF1  5
      COMMON/VCOMP/ VX(20,100),VY(20,100)                               TRANSF1  6
      COMMON/NINETY/ AMACH,GAMF,KM2,P1INF,R1INF,V1INF,P90(20,3),        TRANSF2  2
     * RHO90(20,3),U90(20,3),V90(20,3),W90(20,3),RS90(3),              TRANSF2  3
     * RSZ90(3),RSPH90(3),HRU,R90                                      TRANSF2  4
      COMMON /SHOCKS/ DRSDX(100),DST(50)                                SHOCKS   2
C...INITIALIZE                                                          BLUNT8   9
      WRITE(6,200)                                                      BLUNT8  10
  200 FORMAT(1H1//53X,22HBLUNT BODY CALCULATION/53X,22(1H*)//)          BLUNT8  11
      CALL INITIA                                                       BLUNT8  12
C...DETERMINE STEPSIZE                                                  BLUNT8  13
      CALL EIGEN                                                        BLUNT8  14
C...INTEGRATE EQUATIONS                                                 BLUNT8  15
      DO 1 I=1,ITER                                                     BLUNT8  16
      IT=IT+1                                                           BLUNT8  17
      CALL SHOCK                                                        BLUNT8  18
      CALL BNDRY                                                        BLUNT8  19
      CALL INTEGR                                                       BLUNT8  20
      CALL XIETAD(1)                                                    BLUNT8  21
    1 CONTINUE                                                          BLUNT8  22
C...PRINT DATA                                                          BLUNT8  23
      CALL OUTPUT(1)                                                    BLUNT8  24
C                                                                       BLUNT8  25
```

```
C.....DATA FOR CONTOURS, ALSO WRITTEN TO TAPE9 FOR RERUN                BLUNT8  26
C                                                                       BLUNT8  27
      DO 5 J=2,JMAX                                                     BLUNT8  28
      THETA(J)=ATAN2(Y(J,1),(1.0-X(J,1)))                               BLUNT8  29
      DRSDX(J)=-XEX(J,KMAX,2)/XEY(J,KMAX,2)                             BLUNT8  30
      WRITE(9) J,THETA(J),DRSDX(J)                                      BLUNT8  31
      DO 5 K=1,KMAX                                                     BLUNT8  32
      XC(K,J)=X(J,K)-1.0                                                BLUNT8  33
      YC(K,J)=Y(J,K)                                                    BLUNT8  34
      RP(K,J)=SQRT(XC(K,J)**2+YC(K,J)**2)                               BLUNT8  35
      VX(K,J)=Q(J,K,2)/Q(J,K,1)                                         BLUNT8  36
      VY(K,J)=Q(J,K,3)/Q(J,K,1)                                         BLUNT8  37
      RHOF(K,J)=Q(J,K,1)*D(J,K)                                         BLUNT8  38
      WRITE(9) K,XC(K,J),YC(K,J),RP(K,J),VX(K,J),VY(K,J),RHOF(K,J)      BLUNT8  39
    5 CONTINUE                                                          BLUNT8  40
C                                                                       BLUNT8  41
C     PRINT BODY AND SHOCK LOCATIONS                                    BLUNT8  42
C                                                                       BLUNT8  43
      WRITE(6,100)                                                      BLUNT8  44
      WRITE(6,110) (J,XC(1,J),YC(1,J),XC(KMAX,J),YC(KMAX,J),J=2,JM)     BLUNT8  45
  100 FORMAT(///27H BODY AND FINAL SHOCK SHAPE//                        BLUNT8  46
     *  4X,1HJ,6X,7HXC(BODY),5X,7HR(BODY),6X,8HX(SHOCK),4X,8HR(SHOCK))  BLUNT8  47
  110 FORMAT(2X,I3,5X,F7.4,5X,F7.4,9X,F7.4,5X,F7.4)                     BLUNT8  48
C                                                                       BLUNT8  49
C.....STARTING PLANE FOR MARCHING CODE AT THETA=90 DEGREES              BLUNT8  50
C                                                                       BLUNT8  51
      P90=YC(1,JMAX-1)                                                  BLUNT8  52
      AMACH=XMACH                                                       BLUNT8  53
      P1INF=PINF                                                        BLUNT8  54
      R1INF=RINF                                                        BLUNT8  55
      V1INF=QINF                                                        BLUNT8  56
      KM2=KMAX-2                                                        BLUNT8  57
      HP0=HORN                                                          BLUNT8  58
      GAMF=GAM                                                          BLUNT8  59
      GAM1=GAMM1                                                        BLUNT8  60
      PK1=AMACH**2*0.5*GAM1                                             BLUNT8  61
      PK2=1.0+PK1                                                       BLUNT8  62
      PKPINF=PK2**(-GAM/GAM1)                                           BLUNT8  63
      PKRINF=PK2**(-1./GAM1)                                            BLUNT8  64
      PKQINF=SQRT(PK1/PK2)                                              BLUNT8  65
      DO 300 K=1,KMAX                                                   BLUNT8  66
      DO 300 I=1,3                                                      BLUNT8  67
      RHO=Q(JM,K,1)*D(JM,K)                                             BLUNT8  68
      U=Q(JM,K,2)/Q(JM,K,1)                                             BLUNT8  69
      V=Q(JM,K,3)/Q(JM,K,1)                                             BLUNT8  70
      EN=Q(JM,K,4)*D(JM,K)                                              BLUNT8  71
      P=GAM1*(EN-RHO*0.5*(U*U+V*V))                                     BLUNT8  72
      P90(K,I)=P*PKPINF/PINF                                            BLUNT8  73
      RHO90(K,I)=RHO*PKRINF/RINF                                        BLUNT8  74
      U90(K,I)=U*PKQINF/QINF                                            BLUNT8  75
      V90(K,I)=V*PKQINF/QINF                                            BLUNT8  76
      W90(K,I)=0.0                                                      BLUNT8  77
  300 CONTINUE                                                          BLUNT8  78
      DO 310 KK=1,3                                                     BLUNT8  79
      RS90(KK)=Y(JM,KMAX)                                               BLUNT8  80
      RSZ90(KK)=-XEX(JM,KMAX,2)/XEY(JM,KMAX,2)                          BLUNT8  81
      RSPH90(KK)=0.0                                                    BLUNT8  82
  310 CONTINUE                                                          BLUNT8  83
      RETURN                                                            BLUNT8  84
      END                                                               BLUNT8  85
```

```
      SUBROUTINE ABMATX(J,K,I)                                          ABMATX   2
      COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT,  COM1     2
     > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTURT,PINF,    COM1     3
     <RINF,QINF,CINF,JCS,TM,CLUS,PT,HORN,RNOSE,NCASE,NPUNCH             COM1     4
      COMMON /COM2/ X(25,20),Y(25,20),XET(25,20,2),XEX(25,20,2),        COM2     2
     *  XEY(25,20,2),D(25,20)                                          COM2     3
      COMMON /COM3/ Q(25,20,4),EF(25,4),S(25,20,4),G(4),AB(4,4)         COM3     2
C...FORM JACOBIAN MATRICES AT A GIVEN J-K NODE POINT.  A MATRIX IF I=1,  ABMATX   6
C   B MATRIX IF I=2.                                                    ABMATX   7
      XX=XET(J,K,I)                                                     ABMATX   8
      YY=XEX(J,K,I)                                                     ABMATX   9
      ZZ=XEY(J,K,I)                                                     ABMATX  10
      RI=1.0/Q(J,K,1)                                                   ABMATX  11
      U=Q(J,K,2)*RI                                                     ABMATX  12
      V=Q(J,K,3)*RI                                                     ABMATX  13
      SS=GAMM1*0.5*(U*U+V*V)                                            ABMATX  14
      T=YY*U+ZZ*V                                                       ABMATX  15
      W=GAM*Q(J,K,4)*RI                                                 ABMATX  16
      AB(1,1)=XX                                                        ABMATX  17
      AB(1,2)=YY                                                        ABMATX  18
      AB(1,3)=ZZ                                                        ABMATX  19
      AB(1,4)=0.0                                                       ABMATX  20
      AB(2,1)=YY*SS-U*T                                                 ABMATX  21
```

APPENDIX B

```
      AB(2,2)=XX-YY*(GAM-2.0)*U*T                                       ABMATX  22
      AB(2,3)=-YY*GAMM1*V+ZZ*U                                          ABMATX  23
      AB(2,4)=YY*GAMM1                                                  ABMATX  24
      AB(3,1)=ZZ*SS-V*T                                                 ABMATX  25
      AB(3,2)=YY*V-ZZ*GAMM1*U                                           ABMATX  26
      AB(3,3)=XX-ZZ*(GAM-2.0)*V*T                                       ABMATX  27
      AB(3,4)=ZZ*GAMM1                                                  ABMATX  28
      AB(4,1)=T*(2.0*SS-W)                                              ABMATX  29
      AB(4,2)=(W-SS)*YY-GAMM1*T*U                                       ABMATX  30
      AB(4,3)=(W-SS)*ZZ-GAMM1*T*V                                       ABMATX  31
      AB(4,4)=XX+GAM*T                                                  ABMATX  32
      RETURN                                                            ABMATX  33
      END                                                               ABMATX  34


      SUBROUTINE BNDRY                                                  BNDRY   2
      COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT, COM1    2
     > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTURT,PINF,   COM1    3
     <RINF,GINF,CINF,JCS,TM,CLUS,PT,HURN,RNOSE,NCASE,NPUNCH             COM1    4
      COMMON /COM2/ X(25,20),Y(25,20),XET(25,20,2),XEX(25,20,2),        COM2    2
     *  XEY(25,20,2),U(25,20)                                          COM2    3
      COMMON /COM3/ Q(25,20,4),FF(25,4),S(25,20,4),G(4),AB(4,4)         COM3    2
      DIMENSION P(25,3),PXI(25),PETA(25),U(25,3),UXI(25),UETA(25),      BNDRY   6
     *  V(25,3),VXI(25),VETA(25),R(25,3)                               BNDRY   7
C...USE REFLECTION TO SIMULATE PLANE OF SYMMETRY AT J=2                 BNDRY   8
      DO 12 K=1,KMAX                                                    BNDRY   9
      U(2,K)=U(1,K)                                                     BNDRY  10
      Q(1,K,1)=Q(2,K,1)*DD                                              BNDRY  11
      Q(1,K,2)=Q(2,K,2)*DD                                              BNDRY  12
      Q(1,K,3)=-Q(2,K,3)*DD                                             BNDRY  13
   12 Q(1,K,4)=Q(2,K,4)*DD                                              BNDRY  14
C...USE FIRST ORDER EXTRAPOLATION TO SIMULATE SUPERSONIC OUTFLOW        BNDRY  15
C...BOUNDARY CONDITION AT JMAX                                          BNDRY  16
      DO 1 N=1,4                                                        BNDRY  17
      DO 1 K=2,KM                                                       BNDRY  18
    1 Q(JMAX,K,N)=(2.0*Q(JM,K,N)*D(JM,K)-Q(JM-1,K,N)*D(JM-1,K))/        BNDRY  19
     > D(JMAX,K)                                                        BNDRY  20
C...SATISFY TANGENCY CONDITION USING CHARACTERISTIC EQUATION            BNDRY  21
      DO 3 K=1,3                                                        BNDRY  22
      DO 3 J=1,JMAX                                                     BNDRY  23
      Z=1.0/Q(J,K,1)                                                    BNDRY  24
      P(J,K)=Q(J,K,1)*D(J,K)                                            BNDRY  25
      U(J,K)=Q(J,K,2)*Z                                                 BNDRY  26
      V(J,K)=Q(J,K,3)*Z                                                 BNDRY  27
      E2=Q(J,K,4)*D(J,K)                                                BNDRY  28
    3 P(J,K)=(E2-0.5*R(J,K)*(U(J,K)**2+V(J,K)**2))*GAMM1                BNDRY  29
C...COMPUTE P-XI, U-XI, V-XI, P-ETA, U-ETA, AND V-ETA DERIVATIVES       BNDRY  30
      DO 4 J=2,JM                                                       BNDRY  31
      PXI(J)=(P(J+1,1)-P(J-1,1))*0.5                                    BNDRY  32
      UXI(J)=(U(J+1,1)-U(J-1,1))*0.5                                    BNDRY  33
    4 VXI(J)=(V(J+1,1)-V(J-1,1))*0.5                                    BNDRY  34
      PXI(1)=-PXI(2)                                                    BNDRY  35
      UXI(1)=-UXI(2)                                                    BNDRY  36
      VXI(1)=VXI(2)                                                     BNDRY  37
      PXI(JMAX)=(3.0*P(JMAX,1)-4.0*P(JM,1)+P(JM-1,1))*0.5               BNDRY  38
      UXI(JMAX)=(3.0*U(JMAX,1)-4.0*U(JM,1)+U(JM-1,1))*0.5               BNDRY  39
      VXI(JMAX)=(3.0*V(JMAX,1)-4.0*V(JM,1)+V(JM-1,1))*0.5               BNDRY  40
      DO 5 J=1,JMAX                                                     BNDRY  41
      PETA(J)=(-3.0*P(J,1)+4.0*P(J,2)-P(J,3))*0.5                       BNDRY  42
      UETA(J)=(-3.0*U(J,1)+4.0*U(J,2)-U(J,3))*0.5                       BNDRY  43
      VETA(J)=(-3.0*V(J,1)+4.0*V(J,2)-V(J,3))*0.5                       BNDRY  44
      IF(J.EQ.1.OR.J.EQ.JMAX) GO TO 5                                   BNDRY  45
      P(J,2)=P(J+1,1)+P(J-1,1)-2.0*P(J,1)                               BNDRY  46
    5 CONTINUE                                                          BNDRY  47
      P(1,2)=P(2,2)                                                     BNDRY  48
      P(JMAX,2)=0.0                                                     BNDRY  49
      K=1                                                               BNDRY  50
      DO 2 J=1,JMAX                                                     BNDRY  51
      CBB=SQRT(GAM*P(J,1)/R(J,1))                                       BNDRY  52
      Z=1.0/SQRT(XEX(J,1,2)**2+XEY(J,1,2)**2)                           BNDRY  53
      UBAR=XET(J,1,1)*U(J,1)+XEX(J,1,1)*V(J,1)+XEY(J,1,1)               BNDRY  54
      VBAR=XET(J,1,2)*U(J,1)+XEX(J,1,2)*V(J,1)+XEY(J,1,2)               BNDRY  55
      EE=UBAR*PXI(J)+R(J,1)*CBB**2*(XEX(J,1,1)*UXI(J)+XEY(J,1,1)*VXI(J))BNDRY  56
     > -CBB*Z*(XEX(J,1,2)*(XEX(J,1,1)*PXI(J)+R(J,1)*UBAR*UXI(J))+       BNDRY  57
     > XEY(J,1,2)*(XEY(J,1,1)*PXI(J)+R(J,1)*UBAR*VXI(J)))              BNDRY  58
     > +(V(J,1)/Y(J,1))*FLOAT(JCS)                                      BNDRY  59
      PTAU=CBB/Z*PETA(J)-R(J,1)*CBB**2*(XEX(J,1,2)*UETA(J)+XEY(J,1,2)*  BNDRY  60
     > VETA(J))-EE                                                      BNDRY  61
      P1=P(J,1)+PTAU*DT*0.2+0.10*P(J,2)                                 BNDRY  62
      IF (P1 .LT. 0.0) GO TO 9                                          BNDRY  63
   10 CONTINUE                                                          BNDRY  64
      IF(J.LE.2) P1=PT+(P(3,1)-PT)*0.111111111                          BNDRY  65
      R1=(P1/ENT)**(1.0/GAM)                                            BNDRY  66
      C1=SQRT(2.0*GAM/GAMM1*ABS(PTORT-P1/R1))                           BNDRY  67
      IF(ABS(XEY(J,1,2))-0.000001) 6,6,7                                BNDRY  68
    6 THETB=1.5707963                                                   BNDRY  69
      GO TO 8                                                           BNDRY  70
    7 THETB=ATAN(-XEX(J,1,2)/XEY(J,1,2))                                BNDRY  71
    8 CONTINUE                                                          BNDRY  72
      U1=Q1*COS(THETB)                                                  BNDRY  73
      V1=Q1*SIN(THETB)                                                  BNDRY  74
      THETBD=90.0-THETB*57.2958                                         BNDRY  75
      D1=1.0/D(J,1)                                                     BNDRY  76
      Q(J,1,1)=R1*D1                                                    BNDRY  77
      Q(J,1,2)=R1*U1*D1                                                 BNDRY  78
      Q(J,1,3)=R1*V1*D1                                                 BNDRY  79
      Q(J,1,4)=(P1/GAMM1+0.5*R1*Q1**2)*D1                               BNDRY  80
    2 CONTINUE                                                          BNDRY  81
      RETURN                                                            BNDRY  82
C                                                                       BNDRY  83
C     PRINT MESSAGE FOR NEGATIVE PRESSURE                               BNDRY  84
C                                                                       BNDRY  85
    9 WRITE(6,11) P1,J                                                  BNDRY  86
      P1=ABS(P1)                                                        BNDRY  87
      GO TO 10                                                          BNDRY  88
  11  FORMAT(5X,44HNEGATIVE PRESSURE ON BODY DETECTED BY BNDRY,,2X,     BNDRY  89
     *  3HPP=,1PE10.3,6H AT J=,I2)                                      BNDRY  90
      END                                                               BNDRY  91


      SUBROUTINE BODY(Y,X,RNOSE,ANG,H)                                  BODY    2
      E(A)=EXP(-ABS(A-RNOSE)/H)                                         BODY    3
      F(A,B,K)=0.5*A*((SIN(2.*B)+(-1.)**K*2.*SQRT(E(A)-E(A)**2))        BODY    4
     >       /(E(A)-SIN(B)**2))                                         BODY    5
      G(A,B)=ABS(A*((A**6*SIN(B)*COS(B)+SQRT(A**6-1.))/(A**6*COS(B)**2-1BODY    6
     *          .)))                                                    BODY    7
C                                                                       BODY    8
C..... H .NE. 0.0 ; IONOPAUSE FOR A NONMAGNETIC PLANET                  BODY    9
C..... H .EQ. 0.0 ; EQUATORIAL PLANE FOR A MAGNETIC PLANET              BODY   10
C                                                                       BODY   11
      IF(H .EQ. 0.0)GO TO 10                                            BODY   12
C                                                                       BODY   13
C.....THIS DETERMINES THE BODY SHAPE OF A NONMAGNETIC PLANET            BODY   14
C                                                                       BODY   15
C                                                                       BODY   16
      IMAX=301                                                          BODY   17
      IMAXM1=IMAX-1                                                     BODY   18
      IF(ANG .EQ. 0.0)IMAXM1=1                                          BODY   19
      PI=3.14159265358898                                               BODY   20
      RAD=180./PI                                                       BODY   21
      ANG=ANG*RAD                                                       BODY   22
      DELTAT=ANG/FLOAT(IMAXM1)                                          BODY   23
      THETA=0.                                                          BODY   24
      R1=RNOSE                                                          BODY   25
      ISTART=1                                                          BODY   26
      IF (H.LE.0.1) G O TO 20                                           BODY   27
C                                                                       BODY   28
C.....PERFORM INTEGRATION FROM 0 TO SANGS DEGREES                       BODY   29
C                                                                       BODY   30
   2  CONTINUE                                                          BODY   31
      DO 1 I=ISTART,IMAXM1                                              BODY   32
C.....PREDICTOR                                                         BODY   33
      K=0                                                               BODY   34
      XX=THETA/RAD                                                      BODY   35
      N=1                                                               BODY   36
      R=R1+DELTAT/RAD*F(R1,XX,N)                                        BODY   37
      THETA=THETA+DELTAT                                                BODY   38
      XXX=THETA/RAD                                                     BODY   39
C.....CORRECTOR                                                         BODY   40
   1  CONTINUE                                                          BODY   41
      R=R1+0.5*DELTAT/RAD*(F(R1,XX,N)+F(R,XXX,N))                       BODY   42
      K=K+1                                                             BODY   43
      IF(K .LT. 5)GO TO 1                                               BODY   44
      X=-R*COS(THETA/RAD)+1.0                                           BODY   45
      Y= R*SIN(THETA/RAD)                                               BODY   46
      R1=R                                                              BODY   47
   5  CONTINUE                                                          BODY   48
      RETURN                                                            BODY   49
  20  CONTINUE                                                          BODY   50
      IF (H .LT. 0.01) GO TO 30                                         BODY   51
      DO 35 I=1,IMAXM1                                                  BODY   52
      R=RNOSE+H*SIN(THETA/RAD)**2                                       BODY   53
      X=1.0-R*COS(THETA/RAD)                                            BODY   54
      Y=R*SIN(THETA/RAD)                                                BODY   55
      ISTART=1                                                          BODY   56
      R1=R                                                              BODY   57
      IF(THETA .GE. 5.0)GO TO 2                                         BODY   58
```

```
      THETA=THETA+DELTAT
   35 CONTINUE
C
C       USE SINUSOIDAL NOSE SHAPE FOR H/RO .LT. 1.01
C
   30 R=RNOSE+H*SIN(ANG/RAD)**2
      X=1.-R*COS(ANG/RAD)
      Y=R*SIN(ANG/RAD)
      RETURN
C
C.....THIS DETERMINES THE BODY SHAPE OF A MAGNETIC PLANET
C
   40 CONTINUE
      IMAX=361
      IMAXM1=IMAX-1
      IF(ANG .EQ. 0.)IMAXM1=1
      PI=3.14159265358979
      RAD=180./PI
      ANG=ANG*RAD
      DELTAT=ANG/FLOAT(IMAXM1)
      ANG=ANG+90.
      THETA=0.
      PI=RNOSE
C
C.....PERFORM INTEGRATION FROM 0 TO ANG DEGREES
C
      DO 15 I=1,IMAXM1
C.....PREDICTOR
      XX=THETA/RAD
      R=R1+DELTAT*G(R1,XX)/RAD
      THETA=THETA+DELTAT
      XXX=THETA/RAD
C.....CORRECTOR
      R=R1+0.5*DELTAT*(G(R1,XX)+G(R,XXX))/RAD
      X=-R*COS((THETA-90.)/RAD)+1.
      Y=R*SIN((THETA-90.)/RAD)
      R1=R
   15 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BTRI(IL,IU)                                            BTRI     2
      COMMON /COM3/ Q(25,20,4),F(25,4),S(25,20,4),G(4),AB(4,4)          BTRI     3
      COMMON /COM4/ A(25,4,4),B(25,4,4),C(25,4,4),HD(25,4,4),UD(25,4,4),COM4     2
     *  AX(25),AY(25),BX(25),BY(25)                                    COM4     3
      COMMON/LUD/ L11,L21,L22,L31,L32,L33,L41,L42,L43,L44,V1,V2,V3,V4,  BTRI     5
     1  U12,U13,U14,U23,U24,U34                                         BTRI     6
      DIMENSION H(4,4)                                                  BTRI     7
      REAL      L11,L21,L22,L31,L32,L33,L41,L42,L43,L44                 BTRI     8
C  INVERSION OF BLOCK TRIDIAGONAL... A,B,C ARE 4*4 BLOCKS               BTRI     9
C  F IS FORCING FUNCTION AND SOLUTION IS OUTPUT IN F.  B IS OVERLOADED   BTRI    10
C  BLOCK INVERSIONS USE NONPIVOTED LU DECOMPOSITION                     BTRI    11
C  IL AND IU ARE STARTING AND FINISHING INDICES                         BTRI    12
      IS = IL +1                                                        BTRI    13
      I = IL                                                            BTRI    14
      DO 11 N=1,4                                                       BTRI    15
      DO 11 M=1,4                                                       BTRI    16
   11 H(N,M)=B(I,N,M)                                                   BTRI    17
      CALL LUDEC(H)                                                     BTRI    18
      D1 = V1*F(I,1)                                                    BTRI    19
      D2 = V2*( F(I,2) - L21*D1)                                        BTRI    20
      D3 = V3*( F(I,3) - L31*D1 - L32*D2)                               BTRI    21
      D4 = V4*( F(I,4) - L41*D1 - L42*D2 - L43*D3)                      BTRI    22
      F(I,4) = D4                                                       BTRI    23
      F(I,3) = D3 - U34*D4                                              BTRI    24
      F(I,2) = D2 - U24*D4 - U23*F(I,3)                                 BTRI    25
      F(I,1) = D1 - U14*D4 - U13*F(I,3) - U12*F(I,2)                    BTRI    26
      DO 12 M=1,4                                                       BTRI    27
      D1 = V1*C(I,1,M)                                                  BTRI    28
      D2 = V2*( C(I,2,M) - L21*D1)                                      BTRI    29
      D3 = V3*( C(I,3,M) - L31*D1 - L32*D2)                             BTRI    30
      D4 = V4*( C(I,4,M) -L41*D1 - L42*D2 - L43*D3)                     BTRI    31
      B(I,4,M) = D4                                                     BTRI    32
      B(I,3,M) = D3 - U34*D4                                            BTRI    33
      B(I,2,M) = D2 - U24*D4 - U23*B(I,3,M)                             BTRI    34
      B(I,1,M) = D1 - U14*D4 - U13*B(I,3,M) - U12*B(I,2,M)              BTRI    35
C...FORWARD SWEEP                                                       BTRI    36
   12 CONTINUE                                                          BTRI    37
      DO 13 I=IS,IU                                                     BTRI    38
      IR = I -1                                                         BTRI    39
      DO 14 N=1,4                                                       BTRI    40
      F(I,N) = F(I,N) -A(I,N,1)*F(IR,1) -A(I,N,2)*F(IR,2)               BTRI    41
     1     -A(I,N,3)*F(IR,3) - A(I,N,4)*F(IR,4)                        BTRI    42
      DO 14 M=1,4                                                       BTRI    43
      H(N,M) = B(I,N,M) -A(I,N,1)*B(IR,1,M) -A(I,N,2)*B(IR,2,M)         BTRI    44
     1     -A(I,N,3)*B(IR,3,M) - A(I,N,4)*B(IR,4,M)                     BTRI    45
   14 CONTINUE                                                          BTRI    46
      CALL LUDEC(H)                                                     BTRI    47
      D1 = V1*F(I,1)                                                    BTRI    48
      D2 = V2*( F(I,2) - L21*D1)                                        BTRI    49
      D3 = V3*( F(I,3) - L31*D1 - L32*D2)                               BTRI    50
      D4 = V4*( F(I,4) - L41*D1 - L42*D2 - L43*D3)                      BTRI    51
      F(I,4) = D4                                                       BTRI    52
      F(I,3) = D3 - U34*D4                                              BTRI    53
      F(I,2) = D2 - U24*D4 - U23*F(I,3)                                 BTRI    54
      F(I,1) = D1 - U14*D4 - U13*F(I,3) - U12*F(I,2)                    BTRI    55
      IF( I - IU)16,13,13                                               BTRI    56
   16 CONTINUE                                                          BTRI    57
      DO 15 M=1,4                                                       BTRI    58
      D1 = V1*C(I,1,M)                                                  BTRI    59
      D2 = V2*( C(I,2,M) - L21*D1)                                      BTRI    60
      D3 = V3*( C(I,3,M) - L31*D1 - L32*D2)                             BTRI    61
      D4 = V4*( C(I,4,M) -L41*D1 - L42*D2 - L43*D3)                     BTRI    62
      B(I,4,M) = D4                                                     BTRI    63
      B(I,3,M) = D3 - U34*D4                                            BTRI    64
      B(I,2,M) = D2 - U24*D4 - U23*B(I,3,M)                             BTRI    65
      B(I,1,M) = D1 - U14*D4 - U13*B(I,3,M) - U12*B(I,2,M)              BTRI    66
   15 CONTINUE                                                          BTRI    67
   13 CONTINUE                                                          BTRI    68
C...BACK SUBSTITUTION                                                   BTRI    69
      IT = IL + IU                                                      BTRI    70
      DO 21 II = IS,IU                                                  BTRI    71
      I = IT - II                                                       BTRI    72
      IP = I+1                                                          BTRI    73
      DO 22 N=1,4                                                       BTRI    74
      F(I,N) = F(I,N) - B(I ,N,1)*F(IP,1) - B(I ,N,2)*F(IP,2)           BTRI    75
     1     -B(I ,N,3)*F(IP,3) - B(I ,N,4)*F(IP,4)                       BTRI    76
   22 CONTINUE                                                          BTRI    77
   21 CONTINUE                                                          BTRI    78
      RETURN                                                            BTRI    79
      END                                                               BTRI    80
```

```
      SUBROUTINE DISSIP                                                 DISSIP   2
      COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT, COM1     2
     > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTURT,PINF,   COM1     3
     <RINF,QINF,CINF,JCS,TM,CLUS,PT,HORN,RNOSE,NCASE,NPUNCH             COM1     4
      COMMON /COM2/ X(25,20),Y(25,20),XE(25,20,2),XEX(25,20,2),         COM2     2
     *  XEY(25,20,2),D(25,20)                                          COM2     3
      COMMON /COM3/ Q(25,20,4),CF(25,4),S(25,20,4),G(4),AB(4,4)         COM3     2
C...SMOOTH IN THE X AND Y DIRECTIONS AND ADD SMOOTHING TERM TO S ARRAY  DISSIP   6
      KMM=KM-1                                                          DISSIP   7
      JMM=JM-1                                                          DISSIP   8
C...SMOOTHING IN XI DIRECTION                                           DISSIP   9
      DO 1 N=1,4                                                        DISSIP  10
      DO 2 K=2,KM                                                       DISSIP  11
      S(2,K,N)=S(2,K,N)-SMU*0.125*(-2.0*Q(1,K,N)*D(1,K)+5.0*Q(2,K,N)    DISSIP  12
     > *D(2,K)-4.0*Q(3,K,N)*D(3,K)+Q(4,K,N)*D(4,K))/D(2,K)              DISSIP  13
      S(JM,K,N)=S(JM,K,N)-SMU*0.125*(-2.0*Q(JMAX,K,N)*D(JMAX,K)+        DISSIP  14
     > 5.0*Q(JM,K,N)*D(JM,K)-4.0*Q(JMM,K,N)*D(JMM,K)+Q(JM-2,K,N)        DISSIP  15
     > *D(JM-2,K))/D(JM,K)                                              DISSIP  16
      DO 2 J=3,JMM                                                      DISSIP  17
    2 S(J,K,N)=S(J,K,N)-SMU*0.125*(Q(J-2,K,N)*D(J-2,K)-4.0*Q(J-1,K,N)   DISSIP  18
     > *D(J-1,K)+6.0*Q(J,K,N)*D(J,K)-4.0*Q(J+1,K,N)*D(J+1,K)+Q(J+2,K,N) DISSIP  19
     > *D(J+2,K))/D(J,K)                                                DISSIP  20
C...SMOOTHING IN ETA DIRECTION                                          DISSIP  21
      DO 1 J=2,JM                                                       DISSIP  22
      S(J,2,N)=S(J,2,N)-SMU*0.125*(-2.0*Q(J,1,N)*D(J,1)+5.0*Q(J,2,N)    DISSIP  23
     > *D(J,2)-4.0*Q(J,3,N)*D(J,3)+Q(J,4,N)*D(J,4))/D(J,2)              DISSIP  24
      S(J,KM,N)=S(J,KM,N)-SMU*0.125*(-2.0*Q(J,KMAX,N)*D(J,KMAX)+        DISSIP  25
     > 5.0*Q(J,KM,N)*D(J,KM)-4.0*Q(J,KMM,N)*D(J,KMM)+Q(J,KM-2,N)        DISSIP  26
     > *D(J,KM-2))/D(J,KM)                                              DISSIP  27
      DO 1 K=3,KMM                                                      DISSIP  28
    1 S(J,K,N)=S(J,K,N)-SMU*0.125*(Q(J,K-2,N)*D(J,K-2)-4.0*Q(J,K-1,N)   DISSIP  29
     > *D(J,K-1)+6.0*Q(J,K,N)*D(J,K)-4.0*Q(J,K+1,N)*D(J,K+1)+Q(J,K+2,N) DISSIP  30
     > *D(J,K+2))/D(J,K)                                                DISSIP  31
      RETURN                                                            DISSIP  32
      END                                                               DISSIP  33
```

```
      SUBROUTINE EFCON(J,K,I)                                           EFCON    2
      COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT, COM1     2
     > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTURT,PINF,   COM1     3
     <RINF,QINF,CINF,JCS,TM,CLUS,PT,HORN,RNOSE,NCASE,NPUNCH             COM1     4
      COMMON /COM2/ X(25,20),Y(25,20),XE(25,20,2),XEX(25,20,2),         COM2     2
     *  XEY(25,20,2),D(25,20)                                          COM2     3
      COMMON /COM3/ Q(25,20,4),EF(25,4),S(25,20,4),G(4),AB(4,4)         COM3     2
C...FORM 4 CONSERVATIVE VARIABLES (I=1) OR 4 CONSERVATIVE VARIABLES     EFCON    6
```

```
C...(I=2) AT A GIVEN NODE POINT                                    EFCON   7
      W=Q(J,K,1)                                                   EFCON   8
      RI=1.0/W                                                     EFCON   9
      U=Q(J,K,2)*RI                                                EFCON  10
      V=Q(J,K,3)*RI                                                EFCON  11
      POJ=GAMM1*(Q(J,K,4)-W*0.5*(U*U+V*V))                         EFCON  12
      XX=XET(J,K,I)                                                EFCON  13
      YY=XEX(J,K,I)                                                EFCON  14
      Z=XEY(J,K,I)                                                 EFCON  15
      CAPUV=XX*YY*U+Z*V                                            EFCON  16
      G(1)=W*CAPUV                                                 EFCON  17
      G(2)=Q(J,K,2)*CAPUV+YY*POJ                                   EFCON  18
      G(3)=Q(J,K,3)*CAPUV+Z*POJ                                    EFCON  19
      G(4)=Q(J,K,4)*CAPUV+(CAPUV-XX)*POJ                           EFCON  20
C...SOURCE TERM IN ETA-MOMENTUM EQN. FOR AXISYMMETRIC FLOW         EFCON  21
      IF(JCS.EQ.0.OR.K.LE.KMAX.OR.K.EQ.KMAX) RETURN                EFCON  22
      YI=DT/Y(J,K)                                                 EFCON  23
      S(J,K,1)=S(J,K,1)+Q(J,K,3)*YI                                EFCON  24
      S(J,K,2)=S(J,K,2)+Q(J,K,3)*YI*U                              EFCON  25
      S(J,K,3)=S(J,K,3)+Q(J,K,3)*YI*V                              EFCON  26
      S(J,K,4)=S(J,K,4)+(Q(J,K,4)+POJ)*V*YI                        EFCON  27
      RETURN                                                       EFCON  28
      END                                                          EFCON  29


      SUBROUTINE EIGEN                                             EIGEN   2
      COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT,  COM1   2
     > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTORT,PINF,    COM1   3
     <RINF,OINF,CINF,JCS,TM,CLUS,PI,HORN,RNOSE,NCASE,NPUNCH              COM1   4
      COMMON /COM2/ X(25,20),Y(25,20),XET(25,20,2),XEX(25,20,2),         COM2   2
     *  XEY(25,20,2),D(25,20)                                          COM2   3
      COMMON /COM3/ Q(25,20,4),EF(25,4),S(25,20,4),G(4),AB(4,4)        COM3   2
C...COMPUTE STEPSIZE GIVEN COURANT NUMBER                          EIGEN   6
      IF(IPRT.GT.3) WRITE(6,100)                                   EIGEN   7
      SIGMAX=0.0                                                   EIGEN   8
      DO 1 K=1,KMAX                                                EIGEN   9
      DO 1 J=1,JMAX                                                EIGEN  10
      RI=1.0/Q(J,K,1)                                              EIGEN  11
      U=Q(J,K,2)*RI                                                EIGEN  12
      V=Q(J,K,3)*RI                                                EIGEN  13
      SPSND=SQRT(GAM*GAMM1*(Q(J,K,4)*RI-0.5*(U*U+V*V)))            EIGEN  14
      XIX=XEX(J,K,1)                                               EIGEN  15
      XIY=XEY(J,K,1)                                               EIGEN  16
      ETAX=XET(J,K,2)                                              EIGEN  17
      ETAY=XEY(J,K,2)                                              EIGEN  18
      SIGA=ABS(U*XIX+V*XIY)+SPSND*SQRT(XIX**2+XIY**2)              EIGEN  19
      SIGB=ABS(U*ETAX+V*ETAY)+SPSND*SQRT(ETAX**2+ETAY**2)          EIGEN  20
      SIGAB=AMAX1(SIGA,SIGB)                                       EIGEN  21
      IF(SIGAB.GT.SIGMAX) SIGMAX=SIGAB                             EIGEN  22
      IF(IPRT.EQ.3) GO TO 1                                        EIGEN  23
      WRITE(6,101) J,K,SIGA,SIGB                                   EIGEN  24
    1 CONTINUE                                                     EIGEN  25
      DT=CN/ABS(SIGMAX)                                            EIGEN  26
      H=0.5*DT                                                     EIGEN  27
  100 FORMAT(1H0,3X,1HJ,4X,1HK,7X,4HSIGA,8X,4HSIGB)                EIGEN  28
  101 FORMAT(2I5,2E12.6)                                           EIGEN  29
  102 FORMAT(//22H DETERMINING STEPSIZE=,3X,7HSIGMAX=,1PE11.4,3X,3HCN=,  EIGEN  30
     *  E9.2,3X,3HDT=,E11.4//)                                     EIGEN  31
      RETURN                                                       EIGEN  32
      END                                                          EIGEN  33


      SUBROUTINE INITIA                                            INITIA   2
      COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT,  COM1   2
     > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTORT,PINF,    COM1   3
     <RINF,OINF,CINF,JCS,TM,CLUS,PI,HORN,RNOSE,NCASE,NPUNCH              COM1   4
      COMMON /COM2/ X(25,20),Y(25,20),XET(25,20,2),XEX(25,20,2),         COM2   2
     *  XEY(25,20,2),D(25,20)                                          COM2   3
      COMMON /COM3/ Q(25,20,4),EF(25,4),S(25,20,4),G(4),AB(4,4)        COM3   2
      COMMON /COM4/ A(25,4,4),B(25,4,4),C(25,4,4),HD(25,4,4),UL(25,4,4),  COM4   2
     *  AX(25),AY(25),BX(25),BY(25)                                    COM4   3
C...INITIALISE (AX,BX),(AY,BY) TO (0,1)                            INITIA   7
      DO 9 I=1,25                                                  INITIA   8
      AX(I)=0.0                                                    INITIA   9
      BX(I)=1.0                                                    INITIA  10
      AY(I)=0.0                                                    INITIA  11
    9 BY(I)=1.0                                                    INITIA  12
      PI=3.141592654                                               INITIA  13
      RADI=57.295779451                                            INITIA  14
      PINF=1.0                                                     INITIA  15
      RINF=1.0                                                     INITIA  16
      IPRT=0                                                       INITIA  17
      ALPHA=0.0                                                    INITIA  18
      JCS=1                                                        INITIA  19
      PRESS=1.0/GAM                                                INITIA  20
      SMU=0.0                                                      INITIA  21
      CINF=SQRT(PINF*GAM/RINF)                                     INITIA  22
      ALPH=ALPHA/RADI                                              INITIA  23
      UINF=XMACH*CINF                                              INITIA  24
      UINF=UINF*COS(ALPH)                                          INITIA  25
      VINF=UINF*SIN(ALPH)                                          INITIA  26
      JM=JMAX-1                                                    INITIA  27
      KM=KMAX-1                                                    INITIA  28
      GAMM1=GAM-1.0                                                INITIA  29
      GAM11=1.0/GAMM1                                              INITIA  30
      TAU=0.0                                                      INITIA  31
      IT=0                                                         INITIA  32
C...LOCATE X AND Y NODAL POINTS OF MESH                            INITIA  33
      CALL NODPTS                                                  INITIA  34
C...COMPUTE METRICS                                                INITIA  35
      CALL XIETAD(0)                                               INITIA  36
C...INITIALIZE Q VECTOR TO FREE STREAM VALUES                      INITIA  37
      DO 2 K=1,KMAX                                                INITIA  38
      DO 2 J=1,JMAX                                                INITIA  39
      DI=1.0/D(J,K)                                                INITIA  40
      Q(J,K,1)=RINF*DI                                             INITIA  41
      Q(J,K,2)=RINF*UINF*DI                                        INITIA  42
      Q(J,K,3)=RINF*VINF*DI                                        INITIA  43
      Q(J,K,4)=(PINF*GAM11+RINF*UINF**2*0.5)*DI                    INITIA  44
C...SET S ARRAY TO 0 EVERYWHERE                                    INITIA  45
      DO 1 N=1,4                                                   INITIA  46
    1 S(J,K,N)=0.0                                                 INITIA  47
C...INITIALIZE SECOND OR FOURTH ORDER FINITE-DIFFERENCE COEFFICIENTS  INITIA  48
C.....SECOND ORDER - SEE DATA STATEMENT ABOVE                      INITIA  49
C...INITIALIZE FLOW FIELD FOR BLUNT BODY PROBLEM                   INITIA  50
      GAMP1=GAM+1.0                                                INITIA  51
      KM3=KMAX                                                     INITIA  52
      DO 2 J=2,JMAX                                                INITIA  53
      IF(ABS(XEX(J,1,2))-0.000001) 5,6,7                           INITIA  54
    6 THET=90./RADI                                                INITIA  55
      GO TO 8                                                      INITIA  56
    7 THET=ATAN(-XEY(J,1,2)/XEX(J,1,2))                            INITIA  57
    8 CONTINUE                                                     INITIA  58
      K=KMAX                                                       INITIA  59
      SANG=0.5*PI-ATAN(-XEY(J,K,2)/XEX(J,K,2))                     INITIA  60
      XX=XMACH**2*SIN(SANG)**2                                     INITIA  61
      PS=(2.0*GAM*XX-GAMM1)/GAMP1*PINF                             INITIA  62
      RS=GAMP1*XX/(GAMM1*XX+2.0)*RINF                              INITIA  63
      US=(2.0-2.0*(XX-1.0)/GAMP1/XMACH**2)*CINF                    INITIA  64
      VS=2.0*(XX-1.0)*COS(SANG)/(GAMP1*XMACH**2*SIN(SANG))*CINF    INITIA  65
      IF(J.GT.2) GO TO 3                                           INITIA  66
      X1=(2.0*GAM*XMACH**2-GAMM1)/(GAM+1.0)                        INITIA  67
      X2=(GAM+1.0)*XMACH**2/(GAMM1*XMACH**2+2.0)                   INITIA  68
      P1=X1*PINF                                                   INITIA  69
      R1=X2*RINF                                                   INITIA  70
      ENT=P1/R1**GAM                                               INITIA  71
      PT=(1.0/X2)**(1.0/GAMM1)*(0.5*(GAM+1.0)*XMACH**2)**(GAM/GAMM1)*PINF  INITIA  72
      XX=1.0+1.0*GAMM1*XMACH**2                                    INITIA  73
      PTORT=XX*PINF/RINF                                           INITIA  74
    3 CONTINUE                                                     INITIA  75
      PP=PINF*((PT/PINF-1.0)*(1.0-1.02*SIN(THET)**2+0.12*SIN(THET)**4)+  INITIA  76
     > 1.0)                                                        INITIA  77
      RB=(PB/ENT)**(1.0/GAM)                                       INITIA  78
      QB=SQRT(2.0*GAM/GAMM1*ABS(PTORT-PB/RB))                      INITIA  79
      YY=PI*0.5-THET                                               INITIA  80
      UB=QB*COS(YY)                                                INITIA  81
      VB=QB*SIN(YY)                                                INITIA  82
      ZKM=KM3-1                                                    INITIA  83
      DO 2 K=1,KM3                                                 INITIA  84
      YY=FLOAT(K-1)/ZKM                                            INITIA  85
      PRESS=PB+YY*(PS-PB)                                          INITIA  86
      RHO=RB+YY*(RS-RB)                                            INITIA  87
      UVEL=UB+YY*(US-UB)                                           INITIA  88
      VVEL=VB+YY*(VS-VB)                                           INITIA  89
      DI=1.0/D(J,K)                                                INITIA  90
      Q(J,K,1)=RHO*DI                                              INITIA  91
      Q(J,K,2)=RHO*UVEL*DI                                         INITIA  92
      Q(J,K,3)=RHO*VVEL*DI                                         INITIA  93
    2 Q(J,K,4)=(PRESS*GAM11+RHO*(UVEL**2+VVEL**2)*0.5)*DI          INITIA  94
C...REFLECT METRICS AND DEPENDENT VARIABLES ABOUT PLANE OF SYMMETRY  INITIA  95
      DO 4 K=1,KMAX                                                INITIA  96
      DD=(-1.0)**JCS                                               INITIA  97
      DD=(+1.0)**JCS                                               INITIA  98
      D(1,K)=D(2,K)*DD                                             INITIA  99
      XEX(1,K,1)=-XEX(2,K,1)                                       INITIA 100
      XEY(1,K,1)=XEY(2,K,1)                                        INITIA 101
      XEX(1,K,2)=XEX(2,K,2)                                        INITIA 102
      XEY(1,K,2)=-XEY(2,K,2)                                       INITIA 103
      DO 5 N=1,4                                                   INITIA 104
```

```
     > G(1,K,N)=G(2,K,N)*DD                                              INITIA 105
     * G(1,K,3)=-G(2,K,3)*DD                                             INITIA 106
       RETURN                                                           INITIA 107
       END                                                              INITIA 108


       SUBROUTINE INTEGR
       COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT,  INTEGR   2
      > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTURT,PINF,    COM1     2
     <RINF,QINF,CINF,JCS,TM,CLUS,PT,HORN,RNOSE,NCASE,NPUNCH               COM1     3
       COMMON /COM3/ Q(25,20,4),EF(25,4),S(25,20,4),G(4),AB(4,4)          COM1     4
C...COMPUTE FORCING FUNCTION AND STORE TEMPORARILY IN S ARRAY             COM3     2
       CALL RHS                                                         INTEGR   5
C...ADD FOURTH ORDER DISSIPATION TO SMOOTH SOLUTION                     INTEGR   6
       CALL DISSIP                                                      INTEGR   7
C...SOLVE FOR Q-BAR-BAR                                                 INTEGR   8
       DO 1 K=2,KM                                                      INTEGR   9
C...FILL ELEMENTS OF I+H*DX A FOR BLOCK TRIDIAGONAL INVERSION AT EACH   INTEGR  10
C...K TH LEVEL                                                          INTEGR  11
       CALL LBLTRA(K)                                                   INTEGR  12
C...INVERT BLOCK TRIDIAGONAL MATRIX AT K TH LEVEL AND STORE SOLUTION IN INTEGR  13
C...S ARRAY                                                             INTEGR  14
       CALL BTRI(2,JM)                                                  INTEGR  15
       DO 1 L=1,4                                                       INTEGR  16
       DO 1 J=2,JM                                                      INTEGR  17
     1 S(J,K,L)=EF(J,L)                                                 INTEGR  18
C...SOLVE FOR C-BAR                                                     INTEGR  19
       DO 2 J=2,JM                                                      INTEGR  20
C...FILL ELEMENTS OF I+H*DY B FOR BLOCK TRIDIAGONAL INVERSION AT EACH   INTEGR  21
C...J TH LEVEL                                                          INTEGR  22
       CALL LBLTRB(J)                                                   INTEGR  23
C...INVERT BLOCK TRIDIAGONAL MATRIX AT J TH LEVEL AND STORE SOLUTION IN INTEGR  24
C...J ARRAY                                                             INTEGR  25
       CALL BTRI(2,KM)                                                  INTEGR  26
       DO 2 L=1,4                                                       INTEGR  27
       DO 2 K=2,KM                                                      INTEGR  28
     2 Q(J,K,L)=EF(K,L)+Q(J,K,L)                                        INTEGR  29
       TAU=TAU+DT                                                       INTEGR  30
       RETURN                                                           INTEGR  31
       END                                                              INTEGR  32
                                                                        INTEGR  33


       SUBROUTINE LBLTRA(K)                                             LBLTRA   2
       COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT,  COM1     2
      > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTURT,PINF,    COM1     3
     <RINF,QINF,CINF,JCS,TM,CLUS,PT,HORN,RNOSE,NCASE,NPUNCH               COM1     4
       COMMON /COM3/ Q(25,20,4),EF(25,4),S(25,20,4),G(4),AB(4,4)          COM3     2
       COMMON /COM4/ A(25,4,4),B(25,4,4),C(25,4,4),HD(25,4,4),UD(25,4,4),  COM4     2
      *  AX(25),AY(25),BX(25),BY(25)                                     COM4     3
       DO 1 J=1,JMAX                                                    LBLTRA   6
C...LOAD BLOCK A MATRIX EVALUATED AT N TH LEVEL FOR ALL J INTO HD ARRAY LBLTRA   7
       CALL ABMATX(J,K,1)                                               LBLTRA   8
       DO 1 M=1,4                                                       LBLTRA   9
       DO 1 L=1,4                                                       LBLTRA  10
     1 HD(J,L,M)=AB(L,M)                                                LBLTRA  11
C...FILL OFF-DIAGONAL AND DIAGONAL ELEMENTS BASED ON A 2-ND ORDER       LBLTRA  12
C...CENTRAL DIFFERENCE                                                  LBLTRA  13
       DO 2 J=2,JM                                                      LBLTRA  14
       DO 2 M=1,4                                                       LBLTRA  15
       DO 3 L=1,4                                                       LBLTRA  16
       A(J,L,M)=-HD(J-1,L,M)*H                                          LBLTRA  17
       B(J,L,M)=0.0                                                     LBLTRA  18
     3 C(J,L,M)=HD(J+1,L,M)*H                                           LBLTRA  19
C...FILL FORCING FUNCTION FROM S ARRAY                                  LBLTRA  20
       EF(J,M)=S(J,K,M)                                                 LBLTRA  21
C...SET B ON THE DIAGONAL REPRESENTING THE IDENTITY MATRIX TO ONE       LBLTRA  22
     2 B(J,M,M)=1.0                                                     LBLTRA  23
       RETURN                                                           LBLTRA  24
       END                                                              LBLTRA  25


       SUBROUTINE LBLTRB(J)                                             LBLTRB   2
       COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT,  COM1     2
      > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTURT,PINF,    COM1     3
     <RINF,QINF,CINF,JCS,TM,CLUS,PT,HORN,RNOSE,NCASE,NPUNCH               COM1     4
       COMMON /COM3/ Q(25,20,4),EF(25,4),S(25,20,4),G(4),AB(4,4)          COM3     2
       COMMON /COM4/ A(25,4,4),B(25,4,4),C(25,4,4),HD(25,4,4),UD(25,4,4),  COM4     2
      *  AX(25),AY(25),BX(25),BY(25)                                     COM4     3
       DO 1 K=1,KMAX                                                    LBLTRB   6
C...LOAD BLOCK B MATRIX EVALUATED AT N TH LEVEL FOR ALL K INTO HD ARRAY LBLTRB   7
       CALL ABMATX(J,K,2)                                               LBLTRB   8
       DO 1 M=1,4                                                       LBLTRB   9
       DO 1 L=1,4                                                       LBLTRB  10
     1 HD(K,L,M)=AB(L,M)                                                LBLTRB  11
C...FILL OFF-DIAGONAL AND DIAGONAL ELEMENTS BASED ON A 2-ND ORDER       LBLTRB  12
C...CENTRAL DIFFERENCE                                                  LBLTRB  13
       DO 2 K=2,KM                                                      LBLTRB  14
       DO 2 M=1,4                                                       LBLTRB  15
       DO 3 L=1,4                                                       LBLTRB  16
       A(K,L,M)=-HD(K-1,L,M)*H                                          LBLTRB  17
       B(K,L,M)=0.0                                                     LBLTRB  18
     3 C(K,L,M)=HD(K+1,L,M)*H                                           LBLTRB  19
C...FILL FORCING FUNCTION FROM S ARRAY                                  LBLTRB  20
       EF(K,M)=S(J,K,M)                                                 LBLTRB  21
C...SET B ON THE DIAGONAL REPRESENTING THE IDENTITY MATRIX TO ONE       LBLTRB  22
     2 B(K,M,M)=1.0                                                     LBLTRB  23
       RETURN                                                           LBLTRB  24
       END                                                              LBLTRB  25


       SUBROUTINE LUDEC(A)                                              LUDEC    2
       DIMENSION A(4,4)                                                 LUDEC    3
       COMMON/LUD/ L11,L21,L22,L31,L32,L33,L41,L42,L43,L44,V1,V2,V3,V4,  LUDEC    4
      1  U12,U13,U14,U23,U24,U34                                         LUDEC    5
       REAL       L11,L21,L22,L31,L32,L33,L41,L42,L43,L44               LUDEC    6
C  SUBROUTINE COMPUTES L-U DECOMPOSITION ELEMENTS                        LUDEC    7
       L11 = A(1,1)                                                     LUDEC    8
       V1 = 1./L11                                                      LUDEC    9
       U12 =  V1*A(1,2)                                                 LUDEC   10
       U13 =  V1*A(1,3)                                                 LUDEC   11
       U14 =  V1*A(1,4)                                                 LUDEC   12
       L21 = A(2,1)                                                     LUDEC   13
       L22 = A(2,2) - L21*U12                                           LUDEC   14
       V2 = 1./L22                                                      LUDEC   15
       U23 = ( A(2,3) -L21*U13)* V2                                     LUDEC   16
       U24 = ( A(2,4) - L21*U14)* V2                                    LUDEC   17
       L31 = A(3,1)                                                     LUDEC   18
       L32 = A(3,2) - L31*U12                                           LUDEC   19
       L33 = A(3,3) - L31*U13 -L32*U23                                  LUDEC   20
       V3 = 1./L33                                                      LUDEC   21
       U34 = ( A(3,4) - L31*U14 - L32*U24)* V3                          LUDEC   22
       L41 = A(4,1)                                                     LUDEC   23
       L42 = A(4,2) - L41*U12                                           LUDEC   24
       L43 = A(4,3) - L41*U13 - L42*U23                                 LUDEC   25
       L44 = A(4,4) - L41*U14 -L42*U24 -L43*U34                         LUDEC   26
       V4 = 1./L44                                                      LUDEC   27
       RETURN                                                           LUDEC   28
       END                                                              LUDEC   29


       SUBROUTINE NDLPTS                                                NDLPTS   2
       COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT,  COM1     2
      > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTURT,PINF,    COM1     3
     <RINF,QINF,CINF,JCS,TM,CLUS,PT,HORN,RNOSE,NCASE,NPUNCH               COM1     4
       COMMON /COM2/ X(25,20),Y(25,20),XET(25,20,2),XCX(25,20,2),       COM2     2
      *  XEY(25,20,2),D(25,20)                                           COM2     3
       COMMON /NUBOD/ XX(100),RP(100),NBOD                              NUBOD    2
C...THIS SUBROUTINE DETERMINES THE X AND Y LOCATIONS OF THE NODE POINTS  NDLPTS   6
       LOGICAL NUBODY                                                   NDLPTS   7
       NUBODY=(HORN.LT.0.0)                                             NDLPTS   8
C...TM=THETA MAX                                                        NDLPTS   9
       IF(NUBODY) GO TO 2                                               NDLPTS  10
       DTHET=1.57079633/(FLOAT(JMAX)-2.5)                               NDLPTS  11
       TM=DTHET*(FLOAT(JMAX)-1.5)                                       NDLPTS  12
     2 CONTINUE                                                         NDLPTS  13
C...J=0 FOR CYLINDER, J=1 FOR SPHERE.                                   NDLPTS  14
       DELTC=(GAMM1*XMACH**2+2.0)/((GAM+1.0)*XMACH**2)+0.76*3.0**(1-JCS)NDLPTS  15
       IF(XMACH.LT.3.) DELTC=(2.02476*(XMACH-1.25)**2-3.65190*(XMACH-   NDLPTS  16
      > 1.25)+4.41)/4.0**JCS                                            NDLPTS  17
       IF (NUBODY) GO TO 10                                             NDLPTS  18
       IF(HORN .LT. 0.1 .AND. HORN .GT. 0.0)GO TO 10                    NDLPTS  19
       ANG=90./57.2957795                                               NDLPTS  20
       CALL BODY(Y90,X90,RNOSE,ANG,HORN)                                NDLPTS  21
       DELTC=DELTC*(1.2730+0.0990*XMACH)*(0.9044+0.055*HORN)            NDLPTS  22
       DELTI=DELTC*(3.95-5.3*HORN+3.85*HORN**2)*Y90                     NDLPTS  23
       DELTO=1.1*((GAMM1*XMACH**2+2.)/((GAM+1.)*XMACH**2))*(1.90+.5*HORN)NDLPTS  24
  10   CONTINUE                                                         NDLPTS  25
       DO 4 J=2,JMAX                                                    NDLPTS  26
       IF(NUBODY) GO TO 12                                              NDLPTS  27
       THET=(FLOAT(J)-1.5)*DTHET                                        NDLPTS  28
C...INITIAL SHOCK SHAPE                                                 NDLPTS  29
       IF(HORN .LT. 0.1 .AND. HORN .GT. 0.0)GO TO 15                    NDLPTS  30
       BBB=DELTI**2*(1.-SIN(THET)**2)/SIN(THET)**2/(1.+DELTO)           NDLPTS  31
       FOURAC=4.*BBB*(1.+DELTI)                                         NDLPTS  32
       XA=1.+(BBB-SQRT(BBB**2+FOURAC))*0.5                              NDLPTS  33
       THET90=1.570796326879                                            NDLPTS  34
       IF((THET-THET90) .GE. 0.0)XA=1.+(BBB+SQRT(BBB**2+FOURAC))*0.5    NDLPTS  35
```

```
      YA=DELTI*SQRT((1.+DELTO)-(1.-XA)**2)/SQRT(1.+DELTO)
      GO TO 20                                                          NDLPTS  36
   12 THET=ATAN2(RK(J-1),XX(J-1))                                       NDLPTS  37
   15 CONTINUE                                                          NDLPTS  38
      DELT=(1.0+0.68*THET**2+0.16*THET**4)*DELTO                        NDLPTS  39
      XA=1.0-(1.0+DELT)*COS(THET)                                       NDLPTS  40
      YA=(1.0+DELT)*SIN(THET)                                           NDLPTS  41
   20 CONTINUE                                                          NDLPTS  42
C...BODY SHAPE                                                          NDLPTS  43
      IF(NUBODY) GO TO 30                                               NDLPTS  44
      CALL BODY(YB,XB,RNOSE,THET,HORN)                                  NDLPTS  45
      GO TO 40                                                          NDLPTS  46
   30 XB=1.0-XX(J-1)                                                    NDLPTS  47
      YB=RK(J-1)                                                        NDLPTS  48
   40 CONTINUE                                                          NDLPTS  49
      ZKM=KMAX-1                                                        NDLPTS  50
      DX=(XA-XB)/ZKM                                                    NDLPTS  51
      DY=(YA-YB)/ZKM                                                    NDLPTS  52
      DO 4 K=1,KMAX                                                     NDLPTS  53
      ZK=K-1                                                            NDLPTS  54
      X(J,K)=XB+ZK*DX                                                   NDLPTS  55
      Y(J,K)=YB+ZK*DY                                                   NDLPTS  56
    4 CONTINUE                                                          NDLPTS  57
      DO 5 K=1,KMAX                                                     NDLPTS  58
      X(1,K)=X(2,K)                                                     NDLPTS  59
      Y(1,K)=-Y(2,K)                                                    NDLPTS  60
    5 CONTINUE                                                          NDLPTS  61
      RETURN                                                            NDLPTS  62
      END                                                               NDLPTS  63
                                                                        NDLPTS  64

      SUBROUTINE OUTPUT(L)                                              OUTPUT   2
      COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT, COM1     2
     > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTORT,PINF,   COM1     3
     <RINF,QINF,CINF,JCS,TM,CLUS,PT,HORN,RNOSE,NCASE,NPUNCH             COM1     4
      COMMON /COM2/ X(25,20),Y(25,20),XET(25,20,2),XEX(25,20,2),        COM2     2
     *  XEY(25,20,2),D(25,20)                                          COM2     3
      COMMON /COM3/ Q(25,20,4),EF(25,4),S(25,20,4),G(4),AB(4,4)         COM3     2
      DIMENSION SL(25),CON(8)                                           OUTPUT   6
      GO TO (1,2,3,4,5),L                                               OUTPUT   7
    1 CONTINUE                                                          OUTPUT   8
C...OUTPUT FLOWFIELD DATA                                               OUTPUT   9
      RMS=0.0                                                           OUTPUT  10
      PERRMX=0.0                                                        OUTPUT  11
      HTINF=GAM/GAMM1*PINF/RINF+QINF**2*0.5                             OUTPUT  12
      WRITE(6,111)                                                      OUTPUT  13
      DO 10 K=1,KMAX                                                    OUTPUT  14
      DO 6 J=1,JMAX                                                     OUTPUT  15
      EN=Q(J,K,4)*D(J,K)                                                OUTPUT  16
      RHO=Q(J,K,1)*D(J,K)                                               OUTPUT  17
      U=Q(J,K,2)/Q(J,K,1)                                               OUTPUT  18
      V=Q(J,K,3)/Q(J,K,1)                                               OUTPUT  19
      P=GAMM1*(EN-RHO*0.5*(U*U+V*V))                                    OUTPUT  20
      HT=GAM/GAMM1*P/RHO+0.5*(U*U+V*V)                                  OUTPUT  21
      SS=SQRT(GAM*P/RHO)                                                OUTPUT  22
      PERR=ABS(HT-HTINF)*100.0/HTINF                                    OUTPUT  23
      IF(PERR.GT.PERRMX) PERRMX=PERR                                    OUTPUT  24
      RMS=RMS+PERR**2                                                   OUTPUT  25
      SL(J)=SQRT(U*U+V*V)/SS                                            OUTPUT  26
    6 CONTINUE                                                          OUTPUT  27
      DO 11 J=3,JMAX                                                    OUTPUT  28
      IF((SL(J).LE.1.0.AND.SL(J-1).GE.1.0).OR.(SL(J).GE.1.0.AND.SL(J-1) OUTPUT  29
     <.LE.1.0))GO TO 12                                                 OUTPUT  30
      GO TO 11                                                          OUTPUT  31
   12 JSL=J                                                             OUTPUT  32
      JSLM=JSL-1                                                        OUTPUT  33
      COEF=(1.0-SL(JSLM))/(SL(JSL)-SL(JSLM))                            OUTPUT  34
      XSL=X(JSLM,K)+COEF*(X(JSL,K)-X(JSLM,K))                           OUTPUT  35
      YSL=Y(JSLM,K)+COEF*(Y(JSL,K)-Y(JSLM,K))                           OUTPUT  36
      XSL=1.0-XSL                                                       OUTPUT  37
      WRITE(6,110) XSL,YSL                                              OUTPUT  38
   11 CONTINUE                                                          OUTPUT  39
   10 CONTINUE                                                          OUTPUT  40
      RMS=SQRT(RMS/FLOAT(JMAX*KMAX))                                    OUTPUT  41
      WRITE(6,107) PERRMX,RMS                                           OUTPUT  42
      RETURN                                                            OUTPUT  43
    2 CONTINUE                                                          OUTPUT  44
C...OUTPUT E AND F CONSERVATIVE VARIABLES                               OUTPUT  45
      WRITE(6,103)                                                      OUTPUT  46
      DO 7 K=1,KMAX                                                     OUTPUT  47
      WRITE(6,104) K                                                    OUTPUT  48
      DO 7 J=1,JMAX                                                     OUTPUT  49
      CALL EFCON(J,K,1)                                                 OUTPUT  50
      DO 8 N=1,4                                                        OUTPUT  51
    8 CON(N)=G(N)                                                       OUTPUT  52
      CALL EFCON(J,K,2)                                                 OUTPUT  53
      DO 9 N=1,4                                                        OUTPUT  54
      NN=N+4                                                            OUTPUT  55
    9 CON(NN)=G(N)                                                      OUTPUT  56
    7 WRITE(6,105) J,(CON(N),N=1,8)                                     OUTPUT  57
      RETURN                                                            OUTPUT  58
    3 CONTINUE                                                          OUTPUT  59
      RETURN                                                            OUTPUT  60
    4 CONTINUE                                                          OUTPUT  61
      WRITE(6,106)                                                      OUTPUT  62
      WRITE(6,109) ((J,K,X(J,K),Y(J,K),(XET(J,K,I),XEX(J,K,I),XEY(J,K,I) OUTPUT  63
     > ,I=1,2),D(J,K),K=1,KMAX),J=1,JMAX)                               OUTPUT  64
      RETURN                                                            OUTPUT  65
    5 CONTINUE                                                          OUTPUT  66
      RETURN                                                            OUTPUT  67
  103 FORMAT(1H0,37X,32H<<<< CONSERVATIVE VARIABLES >>>>)               OUTPUT  68
  104 FORMAT(3H0K=,I2//4X,1HJ,6X,2HE1,10X,2HE2,10X,2HE3,10X,2HE4,10X,  OUTPUT  69
     > 2HF1,10X,2HF2,10X,2HF3,10X,2HF4/)                                OUTPUT  70
  105 FORMAT(I5,8E12.4)                                                 OUTPUT  71
  107 FORMAT(21H0<<<< % ERROR IN HT =,E12.4,3X,22HRMS OF % ERROR IN HT = OUTPUT  72
     > ,E12.4,5H >>>>)                                                  OUTPUT  73
  106 FORMAT(1H1,3X,1HJ,4X,1HK,8X,1HX,11X,1HY,10X,4HXI-T,6X,4HXI-X,8X,  OUTPUT  74
     > 4HXI-Y,7X,5HETA-T,7X,5HETA-X,7X,5HETA-Y,8X,1HJ/)                 OUTPUT  75
  109 FORMAT(2I5,9F12.6)                                                OUTPUT  76
  110 FORMAT(5H,XSL=,F7.4,3X,4HRSL=,F7.4)                               OUTPUT  77
  111 FORMAT(//20H FINAL SONIC LINE LOCATION/)                          OUTPUT  78
      END                                                               OUTPUT  79

      SUBROUTINE RMS                                                    RMS      2
      COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT, COM1     2
     > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTORT,PINF,   COM1     3
     <RINF,QINF,CINF,JCS,TM,CLUS,PT,HORN,RNOSE,NCASE,NPUNCH             COM1     4
      COMMON /COM3/ Q(25,20,4),EF(25,4),S(25,20,4),G(4),AB(4,4)         COM3     2
C...THIS SUBROUTINE COMPUTES THE RIGHT HAND SIDE OF THE DELTA FORM      RMS      5
C...EQUATION                                                            RMS      6
C...FORM E CONSERVATIVE VARIABLES AND DIFFERENCE. STORE IN THE S ARRAY  RMS      7
      DO 1 K=2,KM                                                       RMS      8
      DO 2 J=1,JMAX                                                     RMS      9
      CALL EFCON(J,K,1)                                                 RMS     10
      DO 2 N=1,4                                                        RMS     11
    2 EF(J,N)=G(N)                                                      RMS     12
C...CENTRAL DIFFERENCE E CONSERVATIVE VARIABLE                          RMS     13
      DO 1 N=1,4                                                        RMS     14
      DO 1 J=2,JM                                                       RMS     15
    1 S(J,K,N)=(EF(J+1,N)-EF(J-1,N))*H                                  RMS     16
C...FORM F CONSERVATIVE VARIABLES AND DIFFERENCE. ADD TO PREVIOUS S     RMS     17
C...ARRAY                                                               RMS     18
      DO 3 J=2,JM                                                       RMS     19
      DO 4 K=1,KMAX                                                     RMS     20
      CALL EFCON(J,K,2)                                                 RMS     21
      DO 4 N=1,4                                                        RMS     22
    4 EF(K,N)=G(N)                                                      RMS     23
C...CENTRAL DIFFERENCE F CONSERVATIVE VARIABLE                          RMS     24
      DO 3 N=1,4                                                        RMS     25
      DO 3 K=2,KM                                                       RMS     26
      S(J,K,N)=S(J,K,N)-(EF(K+1,N)-EF(K-1,N))*H                         RMS     27
    3 CONTINUE                                                          RMS     28
      RETURN                                                            RMS     29
      END                                                               RMS     30

      SUBROUTINE SHOCK                                                  SHOCK    2
      COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,DT,SMU,IPRT, COM1     2
     > CHORD,NCA,NCB,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTORT,PINF,   COM1     3
     <RINF,QINF,CINF,JCS,TM,CLUS,PT,HORN,RNOSE,NCASE,NPUNCH             COM1     4
      COMMON /COM2/ X(25,20),Y(25,20),XET(25,20,2),XEX(25,20,2),        COM2     2
     *  XEY(25,20,2),D(25,20)                                          COM2     3
      COMMON /COM3/ Q(25,20,4),EF(25,4),S(25,20,4),G(4),AB(4,4)         COM3     2
      DIMENSION P(25,3),PXI(25),PETA(25),U(25,3),UXI(25),UETA(25),      SHOCK    6
     *  V(25,3),VXI(25),VETA(25),R(25,3)                               SHOCK    7
C...COMPUTE THE FLOW VARIABLES ONE MESH INTERVAL BELOW SHOCK            SHOCK    8
      RMS=0.0                                                           SHOCK    9
      QSEM=0.0                                                          SHOCK   10
      JMM=JMAX-2                                                        SHOCK   11
      ZKM=KMAX-1                                                        SHOCK   12
      DO 3 K=1,3                                                        SHOCK   13
      DO 3 J=1,JMAX                                                     SHOCK   14
      KK=KMAX-3+K                                                       SHOCK   15
      Z=1.0/Q(J,KK,1)                                                   SHOCK   16
      R(J,K)=Q(J,KK,1)*D(J,KK)                                          SHOCK   17
      U(J,K)=Q(J,KK,2)*Z                                                SHOCK   18
      V(J,K)=Q(J,KK,3)*Z                                                SHOCK   19
```

```
      E2=0(J,KK,4)*D(J,KK)
    3 P(J,K)=(E2-0.5*R(J,K)*(U(J,K)**2+V(J,K)**2))*GAMM1
C...COMPUTE P-XI, U-XI, P-ETA, U-ETA, AND V-ETA DERIVATIVES
      DO 4 J=2,JM
      PXI(J)=(P(J+1,3)-P(J-1,3))*0.5
      UXI(J)=(U(J+1,3)-U(J-1,3))*0.5
    4 VXI(J)=(V(J+1,3)-V(J-1,3))*0.5
      PXI(1)=-PXI(2)
      UXI(1)=-UXI(2)
      VXI(1)=VXI(2)
      PXI(JMAX)=(3.0*P(JMAX,3)-4.0*P(JM,3)+P(JMM,3))*0.5
      UXI(JMAX)=(3.0*U(JMAX,3)-4.0*U(JM,3)+U(JMM,3))*0.5
      VXI(JMAX)=(3.0*V(JMAX,3)-4.0*V(JM,3)+V(JMM,3))*0.5
      DO 5 J=1,JMAX
      PETA(J)=(3.0*P(J,3)-4.0*P(J,2)+P(J,1))*0.5
      UETA(J)=(3.0*U(J,3)-4.0*U(J,2)+U(J,1))*0.5
      VETA(J)=(3.0*V(J,3)-4.0*V(J,2)+V(J,1))*0.5
      IF(J.EQ.1.OR.J.EQ.JMAX) GO TO 5
      P(J,2)=P(J+1,3)+P(J-1,3)-2.0*P(J,3)
    5 CONTINUE
      P(1,2)=P(2,2)
      P(JMAX,2)=0.0
      DO 1 J=1,JMAX
      K=KMAX
C...DETERMINE SHOCK ANGLE DELTA=ARCTAN(-ETAY/ETAX) J,KMAX
      DELTA=ATAN(-XEY(J,K,2)/XEX(J,K,2))
      SD=SIN(DELTA)
      CD=COS(DELTA)
      U1T=QINF*CD
      UBAR=XETJ,K,1)*U(J,3)+XEX(J,K,1)*V(J,3)+XEY(J,K,1)
      VBAR=XETJ,K,2)*U(J,3)+XEX(J,K,2)*V(J,3)+XEY(J,K,2)
      RCS=GAM*P(J,3)
      PTAU=-UBAR*PXI(J)-VBAR*PETA(J)-RCS*(UXI(J)*XEX(J,K,1)+
     > VXI(J)*XEY(J,K,1)+UETA(J)*XEX(J,K,2)+VETA(J)*XEY(J,K,2)+
     > (V(J,3)/Y(J,K))*FLOAT(JCS))
      P2=P(J,3)+PTAU*DI*0.2+0.1*P(J,2)
      IF(J .EQ. JMAX)P2=2.*P(JM,3)-P(JM-1,3)
      IF(P2.LE.0.0) GO TO 6
      Z=GAM+1.0
      XMX=SQRT(2.5/GAM*(P2/PINF*Z+GAMM1))
      QS=QINF*XMX-U1T
      PB=P(J,3)
      RB=R(J,3)
      UB=U(J,3)
      VB=V(J,3)
      EB=PB/GAMM1+0.5*RB*(UB**2+VB**2)
      U2T=2.0*(1.0-XMX**2)*QINF/((GAM+1.0)*XMX)+U1T
      R2=RINF*(P2/PINF+GAMM1/Z)/(1.0+GAMM1/Z*P2/PINF)
      U2=QINF*SD**2+U2T*CD
      V2=QINF*SD*CD-U2T*SD
      E2=P2/GAMM1+0.5*R2*(U2**2+V2**2)
C...COMPUTE CONSERVATIVE VARIABLES AT SHOCK
      K=KMAX
      DI=1.0/D(J,K)
      Q(J,K,1)=R2*DI
      Q(J,K,2)=R2*U2*DI
      Q(J,K,3)=R2*V2*DI
      Q(J,K,4)=E2*DI
C...DETERMINE ANGLE OF XI=CONST LINE WITH X-AXIS
      K=KMAX
      IF(ABS(XEY(J,K,1))-0.0000001) 7,7,8
    7 THETA=1.57079633
      GO TO 9
    8 CONTINUE
      THETA=ATAN(XEX(J,K,1)/XEY(J,K,1))
    9 CONTINUE
C...COMPUTE SHOCK SPEED IN X AND Y DIRECTIONS
      BETA=THETA-DELTA
      QSE=QS/COS(BETA)
      IF(ABS(QSE) .GE. ABS(QSEM))JQS=J
      IF(ABS(QSE) .GE. ABS(QSEM))QSEM=QSE
      RMS=RMS+QSE**2
      XST=-QSE*COS(THETA)
      YST=QSE*SIN(THETA)
      THETA=THETA*57.29578
      DELTA=DELTA*57.29578
      BETA=BETA*57.29578
C...PROPAGATE SHOCK
      X(J,K)=X(J,K)+XST*DT
      Y(J,K)=Y(J,K)+YST*DT
C...ADJUST OTHER GRID POINTS
      DO 2 K=2,KM
      ZKFAC=FLOAT(K-1)/ZKM
      X(J,K)=(X(J,KMAX)-X(J,1))*ZKFAC+X(J,1)
      Y(J,K)=(Y(J,KMAX)-Y(J,1))*ZKFAC+Y(J,1)
    2 CONTINUE
    1 CONTINUE
      RMS=SQRT(RMS/FLOAT(JMAX))
      WRITE(6,102) IT,RMS,QSEM,JQS
      RETURN
    5 CONTINUE
      WRITE(6,103) J,P2,P(J,3),PTAU
      STOP
  102 FORMAT(10H ITERATION,I4,4X,19HRMS OF SHOCK SPEED=,1PE11.4,3X,
     *  20HMAXIMUM SHOCK SPEED=,E11.4,6H AT J=,I2)
  103 FORMAT(1HC,41HNEGATIVE PRESSURE DETECTED BY SHOCK AT J=,I2/
     *   3X,3HPN=,1PE10.3,3X,5HPO=,3X,E10.3,3X,5HPTAU=,E10.3)
      END


      SUBROUTINE XIETAD (IFLAG)
      COMMON/COM1/JMAX,KMAX,JM,KM,XMACH,ALPHA,GAM,GAMM1,CN,UT,SMU,IPRT,
     > CHORD,NCA,NCG,NCC,AA,H,OMEGA,NU,NL,IT,TAU,ITER,ENT,PTURT,PINF,
     <RINF,QINF,UINF,JCS,TM,CLOS,PT,HORN,RNDSE,NCASE,NPUNCH
      COMMON /COM2/ X(25,20),Y(25,20),XET(25,20,2),XEX(25,20,2),
     *  XEY(25,20,2),D(25,20)
      COMMON /COM3/ Q(25,20,4),SF(25,4),S(25,20,4),G(4),AB(4,4)
      DO 11 K=1,KMAX
      DO 11 J=1,JMAX
      XET(J,K,1)=0.0
      XET(J,K,2)=0.0
   11 CONTINUE
      JMM=JM-1
      KMM=KM-1
C...COMPUTE X-XI AND Y-XI; DXI AND DETA = 1
      DO 1 K=1,KMAX
      DO 2 J=2,JM
      XEY(J,K,2)=(X(J+1,K)-X(J-1,K))*0.5
    2 XEX(J,K,2)=(Y(J+1,K)-Y(J-1,K))*0.5
      XEY(1,K,2)=(-3.0*X(1,K)+4.0*X(2,K)-X(3,K))*0.5
      XEY(JMAX,K,2)=(3.0*X(JMAX,K)-4.0*X(JM,K)+X(JMM,K))*0.5
      XEX(1,K,2)=(-3.0*Y(1,K)+4.0*Y(2,K)-Y(3,K))*0.5
    1 XEX(JMAX,K,2)=(3.0*Y(JMAX,K)-4.0*Y(JM,K)+Y(JMM,K))*0.5
C...COMPUTE X-ETA AND Y-ETA
      DO 3 J=1,JMAX
      DO 4 K=2,KM
      XEY(J,K,1)=(X(J,K+1)-X(J,K-1))*0.5
    4 XEX(J,K,1)=(Y(J,K+1)-Y(J,K-1))*0.5
      XEY(J,1,1)=(-3.0*X(J,1)+4.0*X(J,2)-X(J,3))*0.5
      XEY(J,KMAX,1)=(3.0*X(J,KMAX)-4.0*X(J,KM)+X(J,KMM))*0.5
      XEX(J,1,1)=(-3.0*Y(J,1)+4.0*Y(J,2)-Y(J,3))*0.5
    3 XEX(J,KMAX,1)=(3.0*Y(J,KMAX)-4.0*Y(J,KM)+Y(J,KMM))*0.5
C...COMPUTE XI-X, XI-Y, ETA-X, AND ETA-Y
      DO 5 K=1,KMAX
      DO 5 J=1,JMAX
      DI=1.0/(XEX(J,K,1)*XEY(J,K,2)-XEY(J,K,1)*XEX(J,K,2))
      DII=DI
      IF(IFLAG.EQ.0) GO TO 7
C...ADJUST CONSERVATIVE VARIABLES BASED ON NEW MESH
      DO 6 N=1,4
    6 Q(J,K,N)=Q(J,K,N)*D(J,K)/DII
    7 CONTINUE
C...THE GEOMETRIC JACOBIAN IS DEFINED HERE AND STORED IN THE D ARRAY
      D(J,K)=DII
      XEX(J,K,1)=XEX(J,K,1)*DI
      XEY(J,K,1)=-XEY(J,K,1)*DI
      XEX(J,K,2)=-XEX(J,K,2)*DI
    5 XEY(J,K,2)=XEY(J,K,2)*DI
C...REFLECT METRICS AND DEPENDENT VARIABLES ABOUT PLANE OF SYMMETRY
      IF(IFLAG.EQ.0) GO TO 8
      DO 9 K=1,KMAX
      DD=(+1.0)**JCS
      D(1,K)=D(2,K)*DD
      XEX(1,K,1)=-XEX(2,K,1)
      XEY(1,K,1)=XEY(2,K,1)
      XEX(1,K,2)=XEX(2,K,2)
      XEY(1,K,2)=-XEY(2,K,2)
      DO 10 N=1,4
   10 Q(1,K,N)=Q(2,K,N)*DD
    9 Q(1,K,3)=-Q(2,K,3)*DD
    8 CONTINUE
      RETURN
      END


      SUBROUTINE MARCH
      LEVEL 2,ETEMP,EQ,FQ,GQ,HQ
      COMMON/LAKGE/ETEMP(4,24,41),EQ(4,24,41),
     *       FQ(4,24,41)    , GQ(4,24,41) , HQ(4,24,41)
```

```
      LEVEL 2, RHO,P,U,V,W,ROB,ROBZ,VINF,WINF,ROBPH,RB,RBZ,RBPH,DTDPH,
     *  BCT,DTDZ,DTDR,ACT,ICONST,GAM,CONST,NREGON,RS,KSZ,RSPHI,RST,RSZT,
     *  RSPHIT
      COMMON /PVARB/RHO(24,41) , P(24,41) , U(24,41) , V(24,41) , W(24,4
     *1) ,
     *       POB(41)    , POBZ(41) , VINF(41) , WINF(41) ,
     *       ROBPH(41)  ,                RB(41)   , RBZ(41)  , RBPH(41) ,
     *      DTDPH(24,41), BCT(41)  ,DTDZ(24,41),DTDR(41) , ACT(41)  ,
     *       ICONST(50) , GAM(20)  , CONST(50) ,NREGON   , RS(41)   ,
     *       KSZ(41)    , RSPHI(41), RST(41)   , RSZT(41), RSPHIT(41)
      COMMON /IDVARB/RK,ETA(41),PHIP(41),DTIL(41),DTILE(41),DETA,TP(24)
      COMMON/SVARB/T,Z      , PHI   , DT    , DZ    , DPHI  , ZINT  ,
     *       ZEND   , PI    , ALPHA , GAMMA , SIGMA , XMACH , TAPE1 ,
     *       TAPE2  , DISK1 , ALPH  , DISK2 , SIGM  , NPRNT , DZDT  ,
     *       DZDPH  , ZM    , TMWD  , TMLD  , TMW   , TML   , TIMW  ,
     *       TTML   , RZ    , BZ    , NIPHI , NIT   , KPHI  , NITER ,
     *       NPHI   , NPHI1 , NPHI2 , NPHI3 , NPHM1 , NPHM2 , NPHM3 ,
     *       NT     , NT1   , NT2   , NT3   , PHIFD , NCONE , RADI  ,
     *       PHIF   , METHOD, LAG   , NBC   , PINF  , RHOIN , UINF  ,
     *OINF,GASCON,NREAL,NPUNCH
      INTEGER DISK1,DISK2,TAPE1,TAPE2
      COMMON /L1800/NTEST(11),NCRTA,NOSXY(4),NCRTB(3),NSWCH(32),IN(32)
      COMMON/XYZ/NXYAY(3),X1(160),X2(160),X3(160),Y1(160),Y2(160),Y3(160
     *),Z4(160),Z2(160),Z3(160)
      COMMON/JOE/ZL1,CF1,CF2,ZLF,ZTRAN,DZTRAN
      COMMON/EPSL/EPSL1,EPSL2
      COMMON/ENTRO/S(41),ZBS,ZFLD,ITPRTB,ITPRTF,NCASE,NTOSUS
      COMMON/CFS/NCFS
      COMMON/CLUSTR/XJ,XI(24),TXI(24),TXIT(24)
      CALL SFTDAT
      CALL GEOM3(0,PHIP,NPHI,Z,RB,RBZ,RBPH,IPRNT)
      SIGMA=ATAN(CF2)*57.29578
      ICONST(48)=0
      CALL INITA
      STPSZE=NT-1
      CALL BNDRYM(2)
      DO 4 JUDI=1,NITER
      ICONST(5)=JUDI
C.........COMPUTE AUTOMATIC STEPSIZE
      IF (JUDI .EQ. 1) GO TO 3
      IF (MOD(JUDI,ICONST(49)).NE.0) GO TO 5
    3 CALL EIGENM
      IF (DZ .LT. 1.0) GO TO 5
      DZDT=STPSZE
      DZ=DZDT*DT
      DZDPH=DZ/DLTA
    5 CONTINUE
      CALL DIFFR
C.........WRITE DATA FOR RERUN ON TAPE9
C        AND STORE IN ARRAYS USED BY CONTOUR ROUTINES
      CALL OUTPTM
      IF (Z .GT. ZEND) GO TO 19
    4 CONTINUE
   19 CONTINUE
C.........ALL DATA REQUIRED FOR RERUN OF THIS CASE IS NOW ON TAPE9
      END FILE 9
      RETURN
      END
```

```
      SUBROUTINE BNDRYM(K1)
      LEVEL 2,ETEMP,EO,FO,GO,HO
      COMMON/LARGE/ETEMP(4,24,41),EO(4,24,41),
     *       FO(4,24,41)    , GO(4,24,41) , HO(4,24,41)
      LEVEL 2, RHO,P,U,V,W,ROB,ROBZ,VINF,WINF,ROBPH,RB,RBZ,RBPH,DTDPH,
     *  BCT,DTDZ,DTDR,ACT,ICONST,GAM,CONST,NREGON,RS,KSZ,RSPHI,RST,RSZT,
     *  RSPHIT
      COMMON /PVARB/RHO(24,41) , P(24,41) , U(24,41) , V(24,41) , W(24,4
     *1) ,
     *       ROB(41)    , ROBZ(41) , VINF(41) , WINF(41) ,
     *       ROBPH(41)  ,                RB(41)   , RBZ(41)  , RBPH(41) ,
     *      DTDPH(24,41), BCT(41)  ,DTDZ(24,41),DTDR(41) , ACT(41)  ,
     *       ICONST(50) , GAM(20)  , CONST(50) ,NREGON   , RS(41)   ,
     *       KSZ(41)    , RSPHI(41), RST(41)   , RSZT(41), RSPHIT(41)
      COMMON /IDVARB/RK,ETA(41),PHIP(41),DTIL(41),DTILE(41),DETA,TP(24)
      COMMON/SVARB/T,Z      , PHI   , DT    , DZ    , DPHI  , ZINT  ,
     *       ZEND   , PI    , ALPHA , GAMMA , SIGMA , XMACH , TAPE1 ,
     *       TAPE2  , DISK1 , ALPH  , DISK2 , SIGM  , NPRNT , DZDT  ,
     *       DZDPH  , ZM    , TMWD  , TMLD  , TMW   , TML   , TTMW  ,
     *       TTML   , RZ    , BZ    , NIPHI , NIT   , KPHI  , NITER ,
     *       NPHI   , NPHI1 , NPHI2 , NPHI3 , NPHM1 , NPHM2 , NPHM3 ,
     *       NT     , NT1   , NT2   , NT3   , PHIFD , NCONE , RADI  ,
     *       PHIF   , METHOD, LAG   , NBC   , PINF  , RHOIN , UINF  ,
     *OINF,GASCON,NREAL,NPUNCH
      INTEGER DISK1,DISK2,TAPE1,TAPE2
      COMMON /L1800/NTEST(11),NCRTA,NOSXY(4),NCRTB(3),NSWCH(32),IN(32)
      COMMON/XYZ/NXYAY(3),X1(160),X2(160),X3(160),Y1(160),Y2(160),Y3(160
     *),Z4(160),Z2(160),Z3(160)
      COMMON/ENTRO/S(41),ZBS,ZFLD,ITPRTB,ITPRTF,NCASE,NTOSUS
      DIMENSION PK13(41),PK14(41),PK21(41),PK22(41),PK23(41)
      LOGICAL IEND
      ARSIN(A)=ASIN(A)
      GO TO (10,15,11),K1
   10 CONTINUE
C..WEAK OR SMALL ANGLE CORRECTIONS (USES PRANDTL-MEYER RELATIONS)
      DO 9 K=3,NPHI
      PK4=1.0/SQRT(PHIZ(K)**2+1.0*(RBPH(K)/RB(K))**2)
      PK1=-RBZ(K)*PK4
      PK2=PK4
      PK3=-RBPH(K)/RB(K)*PK4
      IT2ND=.FALSE.
      QSQ=U(3,K)**2+V(3,K)**2+W(3,K)**2
      IF(P(3,K).GE.0.) GO TO 4
C..NEGATIVE SURFACE PRESSURE
      ICHECK=1
      X=1.0-Z
      IF (K .EQ. 3) WRITE(6,100) X,P(3,K),RHO(3,K),U(3,K),V(3,K)
      P(3,K)=PINF*(1.-0.5*GAMMA)
      RHO(3,K)=(P(3,K)/S(K))**(1.0/GAMMA)
      Q2K=SQRT(1.0-P(3,K)/RHO(3,K))
      U(3,K)=U(3,K)*Q3K/QSQ**0.5
      V(3,K)=V(3,K)*Q3K/QSQ**0.5
      W(3,K)=W(3,K)*Q3K/QSQ**0.5
      QSQ=Q3K**2
    4 CONTINUE
      IF(RHO(3,K) .GE. 0.01)GO TO 5
      ICHECK=2
      X=1.0-Z
      IF (K .EQ. 3) WRITE(6,100) X,P(3,K),RHO(3,K),U(3,K),V(3,K)
      RHO(3,K)=(P(3,K)/S(K))**(1.0/GAMMA)
      Q3K=SQRT(1.0-P(3,K)/RHO(3,K))
      U(3,K)=U(3,K)*Q3K/QSQ**0.5
      V(3,K)=V(3,K)*Q3K/QSQ**0.5
      W(3,K)=W(3,K)*Q3K/QSQ**0.5
    5 CONTINUE
      PK5=SQRT(QSQ)
      PK6=(PK1*U(3,K)+PK2*V(3,K)+PK3*W(3,K))/PK5
      PK7=ARSIN(PK6)
      PK8=GAM(1)*P(3,K)/RHO(3,K)
      PK9=PK8**2/PK6
      PK10=PK9-1.0
      IF(PK10 .GT. 0.0)GO TO 6
      ICHECK=3
      X=1.0-Z
      IF (K .EQ. 3) WRITE(6,100) X,P(3,K),RHO(3,K),U(3,K),V(3,K)
      PK10=0.5
      PK9=1.5
      PK8=PK9**2/PK9
      RHO(3,K)=GAM(1)*P(3,K)/PK8
      Q2K=SQRT(1.0-P(3,K)/RHO(3,K))
      U(3,K)=U(3,K)*Q3K/QSQ**0.5
      V(3,K)=V(3,K)*Q3K/QSQ**0.5
      W(3,K)=W(3,K)*Q3K/QSQ**0.5
    6 CONTINUE
      PK11=GAMMA*PK9/SQRT(PK10)
      PK12=GAMMA*PK9*((GAMMA+1.0)*PK9**2-4.0*PK10)/(4.0*PK10**2)
C..NEXT TWO TERMS COMPUTE COEFFICIENTS FOR DENSITY EXPANSION
      IF(NSWCH(5).EQ.0) GO TO 2
      PK25=PK21/GAMMA
      PK26=PK9*((3.0-GAMMA)*PK9*(PK9-2.0)+4.0)/(4.0*PK10**2)
    2 CONTINUE
      PK13(K)=P(3,K)*(1.0-PK11*PK7+PK12*PK7**2)
      FACTOR=0.5*GAMMA*PK9/(PK10**3.5)
      TERM1=(GAMMA+2.0)*PK9**4/6.0
      TERM2=-(5.0+7.0*GAMMA-2.0*GAMMA**2)*PK9**3/6.0
      TERM3= 5.0*(GAMMA+1.0)*PK9**2/3.0
      TERM4=4.0/3.0-2.0*PK9
      COEFF3=FACTOR*(TERM1+TERM2+TERM3+TERM4)
      PTEST=PK13(K)-P(3,K)*COEFF3*PK7**3
      IF (ABS(PK7).LT.ABS((.01))GO TO 123
      XM1 = SQRT(PK9)
      CALL PMTURN(XM1,PK7,P2P1,NITS,GAMMA)
      PTRUE = P(3,K)*P2P1
      PTEST = PTRUE
  123 CONTINUE
      PK13(K)=PTEST
      PK14(K)=(PK13(K)/S(K))**(1.0/GAMMA)
      IF(NSWCH(5).EQ.0) GO TO 3
      PK14(K)=PHL(3,K)*(1.0-PK25*PK7+PK26*PK7**2)
      S(K)=P(3,K)/RHO(3,K)**GAMMA
```

```
3     CONTINUE                                                          BNDRYM  91
      PK15=SQRT(1.0-PK13(K)/PK14(K))                                    BNDRYM  92
      PK16=PK6*PK5*PK4                                                  BNDRYM  93
      PK17=U(3,K)+PK15*RBZ(K)                                           BNDRYM  94
      PK18=V(3,K)-PK16                                                  BNDRYM  95
      PK19=W(3,K)+PK16*RBPH(K)/RB(K)                                    BNDRYM  96
      PK20=SQRT(PK17**2+PK18**2+PK19**2)                                BNDRYM  97
      PK24=PK15/PK20                                                    BNDRYM  98
      PK21(K)=PK24*PK17                                                 BNDRYM  99
      PK22(K)=PK24*PK18                                                 BNDRYM 100
      PK23(K)=PK24*PK19                                                 BNDRYM 101
4     CONTINUE                                                          BNDRYM 102
11    CONTINUE                                                          BNDRYM 103
C..RESET BODY VARIABLES TO THOSE CALCULATED BY ABBETTS SCHEME           BNDRYM 104
      DO 12 K=3,NPHI                                                    BNDRYM 105
      P(3,K)=PK13(K)                                                    BNDRYM 106
      RHO(3,K)=PK14(K)                                                  BNDRYM 107
      U(3,K)=PK21(K)                                                    BNDRYM 108
      V(3,K)=PK22(K)                                                    BNDRYM 109
      W(3,K)=PK23(K)                                                    BNDRYM 110
12    CONTINUE                                                          BNDRYM 111
      GO TO 21                                                          BNDRYM 112
18    CONTINUE                                                          BNDRYM 113
C                                                                       BNDRYM 114
C..APPLY REFLECTION PRINCIPLE AT PLANES OF SYMMETRY                     BNDRYM 115
C                                                                       BNDRYM 116
      DO 1 K=1,2                                                        BNDRYM 117
      M=6-K                                                             BNDRYM 118
      L=NPHI+K                                                          BNDRYM 119
      N=NPHI-K                                                          BNDRYM 120
      DO 1 J=3,NT2                                                      BNDRYM 121
      RHO(J,K)=RHO(J,M)                                                 BNDRYM 122
      RHO(J,L)=RHO(J,N)                                                 BNDRYM 123
      P(J,K)=P(J,M)                                                     BNDRYM 124
      P(J,L)=P(J,N)                                                     BNDRYM 125
      U(J,K)=U(J,M)                                                     BNDRYM 126
      U(J,L)=U(J,N)                                                     BNDRYM 127
      V(J,K)=V(J,M)                                                     BNDRYM 128
      V(J,L)=V(J,N)                                                     BNDRYM 129
      W(J,K)=-W(J,M)                                                    BNDRYM 130
      W(J,L)=-W(J,N)                                                    BNDRYM 131
      W(J,3)=0.0                                                        BNDRYM 132
      W(J,NPHI)=0.0                                                     BNDRYM 133
1     CONTINUE                                                          BNDRYM 134
21    CONTINUE                                                          BNDRYM 135
  100 FORMAT(1HO,49HNEGATIVE PRESSURE OR DENSITY ON BODY DETECTED BY ,  BNDRYM 136
     *   12H8NDRY AT K=,F7.3/3X,3HP8=,1PE10.3,3X,5HRHO8=,E10.3,3X,      BNDRYM 137
     *   4HV8=,E10.3,3X,4HW8=,E10.3)                                    BNDRYM 138
      RETURN                                                            BNDRYM 139
      END                                                               BNDRYM 140


      SUBROUTINE DIFFR                                                  DIFFR   2
      LEVEL 2,ETEMP,E0,F0,G0,H0                                         CVARB   2
      COMMON/LARGE/ETEMP(4,24,41),E0(4,24,41),                          CVARB   3
     *       F0(4,24,41)     , G0(4,24,41) , H0(4,24,41)                CVARB   4
      LEVEL 2, RHO,P,U,V,W,RO8,RO8Z,V1NF,W1NF,RO8PH,RB,RBZ,RBPH,DTDPH,  PVARB   2
     *  BCT,DTDZ,DTDR,ACT,ICONST,GAM,CONST,NREGON,RS,RSZ,RSPHI,RST,RSZT,  PVARB   3
     *  RSPHIT                                                         PVARB   4
      COMMON /PVARB/RHO(24,41) , P(24,41) , U(24,41) , V(24,41) , W(24,4PVARB   5
     *1) ,                                                            PVARB   6
     *       RO8(41)     , RO8Z(41) , V1NF(41) , W1NF(41) ,           PVARB   7
     *       RO8PH(41)   ,            RB(41)   , RBZ(41)  , RBPH(41) , PVARB   8
     *      DTDPH(24,41), BCT(41)  ,DTDZ(24,41),DTDR(41) , ACT(41)  , PVARB   9
     *       ICONST(50) , GAM(20)  , CONST(50) ,NREGON    , RS(41)   , PVARB  10
     *       RSZ(41)    , RSPHI(41), RST(41)   , RSZT(41), RSPHIT(41) PVARB  11
      COMMON /IDVARB/RK,ETA(41),PHIP(41),DTIL(41),DTILE(41),DETA,TP(24) IDVARB  2
      COMMON/SVARB/T,Z      , PHI   , DT    , DZ    , DPHI  , ZINT  ,  SVARB   2
     *       ZEND  , PI    , ALPHA , GAMMA , SIGMA , XMACH , TAPE1 ,  SVARB   3
     *       TAPE2 , DISK1 , ALPH  , DISK2 , SIGM  , NPRNT , DZDT  ,  SVARB   4
     *       DZDPH , ZM    , TMWO  , TMLD  , TMW   , TML   , TTMW  ,  SVARB   5
     *       TTML  , RZ    , BZ    , NIPHI , NIT   , KPHI  , NITER ,  SVARB   6
     *       NPHI  , NPHI1 , NPHI2 , NPHI3 , NPHM1 , NPHM2 , NPHM3 ,  SVARB   7
     *       NT    , NT1   , NT2   , NT3   , PHIFO , NCONE , RADI  ,  SVARB   8
     *       PHIF  , METHOD, LAG   , N8C   , PINF  , RHOIN , UINF  ,  SVARB   9
     *QINF,GASCON,NREAL,NPUNCH                                          SVARB  10
      INTEGER DISK1,DISK2,TAPE1,TAPE2                                   SVARB  11
      COMMON /L1800/NTEST(11),NCRTA,NOSXY(4),NCRTB(3),NSWCH(32),IN(32)  L1800  2
      COMMON/ENTRO/S(41),ZBS,ZFLD,ITPRTB,ITPRTF,NCASE,NTOSOS            ENTRO   2
C..FORM CONSERVATIVE VARIABLES AT ALL POINTS                            DIFFR   9
      CALL IOCON(0)                                                     DIFFR  10
      DO 1 K=3,NPHI                                                     DIFFR  11
      DO 1 J=3,NT2                                                      DIFFR  12
      DO 1 N=1,4                                                        DIFFR  13
      IF(J.EQ.NT2) GO TO 2                                              DIFFR  14
C..PREDICTOR STEP AT BODY AND IN FIELD                                  DIFFR  15
      ETEMP(N,J,K)=E0(N,J,K)-(DZDT*(F0(N,J+2,K)-F0(N,J,K))              DIFFR  16
     *                       +DZDPH*(G0(N,J,K+1)-G0(N,J,K)))+DZ*H0(N,J,K))DIFFR  17
      GO TO 1                                                           DIFFR  18
2     CONTINUE                                                          DIFFR  19
C..PREDICTOR STEP AT SHOCK                                              DIFFR  20
      ETEMP(N,J,K)=E0(N,J,K)-(DZDT*(F0(N,J,K)-F0(N,J-1,K))              DIFFR  21
     *                       +DZDPH*(G0(N,J,K+1)-G0(N,J,K)))+DZ*H0(N,J,K))DIFFR  22
      GO TO 1                                                           DIFFR  23
1     CONTINUE                                                          DIFFR  24
      Z=Z+DZ                                                            DIFFR  25
C..DECODE CONSERVATIVE VARIABLES                                        DIFFR  26
      CALL IOCON(2)                                                     DIFFR  27
C..CALCULATE PREDICTED SHOCK VALUES                                     DIFFR  28
      CALL SHOCKM(1)                                                    DIFFR  29
C..CALCULATES GEOMETRIC FACTORS BASED ON NEW BODY AND SHOCK GEOMETRY     DIFFR  30
      CALL GEOM1(1)                                                     DIFFR  31
C..APPLIES PLANE OF SYMMETRY BOUNDARY CONDITIONS                        DIFFR  32
      CALL BNDRYM(2)                                                    DIFFR  33
C..FORM INTERMEDIATE CONSERVATIVE VARIABLES AT ALL POINTS               DIFFR  34
      CALL IOCON(1)                                                     DIFFR  35
      DO 3 KPHI=3,NPHI                                                  DIFFR  36
      DO 3 J=3,NT2                                                      DIFFR  37
      DO 3 N=1,4                                                        DIFFR  38
      K=KPHI                                                            DIFFR  39
C..DISSIPATION FUNCTION                                                 DIFFR  40
      DISS=0.0                                                          DIFFR  41
      IF (CONST(4).NE.0.) .OR. CONST(5).NE.0.0) CALL DISSPM(N,J,K,DISS) DIFFR  42
      IF(J.EQ.3) GO TO 7                                                DIFFR  43
      IF(J.EQ.NT2) GO TO 5                                              DIFFR  44
C..CORRECTOR IN FIELD                                                   DIFFR  45
      ETEMP(N,J,K)=0.5*(E0(N,J,K)+ETEMP(N,J,K)-(DZDT*(F0(N,J,K)         DIFFR  46
     *-F0(N,J-1,K))+DZDPH*(G0(N,J,K)-G0(N,J,K-1)))+DZ*H0(N,J,K))+DISS)  DIFFR  47
      GO TO 3                                                           DIFFR  48
5     CONTINUE                                                          DIFFR  49
C..CORRECTOR AT SHOCK                                                   DIFFR  50
      ETEMP(N,J,K)=0.5*(ETEMP(N,J,K)+E0(N,J,K)-(DZDT*(F0(N,J,K)         DIFFR  51
     *-F0(N,J-1,K))+DZDPH*(G0(N,J,K)-G0(N,J,K-1)))+DZ*H0(N,J,K))+DISS)  DIFFR  52
      GO TO 3                                                           DIFFR  53
7     CONTINUE                                                          DIFFR  54
C..CORRECTOR AT BODY                                                    DIFFR  55
      ETEMP(N,J,K)=0.5*(ETEMP(N,J,K)+E0(N,J,K)-(DZDT*(F0(N,4,K)-F0(N,3,K DIFFR  56
     *)                                                                 DIFFR  57
     *)+DZDPH*(G0(N,3,K)-G0(N,3,K-1))+DZ*H0(N,3,K))+DISS)                DIFFR  58
3     CONTINUE                                                          DIFFR  59
C..DECODE CONSERVATIVE VARIABLES                                        DIFFR  60
      CALL IOCON(2)                                                     DIFFR  61
C..CALCULATE CORRECTED SHOCK VALUES                                     DIFFR  62
      CALL SHOCKM(2)                                                    DIFFR  63
C..CALCULATES GEOMETRIC FACTORS BASED ON OLD BODY AND NEW SHOCK GEOMETRY DIFFR  64
      CALL GEOM1(2)                                                     DIFFR  65
C..RESETS BODY VARIABLES                                                DIFFR  66
      CALL BNDRYM(1)                                                    DIFFR  67
C..APPLIES PLANE OF SYMMETRY BOUNDARY CONDITIONS                        DIFFR  68
      CALL BNDRYM(2)                                                    DIFFR  69
      RETURN                                                            DIFFR  70
      END                                                               DIFFR  71


      SUBROUTINE DISSPM(N,J,K,DISS)                                     DISSPM  2
      LEVEL 2,ETEMP,E0,F0,G0,H0                                         DISSPM  3
      COMMON/LARGE/ETEMP(4,24,41),E0(4,24,41),                          DISSPM  4
     *       F0(4,24,41)     , G0(4,24,41) , H0(4,24,41)                DISSPM  5
      LEVEL 2, RHO,P,U,V,W,RO8,RO8Z,V1NF,W1NF,RO8PH,RB,RBZ,RBPH,DTDPH,  DISSPM  6
     *  BCT,DTDZ,DTDR,ACT,ICONST,GAM,CONST,NREGON,RS,RSZ,RSPHI,RST,RSZT,  DISSPM  7
     *  RSPHIT                                                         DISSPM  8
      COMMON/PVARB/RHO(24,41),P(24,41),U(24,41),V(24,41),W(24,41),      DISSPM  9
     *       RO8(41)     , RO8Z(41) , V1NF(41) , W1NF(41) ,           -DISSPM 10
     *       RO8PH(41)   ,            RB(41)   , RBZ(41)  , RBPH(41) , -DISSPM 11
     *      DTDPH(24,41), BCT(41)  ,DTDZ(24,41),DTDR(41) , ACT(41)  , -DISSPM 12
     *       ICONST(50) , GAM(20)  , CONST(50) ,NREGON    , RS(41)   , -DISSPM 13
     *       RSZ(41)    , RSPHI(41), RST(41)   , RSZT(41), RSPHIT(41)  DISSPM 14
      COMMON/IDVARB/RK,ETA(41),PHIP(41),DTIL(41),DTILE(41),DETA,TP(24)  DISSPM 15
      COMMON/SVARB/T,Z      , PHI   , DT    , DZ    , DPHI  , ZINT  ,  DISSPM 16
     *       ZEND  , PI    , ALPHA , GAMMA , SIGMA , XMACH , TAPE1 , -DISSPM 17
     *       TAPE2 , DISK1 , ALPH  , DISK2 , SIGM  , NPRNT , DZDT  , -DISSPM 18
     *       DZDPH , ZM    , TMWO  , TMLD  , TMW   , TML   , TTMW  , -DISSPM 19
     *       TTML  , RZ    , BZ    , NIPHI , NIT   , KPHI  , NITER , -DISSPM 20
     *       NPHI  , NPHI1 , NPHI2 , NPHI3 , NPHM1 , NPHM2 , NPHM3 , -DISSPM 21
     *       NT    , NT1   , NT2   , NT3   , PHIFO , NCONE , RADI  , -DISSPM 22
     *       PHIF  , METHOD, LAG   , N8C   , PINF  , RHOIN , UINF  , -DISSPM 23
     *       QINF                                                     DISSPM 24
      INTEGER DISK1,DISK2,TAPE1,TAPE2                                   DISSPM 25
C                                                                       DISSPM 26
```

```
C
C.....  CONST(4)=S  , LAX DAMPING
C.....  CONST(4)=0  , NO DAMPING
C.....  CONST(4)=2  , 4TH ORDER DAMPING
C
C
      IF(CONST(4))21,1,20
C...DISSIPATION TERM IN THE RADIAL DIRECTION
 20   IF(J .GE. 5 .AND. J .LE. NT)GO TO 5
      IF(J.LT.5) GO TO 7
      JD=NT
      GO TO 6
    7 JD=5
      GO TO 6
    5 JD=J
    6 DISSR=-CONST(4)*0.01*(EO(N,JD+2,K)+EO(N,JD-2,K)-4.0*(EO(N,JD+1,K)    -
     * +EO(N,JD-1,K))+6.0*EO(N,JD,K))
      GO TO 2
 21   CONTINUE
      IF(J .GE. 4 .AND. J .LE. NT)GO TO 5)
      IF(J.LT. 4)GO TO 70
      JD=NT
      GO TO 60
 70   JD=4
      GO TO 60
 50   JD=J
 60   DISSR=-CONST(4)*(.125*(EO(N,JD+1,K)+EO(N,JD-1,K))-0.25*EO(N,JD,K))
      GO TO 2
    1 DISSR=0.0
C
C
C.....  CONST(5)=2  , LAX DAMPING
C.....  CONST(5)=0  , NO DAMPING
C.....  CONST(5)=2  , 4TH ORDER DAMPING
C
C
 2    IF(CONST(5))31,3,11
C...DISSIPATION TERM IN THE MERIDIANOL DIRECTION
 31   CONTINUE
      IF(K .GE. 4 .AND. K .LE. NPHM1)GO TO 80
      IF(K .LT. 4)GO TO 100
      KD=NPHM1
      GO TO 90
 100  KD=4
      GO TO 90
 80   KD=K
 90   DISSP=-CONST(5)*(.125*(EO(N,J,KD+1)+EO(N,J,KD-1))-0.25*EO(N,J,KD))
      GO TO 4
   11 CONTINUE
      IF(K.EQ.3) P(J,1)=P(J,5)
      IF(K.EC.NPH1) P(J,NPH1Z)=P(J,NPHM2)
      PF1=ABS(P(J,K+2)-2.0*P(J,K+1)+P(J,K))/(P(J,K+2)+2.0*P(J,K+1)+P(J,K
     *))
      PF2=ABS(P(J,K+1)-2.0*P(J,K)+P(J,K-1))/(P(J,K+1)+2.0*P(J,K)+P(J,K-1
     *))
      PF3=ABS(P(J,K)-2.0*P(J,K-1)+P(J,K-2))/(P(J,K)+2.0*P(J,K-1)+P(J,K-2
     *))
-     DISSP=0.5/DETA*((PF1+PF2)*(EO(N,J,K+1)-EO(N,J,K))-
     * (PF2+PF3)*(EO(N,J,K)-EO(N,J,K-1)))*CONST(5)
      GO TO 4
    3 DISSP=0.0
    4 DISS=DISSR+DISSP
      RETURN
      END


      SUBROUTINE EIGENM
      LEVEL 2,ETEMP,EO,FO,GO,HO
      COMMON/LARGE/ETEMP(4,24,41),EO(4,24,41),
     *       FO(4,24,41)    , GO(4,24,41) , HO(4,24,41)
      LEVEL 2, RHO,P,U,V,W,ROB,KOBZ,VINF,WINF,ROBPH,RB,RBZ,RBPH,DTDPH,
     *  BCT,DTDZ,DTDR,ACT,ICONST,GAM,CONST,NREGON,RS,RSZ,RSPHI,RST,RSZT,
     *  RSPHIT
      COMMON /PVARB/RHO(24,41) , P(24,41) , U(24,41) , V(24,41) , W(24,4
     *1) ,
     *       ROB(41)    , ROBZ(41) , VINF(41) , WINF(41) ,
     *       ROBPH(41)  ,          RB(41)   , RBZ(41)   , RBPH(41) ,
     *      DTDPH(24,41), BCT(41)  ,DTDZ(24,41),DTDR(41) , ACT(41)  ,
     *       ICONST(50) , GAM(20)  , CONST(50) ,NREGON   , RS(41)    ,
     *       RSZ(41)    , RSPHI(41), RST(41)   , RSZT(41), RSPHIT(41)
      COMMON /IDVARB/RK,ETA(41),PHIP(41),DTIL(41),DTILE(41),DETA,TP(24)
      COMMON/SVARB/T,Z      , PHI    , DT    , OZ    , DPHI  , Z1NT  ,
     *       ZEND   , PI     , ALPHA , GAMMA , SIGMA , XMACH , TAPE1 ,
     *       TAPE2 , DISK1 , ALPH  , DISK2 , SIGM  , NPRNT , DZDT  ,
     *       DZDPH , ZM     , TMWD  , TMLD  , TMW   , TML   , TTMW  ,
     *       TTPL   , KZ     , BZ    , NAPHI , NIT   , KPHI  , NITER ,
     *       NPHI  , NPHI1 , NPHI2 , NPHI3 , NPHM1 , NPHM2 , NPHM3 ,
     *       NT    , NTA   , NT2   , NT3   , PHIFD , NCONE , RADI  ,
     *       PHIF  , METHOD, LAG   , NDC   , PINF  , KHOLN , UINF  ,
     *CINF,CASCON,NREAL,NPUNCH
      INTEGER DISK1,DISK2,TAPE1,TAPE2
      COMMON/CLUSTR/KO,X1(24),X1(24),X1T(24)
      DIMENSION SG12(24,41),SG34(24,41)
C.........K4=1 IMPLYS EIGENVALUES AND STEPSIZE
      IPRNT=ICONST(4)
      SIG12M=0.0
      SIG34M=0.0
      DO 1 K=3,KPHI
      DO 1 J=3,NTZ
      T=XI(J)
      R=T*(ROB(K)-RB(K))+RB(K)
      C2=GAM(1)*P(J,K)/R*D(J,K)
      IF(C2) 17,17,18
 17   CONTINUE
      C2=-C2
 18   CONTINUE
      C=SQRT(C2)
      BP=DTDPH(J,K)*(ROB(K)-RB(K))/R
      G1=(V(J,K)+W(J,K)*BP)*U(J,K)
      GGD1=U(J,K)**2*(1.0+BP**2)+(V(J,K)+W(J,K)*BP)**2-C2*(1.0+BP**2)
      IF(GGD1) 19,19,20
 19   CONTINUE
      GGD1=-GGD1
      I=J-2
      IF (K .EQ. 3) WRITE(6,103) I
 20   CONTINUE
      G2=C*SQRT(GGD1)
      G3=U(J,K)**2-C2
      SIGB1=(G1+G2)/G3
      SIGB2=(G1-G2)/G3
      GGD2=U(J,K)**2+W(J,K)**2-C2
      IF(GGD2) 21,21,22
 21   CONTINUE
      GGD2=-GGD2
      I=J-2
      IF (K .EQ. 3) WRITE(6,104) I
 22   CONTINUE
      G4= U(J,K)*W(J,K)
      G5=C*SQRT(GGD2)
      SIGB3=(G4+G5)/G3/R*DTIL(K)
      SIGB4=(G4-G5)/G3/R*DTIL(K)
C.........COMPUTE LOCAL T AND PHI EIGENVALUES
      SIG1=ABS((DTDZ(J,K)+SIGB1*DTDR(K))*TAI(J))
      SIG2=ABS((DTDZ(J,K)+SIGB2*DTDR(K))*TAI(J))
      SIG12=AMAX1(SIG1,SIG2)
      SIG3=ABS(SIGB3)
      SIG4=ABS(SIGB4)
      SIG34=AMAX1(SIG3,SIG4)
      SG12(J,K)=SIG12
      SG34(J,K)=SIG34
      IF(SIG12.LE.SIG12M) GO TO 2
C.........LOCATE MAXIMUM U-V EIGENVALUE
      JMAX1=J
      KMAX1=K
      SIG12M=SIG12
      ICONST(11)=JMAX1
      ICONST(12)=KMAX1
 2    CONTINUE
      IF(SIG34.LE.SIG34M) GO TO 3
C.........LOCATE MAXIMUM U-W EIGENVALUE
      JMAX2=J
      KMAX2=K
      ICONST(13)=JMAX2
      ICONST(14)=KMAX2
      SIG34M=SIG34
 3    CONTINUE
 1    CONTINUE
C.........COMPUTE STEPSIZE BASED ON MAXIMUM EIGENVALUE
      DZ12=DT*CONST(9)/SIG12M
      DZ34=DETA*CONST(9)/SIG34M
      IF(DZ12.GT.DZ34) GO TO 4
      DZDT=CONST(9)/SIG12M
      DZ=DZDT*DT
      DZDPH=DZ/DETA
      ICONST(13)=100*ICONST(13)
      ICONST(14)=100*ICONST(14)
      GO TO 6
 4    CONTINUE
      DZDPH=CONST(9)/SIG34M
      DZ=DZDPH*DETA
```

```
      DZDT=DZ/DT                                                        EIGENM  86
      ICONST(11)=100*ICONST(11)                                         EIGENM  87
      ICONST(12)=100*ICONST(12)                                         EIGENM  88
6     CONTINUE                                                          EIGENM  89
      RETURN                                                            EIGENM  90
  103 FORMAT(1HO,41HNEGATIVE SIGMA-BAR-1 IN EIGENM INDICATES ,          EIGENM  91
     *   19HSUBSONIC FLOW AT I=,I2)                                     EIGENM  92
  104 FORMAT(1HO,41HNEGATIVE SIGMA-BAR-2 IN EIGENM INDICATES ,          EIGENM  93
     *   19HSUBSONIC FLOW AT I=,I2)                                     EIGENM  94
      END                                                               EIGENM  95


      SUBROUTINE GEOM(RNOSE,ANG,RZ,DRDZ,ZSTA,M,R90)                    GEOM     2
      DIMENSION ZSTA(200),DRDZ(200),RZ(200)                             GEOM     3
      COMMON /NOBOD/ XBOD(100),YBOD(100),NBOD                           GEOM     4
C                                                                       GEOM     5
C.....FUNCTION DEFINITIONS                                              GEOM     6
C                                                                       GEOM     7
      E(A)=EXPI(-ABS(A-RNOSE)/H)                                        GEOM     8
      F(A,B)=.5*A*((SIN(2.*B)-2.*SQRT(E(A)-E(A)**2))/(E(A)-SIN(B)**2))  GEOM     9
      G(A,B)=ABS(A*((A**5*SIN(B)*COS(B)+SQRT(A**6-1.))/(A**6*COS(B)**2- GEOM    10
     <           1.)))                                                  GEOM    11
C                                                                       GEOM    12
C.....M .NE. 0.0 ) IONOPAUSE FOR ANONMAGNETIC PLANET                    GEOM    13
C.....M .EQ. 0.0 ) EQUATORIAL PLANE FOR A MAGNETIC PLANET               GEOM    14
C                                                                       GEOM    15
      IF(M .EQ. 0.0)GO TO 10                                            GEOM    16
      IF(M.LT.0.0) GO TO 20                                             GEOM    17
      IF (M .LT. 0.01) GO TO 30                                         GEOM    18
C                                                                       GEOM    19
C.....THIS DETERMINES THE BODY SHAPE OF A NONMAGNETIC PLANET            GEOM    20
C                                                                       GEOM    21
C                                                                       GEOM    22
      IMAX=200                                                          GEOM    23
      PI=3.1415926535898                                                GEOM    24
      RADI=180./PI                                                      GEOM    25
      DELTAT=ANG/FLOAT(IMAX)                                            GEOM    26
      THETA=90.0                                                        GEOM    27
      R1=R90                                                            GEOM    28
C                                                                       GEOM    29
C.....PERFORM AN INTEGRATION FROM 0 TO 170 DEGREES                      GEOM    30
C                                                                       GEOM    31
      ZSTA(1)=1.0                                                       GEOM    32
      RZ(1)=R90                                                         GEOM    33
      XX=THETA/RADI                                                     GEOM    34
      DRDZ(1)=F(R1,XX)/R1                                               GEOM    35
      J=2                                                               GEOM    36
      IMAXM1=IMAX-1                                                     GEOM    37
      DO 5 I=1,IMAXM1                                                   GEOM    38
C.....PREDICTOR                                                         GEOM    39
      XX=THETA/RADI                                                     GEOM    40
      R=R1+DELTAT/RADI*F(R1,XX)                                         GEOM    41
      THETA=THETA+DELTAT                                                GEOM    42
      XXX=THETA/RADI                                                    GEOM    43
C.....CORRECTOR                                                         GEOM    44
      R=R1+0.5*DELTAT/RADI*(F(R1,XX)+F(R,XXX))                          GEOM    45
      DRDTH=F(R,XXX)                                                    GEOM    46
      R1=R                                                              GLOM    47
      ZSTA(J)=1.+R*COS((180.-THETA)/RADI)                               GEOM    48
      RZ(J)=R*SIN((180.-THETA)/RADI)                                    GEOM    49
      DRDZ(J)=(-DRDTH*SIN(XXX)+R*COS(XXX))/(-DRDTH*COS(XXX)+R*SIN(XXX)) GEOM    50
      J=J+1                                                             GEOM    51
5     CONTINUE                                                          GEOM    52
      RETURN                                                            GEOM    53
C                                                                       GEOM    54
C       USE CYLINDRICAL BODY FOR M/R90 .LT. 0.01                        GEOM    55
C                                                                       GEOM    56
30    ZSTA(1)=1.0                                                       GEOM    57
      ZSTA(2)=1.0                                                       GEOM    58
      RZ(1)=R90                                                         GEOM    59
      RZ(2)=R90                                                         GEOM    60
      DRDZ(1)=0.0                                                       GEOM    61
      DRDZ(2)=0.0                                                       GEOM    62
      RETURN                                                            GEOM    63
10    CONTINUE                                                          GEOM    64
C                                                                       GEOM    65
C.....THIS DETERMINES THE BODY SHAPE OF A MAGNETIC PLANET               GEOM    66
C                                                                       GEOM    67
C                                                                       GEOM    68
      J=1                                                               GEOM    69
      ANG=ANG*90.                                                       GEOM    70
      IMAX=359                                                          GEOM    71
      PI=3.1415926535898                                                GEOM    72
      RADI=180./PI                                                      GEOM    73
      DELTAT=ANG/FLOAT(IMAX)                                            GEOM    74
      THETA=90.                                                         GEOM    75
      R1=RNOSE                                                          GEOM    76
C                                                                       GEOM    77
C.....PERFORM AN INTEGRATION FROM 90 TO 260 DEGREES                     GEOM    78
C                                                                       GEOM    79
      DO 15 I=1,IMAX                                                    GEOM    80
C.....PREDICTOR                                                         GEOM    81
      XX=THETA/RADI                                                     GEOM    82
      R=R1+DELTAT/RADI*G(R1,XX)                                         GEOM    83
      THETA=THETA+DELTAT                                                GEOM    84
      XXX=THETA/RADI                                                    GEOM    85
C.....CORRECTOR                                                         GEOM    86
      R=R1+0.5*DELTAT/RADI*(G(R1,XX)+G(R,XXX))                          GEOM    87
      DRDTH=G(R,XXX)                                                    GEOM    88
      R1=R                                                              GEOM    89
      IF(THETA .LT. 180.)GO TO 15                                       GEOM    90
      ZSTA(J)=1.+R*COS((270.-THETA)/RADI)                               GEOM    91
      RZ(J)=R*SIN((270.-THETA)/RADI)                                    GEOM    92
      DRDZ(J)=(-DRDTH*COS(XXX)+R*SIN(XXX))/(-DRDTH*SIN(XXX)-R*COS(XXX)) GEOM    93
      J=J+1                                                             GEOM    94
 15   CONTINUE                                                          GEOM    95
      RETURN                                                            GEOM    96
C                                                                       GEOM    97
C.....BODY SHAPE TABLE SUPPLIED BY USER                                 GEOM    98
C                                                                       GEOM    99
   20 CONTINUE                                                          GEOM   100
      J=1                                                               GEOM   101
      DX2=0.0                                                           GEOM   102
      DR2=0.0                                                           GEOM   103
      NBODM=NBOD-1                                                      GEOM   104
      DO 25 I=1,NBODM                                                   GEOM   105
      DX1=DX2                                                           GEOM   106
      DR1=DR2                                                           GEOM   107
      DX2=XBOD(I+1)-XBOD(I)                                             GEOM   108
      DR2=YBOD(I+1)-YBOD(I)                                             GEOM   109
      IF (XBOD(I) .GT. 0.0) GO TO 25                                    GEOM   110
      ZSTA(J)=1.0-XBOD(I)                                               GEOM   111
      RZ(J)=YBOD(I)                                                     GEOM   112
      D1=SQRT(DR1*DR1+DX1*DX1)                                          GEOM   113
      D2=SQRT(DR2*DR2+DX2*DX2)                                          GEOM   114
      DRDZ(J)=-(DR1*D2/DX1+DR2*D1/DX2)/(D1+D2)                          GEOM   115
      J=J+1                                                             GEOM   116
   25 CONTINUE                                                          GEOM   117
      ZSTA(J)=1.0-XBOD(NBOD)                                            GEOM   118
      RZ(J)=YBOD(NBOD)                                                  GEOM   119
      DRDZ(J)=-(DR2*(2.0*D2+D1)/DX2-DR1*D2/DX1)/(D1+D2)                 GEOM   120
      RETURN                                                            GEOM   121
      END                                                               GEOM   122


      SUBROUTINE GLOM2(KS)                                              GEOM1    2
      LEVEL 2,TEMP,LO,FO,GO,HO                                          CVARB    2
      COMMON/LARGE/TEMP(4,24,41),E(4,24,41),                            CVARB    3
     *       FO(4,24,41)    , GO(4,24,41) , HO(4,24,41)                CVARB    4
      LEVEL 2, RHO,P,U,V,W,RDB,RDBZ,VINF,WINF,RDBPH,RB,RBZ,RBPH,DTDPH,  PVARB    2
     *  BCT,DTDZ,DTDR,ACT,ICONST,GAM,CONST,NK,GUN,RS,KSZ,RSPHI,RST,RSZT,  PVARB    3
     *  RSPHIT                                                         PVARB    4
      COMMON /PVARB/RHO(24,41) , P(24,41) , U(24,41) , V(24,41) , W(24,4 PVARB    5
     *1) ,                                                            PVARB    6
     *       RDB(41)    , RDBZ(41) , VINF(41) , WINF(41) ,            PVARB    7
     *       RDBPH(41)  ,            RB(41)   , RBZ(41)  , RBPH(41) , PVARB    8
     *     DTDPH(24,41), BCT(41)  ,DTDZ(24,41),DTDR(41) , ACT(41)  ,  PVARB    9
     *       ICONST(20) , GAM(20)  , CONST(20) ,KREGUN   , RS(41)  ,   PVARB   10
     *       RSZ(41)    , RSPHI(41), KST(41)   , RSZT(41), RSPHIT(41)  PVARB   11
      COMMON /IDVARB/RK,ETA(41),PHIP(41),DTDL(41),DTDLE(41),DETA,TP(24) IDVARB    2
      COMMON/SVARB/Y,Z     , PHI    , ET     , DP41   , ZINT  ,         SVARB    2
     *       ZEND  , PI     , ALPHA  , GAMMA  , SIGMA  , XMACH , TAPE1 ,SVARB    3
     *       TAPE2 , DISK1  , ALPH   , DISK2  , SIGM   , NPRNT , DZDI  ,SVARB    4
     *       DZDPH , ZM     , TMWO   , TMLO   , TMW    , TML   , TTMW  ,SVARB    5
     *       ITML  , RZ     , RZ     , N1PH1  , NIT    , KPHI  , NITER ,SVARB    6
     *       NPH1  , NP41   , NPH12  , NPH13  , NPHM1  , NPHM2 , NPHM3 ,SVARB    7
     *       NT    , NT1    , NT2    , NT3    , PHIFO  , NCONE , RAD1  ,SVARB    8
     *       PHIF  , MSTHOD , LAG    , NOC    , PINF   , KHDIN , UINF  ,SVARB    9
     *,DINF,GAMGUN,NPLAL,NPLNCH                                         SVARB   10
      INTEGER DISK1,DISK2,TAPE1,TAPE2                                   SVARB   11
      COMMON/CLUSTR/XJ,XI(24),FXI(24),TXIT(24)                          CLUSTR   2
      IPRNT=ICONST(4)                                                   GEOM1    8
      IF(KS.EQ.2) GO TO 12                                              GEOM1    9
      CALL CROMS(1,PHIP,NPH1,Z,RB,RBZ,RBPH,IPRNT)                       GEOM1   10
 10   CONTINUE                                                          GEOM1   11
      CALL GLOM2(KS)                                                    GEOM1   12
      DO 1 J=3,NT2                                                      GEOM1   13
      I=XI(J)                                                           GEOM1   14
      DO 2 K=2,NPHI2                                                    GEOM1   15
      PHI=PHI(K)                                                        GEOM1   16
```

```
      A=-RBZ(K)-T*(ROBZ(K)-RBZ(K))                                          GEOM1   17
      B=-RBPH(K)-T*(ROBPH(K)-RBPH(K))                                       GEOM1   18
      C=RBB(K)-RB(K)                                                        GEOM1   19
      D=-(ROBZ(K)-RBZ(K))                                                   GEOM1   20
      E=-(ROBPH(K)-RBPH(K))                                                 GEOM1   21
      DTDZ(J,K)=A/C                                                         GEOM1   22
      DTDPH(J,K)=B/C                                                        GEOM1   23
      DTDR(K)=1./C                                                          GEOM1   24
      ACT(K)=D/C                                                            GEOM1   25
      BCT(K)=E/C                                                            GEOM1   26
      R=C*T+RB(K)                                                           GEOM1   27
      X=R*SIN(PHI)                                                          GEOM1   28
      Y=-R*COS(PHI)                                                         GEOM1   29
2     CONTINUE                                                              GEOM1   30
1     CONTINUE                                                              GEOM1   31
      RETURN                                                                GEOM1   32
      END                                                                   GEOM1   33


      SUBROUTINE GEOM2(K3)                                                  GEOM2   2
      LEVEL 2,ETEMP,ED,FD,GD,HD                                             CVARB   2
      COMMON/LARGE/ETEMP(4,24,41),ED(4,24,41),
     *       FD(4,24,41)     , GD(4,24,41) , HD(4,24,41)                    CVARB   3
      LEVEL 2, RHO,P,U,V,W,ROB,ROBZ,VINF,WINF,ROBPH,Rb,RBZ,RBPH,DTDPH,      PVARB   2
     *  BCT,DTDZ,DTDR,ACT,ICONST,GAM,CONST,NREGON,RS,RSZ,RSPHI,RST,RSZT,     PVARB   3
     *  RSPHIT                                                              PVARB   4
      COMMON /PVARB/RHO(24,41) , P(24,41) , U(24,41) , V(24,41) , W(24,4    PVARB   5
     *1) ,                                                                  PVARB   6
     *       ROB(41)     , ROBZ(41) , VINF(41) , WINF(41) ,                 PVARB   7
     *       ROBPH(41)   ,            RB(41)   , RBZ(41)  , RBPH(41) ,     PVARB   8
     *      DTDPH(24,41), BCT(41)  ,DTDZ(24,41),DTDR(41) , ACT(41)  ,     PVARB   9
     *       ICONST(50) , GAM(20)  , CONST(50) ,NREGON    , RS(41)   ,     PVARB   10
     *       RSZ(41)    , RSPHI(41), RST(41)   , RSZT(41), RSPHIT(41)      PVARB   11
      COMMON /IDVARB/RK,ETA(41),PHIP(41),DTIL(41),DTILb(41),DETA,TP(24)     IDVARB  2
      COMMON/SVARB/T,Z      , PHI    , DT     , DZ     , DPHI   , ZINT   ,  SVARB   2
     *       ZEND   , PI     , ALPHA  , GAMMA  , SIGMA  , XMACH  , TAPE1  ,  SVARB   3
     *       TAPE2  , DISK1  , ALPH   , DISK2  , SIGM   , NPRNT  , DZDT   ,  SVARB   4
     *       DZDPH  , ZM     , TMWD   , TMLD   , TMW    , TML    , TTMW   ,  SVARB   5
     *       TTML   , RZ     , BZ     , NIPHI  , NIT    , KPHI   , NITER  ,  SVARB   6
     *       NPHI   , NPHI1  , NPHI2  , NPHI3  , NPHM1  , NPHM2  , NPHM3  ,  SVARB   7
     *       NT     , NT1    , NT2    , NT3    , PHIFD  , NCONE  , RADI   ,  SVARB   8
     *       PHIF   , METHOD, LAG    , NBC    , PINF   , RHOIN  , UINF   ,  SVARB   9
     *QINF,GASCON,NREAL,NPUNCH                                              SVARB   10
      INTEGER DISK1,DISK2,TAPE1,TAPE2                                       SVARB   11
      DO 1 K=1,NPHI2                                                        GEOM2   7
      PHI=PHIP(K)                                                           GEOM2   8
      GO TO (3,2),K3                                                        GEOM2   9
2     CONTINUE                                                              GEOM2   10
      ROB(K)=RS(K)                                                          GEOM2   11
      ROBZ(K)=RSZ(K)                                                        GEOM2   12
      ROBPH(K)=RSPHI(K)                                                     GEOM2   13
      GO TO 4                                                               GEOM2   14
3     CONTINUE                                                              GEOM2   15
      ROB(K)=RST(K)                                                         GEOM2   16
      ROBZ(K)=RSZT(K)                                                       GEOM2   17
      ROBPH(K)=RSPHIT(K)                                                    GEOM2   18
4     CONTINUE                                                              GEOM2   19
1     CONTINUE                                                              GEOM2   20
      RETURN                                                                GEOM2   21
      END                                                                   GEOM2   22


      SUBROUTINE GEOM3(K7,PHIP,NPHI,Z,RB,RBZ,RBPH,IPRNT)                    GEOM3   2
      COMMON/JOE/ZL1,CF1,CF2,ZLF,ZTRAN,DZTRAN                               JOE     2
      COMMON /ACCNT/NRECA(51)                                               ACCNT   2
      COMMON/NINETY/ AMACH,GAMF,KM2,P1INF,R1INF,V1INF,P90(20,3),            TRANSFZ 2
     * RHO90(20,3),U90(20,3),V90(20,3),W90(20,3),RS90(3),                   TRANSFZ 3
     * RSZ90(3),RSPH90(3),HRD,R90                                           TRANSFZ 4
      LEVEL 2, RB,RBPH,RBZ                                                  GEOM3   6
      DIMENSION RB(41),RBPH(41),RBZ(41),PHIP(41)                            GEOM3   7
      DIMENSION ZSTA(200),DRDZ(200),RZ(200)                                 GEOM3   8
C                                                                           GEOM3   9
C.....CONSTANTS                                                             GEOM3   10
C                                                                           GEOM3   11
      IF(K7.NE.0) GO TO 1                                                   GEOM3   12
      CF2=1.0                                                               GEOM3   13
      ANG=89.5                                                              GEOM3   14
      NST=1                                                                 GEOM3   15
      H=HRD                                                                 GEOM3   16
      RNOSE=1.                                                              GEOM3   17
      CALL GEOM(RNOSE,ANG,RZ,DRDZ,ZSTA,H,R90)                               GEOM3   18
      RETURN                                                                GEOM3   19
C                                                                           GEOM3   20
C.....FIND CORRECT Z INTERVAL                                               GEOM3   21
C                                                                           GEOM3   22
    1 CONTINUE                                                              GEOM3   23
      N=NST                                                                 GEOM3   24
   10 IF((Z.GE.ZSTA(N)).AND.(Z.LE.ZSTA(N+1)))GO TO 20                       GEOM3   25
      N=N+1                                                                 GEOM3   26
      GO TO 10                                                              GEOM3   27
   20 CONTINUE                                                              GEOM3   28
      NST=N                                                                 GEOM3   29
      RBODY=RZ(N)+(RZ(N+1)-RZ(N))/(ZSTA(N+1)-ZSTA(N))*(Z-ZSTA(N))           GEOM3   30
      RBODYZ=DRDZ(N)+(DRDZ(N+1)-DRDZ(N))/(ZSTA(N+1)-ZSTA(N))*(Z-ZSTA(N))    GEOM3   31
      RBODPH=0.0                                                            GEOM3   32
      DO 10 K=3,NPHI                                                        GEOM3   33
      RB(K)=RBODY                                                           GEOM3   34
      RBPH(K)=RBODPH                                                        GEOM3   35
      RBZ(K)=RBODYZ                                                         GEOM3   36
   10 CONTINUE                                                              GEOM3   37
      DO 30 K=1,2                                                           GEOM3   38
      M=6-K                                                                 GEOM3   39
      I=NPHI+K                                                              GEOM3   40
      N=NPHI-K                                                              GEOM3   41
      RB(K)=RB(M)                                                           GEOM3   42
      RB(I)=RB(N)                                                           GEOM3   43
      RBZ(K)=RBZ(M)                                                         GEOM3   44
      RBZ(I)=RBZ(N)                                                         GEOM3   45
      RBPH(K)=-RBPH(M)                                                      GEOM3   46
      RBPH(I)=-RBPH(N)                                                      GEOM3   47
   30 CONTINUE                                                              GEOM3   48
      RETURN                                                                GEOM3   49
      END                                                                   GEOM3   50


      SUBROUTINE INITA                                                      INITA   2
      LEVEL 2,ETEMP,ED,FD,GD,HD                                             CVARB   2
      COMMON/LARGE/ETEMP(4,24,41),ED(4,24,41),                              CVARB   3
     *       FD(4,24,41)     , GD(4,24,41) , HD(4,24,41)                    CVARB   4
      LEVEL 2, RHO,P,U,V,W,ROB,ROBZ,VINF,WINF,ROBPH,Rb,RBZ,RBPH,DTDPH,      PVARB   2
     *  BCT,DTDZ,DTDR,ACT,ICONST,GAM,CONST,NREGON,RS,RSZ,RSPHI,RST,RSZT,     PVARB   3
     *  RSPHIT                                                              PVARB   4
      COMMON /PVARB/RHO(24,41) , P(24,41) , U(24,41) , V(24,41) , W(24,4    PVARB   5
     *1) ,                                                                  PVARB   6
     *       ROB(41)     , ROBZ(41) , VINF(41) , WINF(41) ,                 PVARB   7
     *       ROBPH(41)   ,            RB(41)   , RBZ(41)  , RBPH(41) ,     PVARB   8
     *      DTDPH(24,41), BCT(41)  ,DTDZ(24,41),DTDR(41) , ACT(41)  ,     PVARB   9
     *       ICONST(50) , GAM(20)  , CONST(50) ,NREGON    , RS(41)   ,     PVARB   10
     *       RSZ(41)    , RSPHI(41), RST(41)   , RSZT(41), RSPHIT(41)      PVARB   11
      COMMON /IDVARB/RK,ETA(41),PHIP(41),DTIL(41),DTILb(41),DETA,TP(24)     IDVARB  2
      COMMON/SVARB/T,Z      , PHI    , DT     , DZ     , DPHI   , ZINT   ,  SVARB   2
     *       ZEND   , PI     , ALPHA  , GAMMA  , SIGMA  , XMACH  , TAPE1  ,  SVARB   3
     *       TAPE2  , DISK1  , ALPH   , DISK2  , SIGM   , NPRNT  , DZDT   ,  SVARB   4
     *       DZDPH  , ZM     , TMWD   , TMLD   , TMW    , TML    , TTMW   ,  SVARB   5
     *       TTML   , RZ     , BZ     , NIPHI  , NIT    , KPHI   , NITER  ,  SVARB   6
     *       NPHI   , NPHI1  , NPHI2  , NPHI3  , NPHM1  , NPHM2  , NPHM3  ,  SVARB   7
     *       NT     , NT1    , NT2    , NT3    , PHIFD  , NCONE  , RADI   ,  SVARB   8
     *       PHIF   , METHOD, LAG    , NBC    , PINF   , RHOIN  , UINF   ,  SVARB   9
     *QINF,GASCON,NREAL,NPUNCH                                              SVARB   10
      INTEGER DISK1,DISK2,TAPE1,TAPE2                                       SVARB   11
      COMMON/JOE/ZL1,CF1,CF2,ZLF,ZTRAN,DZTRAN                               JOE     2
      COMMON/ENTRO/S(41),ZBS,ZFLD,ITPRTB,ITPRTF,NCASE,NTOSOS                ENTRO   2
      COMMON/CLUSTR/RJ,XI(24),TXI(24),TXIT(24)                              CLUSTR  2
      COMMON/NINETY/ AMACH,GAMF,KM2,P1INF,R1INF,V1INF,P90(20,3),            TRANSFZ 2
     * RHO90(20,3),U90(20,3),V90(20,3),W90(20,3),RS90(3),                   TRANSFZ 3
     * RSZ90(3),RSPH90(3),HRD,R90                                           TRANSFZ 4
      RADI=57.29578                                                         INITA   11
      PI=3.14159265                                                         INITA   12
      ZIN1=1.0                                                              INITA   13
      Z=ZINT                                                                INITA   14
      ZEND=ZLF+1.0                                                          INITA   15
      ALPHA=0.0                                                             INITA   16
      NIPHI=2                                                               INITA   17
      NIT=KM2                                                               INITA   18
      XMACH=AMACH                                                           INITA   19
      GAMMA=GAMF                                                            INITA   20
      ALPH=ALPHA/RADI                                                       INITA   21
      SIGM=SIGMA/RADI                                                       INITA   22
      PHIF=180.0/RADI                                                       INITA   23
      NPHI=NIPHI+3                                                          INITA   24
      NPHI1=NPHI+1                                                          INITA   25
      NPHI2=NPHI+2                                                          INITA   26
      NPHM1=NPHI-1                                                          INITA   27
      NPHM2=NPHI-2                                                          INITA   28
      NT=NIT+2                                                              INITA   29
      NT1=NT+1                                                              INITA   30
      NT2=NT+2                                                              INITA   31
      ICONST(1)=1                                                           INITA   32
      ICONST(10)=1                                                          INITA   33
      IPRNT=ICONST(4)                                                       INITA   34
      IF (ICONST(48).NE.1) ICONST(5)=0                                      INITA   35
      LAG=1                                                                 INITA   36
```

```
      DETA=PHIF/FLOAT(NIPHI)                                            INITA  37
      DPHI=DETA                                                         INITA  38
      DT=1.0/FLOAT(NT-1)                                                INITA  39
      DZ=DZDT*DT                                                        INITA  40
      DZDPH=DZ/DETA                                                     INITA  41
      GAM(1)=(GAMMA-1.0)*0.5                                            INITA  42
      GAM(2)=GAM(1)/GAMMA                                               INITA  43
C..MERIDIONAL CLUSTERING                                                INITA  44
      NITER=75                                                          INITA  45
      PHIFD=0.0                                                         INITA  46
      RK=0.0                                                            INITA  47
      RJ=0.0                                                            INITA  48
      IF(RK.EQ.0.0) GO TO 15                                            INITA  49
      YC=0.5/RK*ALOG((1.0+(EXP(RK)-1.0)*PHIFD/180.0)/                   INITA  50
     * (1.0-(1.0-EXP(-RK))*PHIFD/180.0))                                INITA  51
      YO1=SINH(RK*YO)                                                   INITA  52
      YO2=YO1/(RK*PHIFD/RADI)                                           INITA  53
   15 CONTINUE                                                          INITA  54
      DO 35 II=2,NPHI1                                                  INITA  55
      ZI=II-3                                                           INITA  56
      ETA(II)=ZI*DETA                                                   INITA  57
      IF(RK.GT.0.0) GO TO 40                                            INITA  58
      PHIP(II)=ETA(II)                                                  INITA  59
      DTIL(II)=1.0                                                      INITA  60
      DTILE(II)=0.0                                                     INITA  61
      GO TO 35                                                          INITA  62
40    CONTINUE                                                          INITA  63
      YO3=RK*(ETA(II)/PI-YO)                                            INITA  64
      SHETA=SINH(YO3)                                                   INITA  65
      CHETA=COSH(YO3)                                                   INITA  66
      PHIP(II)=PHIFD/RADI*(1.0+SHETA/YO1)                               INITA  67
      DTIL(II)=YO2*PI/CHETA                                             INITA  68
      DTILE(II)=-YO2*RK*SHETA/CHETA**2                                  INITA  69
35    CONTINUE                                                          INITA  70
      PHIP(2)=-PHIP(4)                                                  INITA  71
      PHIP(NPHI1)=-PHIP(NPHM1)                                          INITA  72
      DTIL(2)=DTIL(4)                                                   INITA  73
      DTIL(NPHI1)=DTIL(NPHM1)                                           INITA  74
      DTILE(2)=-DTILE(4)                                                INITA  75
      DTILE(NPHI1)=-DTILE(NPHM1)                                        INITA  76
      DTILE(3)=0.0                                                      INITA  77
      DTILE(NPHI)=0.0                                                   INITA  78
C..RADIAL CLUSTERING                                                    INITA  79
      SINHRJ=SINH(RJ)                                                   INITA  80
      DO 36 II=2,NT2                                                    INITA  81
      ZI=II-2                                                           INITA  82
      TP(II)=ZI*DT                                                      INITA  83
      IF(RK.EQ.0.0) GO TO 41                                            INITA  84
      OT=RJ*TP(II)                                                      INITA  85
      SOT=SINH(OT)                                                      INITA  86
      COT=COSH(OT)                                                      INITA  87
      XI(II)=SOT/SINHRJ                                                 INITA  88
      TXI(II)=SINHRJ/(RJ*COT)                                           INITA  89
      TXIT(II)=-SINHRJ*SOT/COT**2                                       INITA  90
      GO TO 42                                                          INITA  91
41    XI(II)=TP(II)                                                     INITA  92
      TXI(II)=1.0                                                       INITA  93
      TXIT(II)=0.0                                                      INITA  94
42    CONTINUE                                                          INITA  95
36    CONTINUE                                                          INITA  96
      NREAL=0                                                           INITA  97
      NCONE=2                                                           INITA  98
      NTDSOS=0                                                          INITA  99
      ZBS=9999.0                                                        INITA 100
      ZFLD=9999.0                                                       INITA 101
      ITPRIB=9999                                                       INITA 102
      ITPRIF=9999                                                       INITA 103
      CONST(9)=.6                                                       INITA 104
      ICONST(49)=1                                                      INITA 105
      TAPE1=1                                                           INITA 106
      TAPE2=2                                                           INITA 107
      DISK1=1                                                           INITA 108
      DISK2=3                                                           INITA 109
C                                                                       INITA 110
C..CALCULATE FREE STREAM QUANTITIES                                     INITA 111
C                                                                       INITA 112
      GAM(3)=GAMMA/(GAMMA-1.0)                                          INITA 113
      GAM(4)=1.0/(GAMMA-1.0)                                            INITA 114
      AA=1.0+GAM(1)*XMACH**2                                            INITA 115
      BB=AA-1.0                                                         INITA 116
      PINF=1.0/AA**GAM(3)                                               INITA 117
      RHOIN=1.0/AA**GAM(4)                                              INITA 118
      UINF=SQRT(BB/AA)                                                  INITA 119
      CONST(1)=SIN(ALPH)                                                INITA 120
      CONST(2)=COS(ALPH)                                                INITA 121
      UINF=UINF*CONST(2)                                                INITA 122
      DO 3 K=2,NPHI1                                                    INITA 123
      PHI=PHIP(K)                                                       INITA 124
      VINF(K)=-UINF*CONST(1)*COS(PHI)                                   INITA 125
      WINF(K)=UINF*CONST(1)*SIN(PHI)                                    INITA 126
3     CONTINUE                                                          INITA 127
      DO 26 J=3,NT2                                                     INITA 128
      DO 26 K=3,NPHI                                                    INITA 129
      P(J,K)=P90(J-2,K-2)                                               INITA 130
      RHO(J,K)=RHO90(J-2,K-2)                                           INITA 131
      U(J,K)=U90(J-2,K-2)                                               INITA 132
      V(J,K)=V90(J-2,K-2)                                               INITA 133
      W(J,K)=W90(J-2,K-2)                                               INITA 134
26    CONTINUE                                                          INITA 135
      DO 29 K=3,NPHI                                                    INITA 136
      RS(K)=RS90(K-2)                                                   INITA 137
      RSZ(K)=RSZ90(K-2)                                                 INITA 138
      RSPHI(K)=RSPH90(K-2)                                              INITA 139
29    CONTINUE                                                          INITA 140
      ZINT=1.0                                                          INITA 141
      DO 25 K=1,2                                                       INITA 142
      M=6-K                                                             INITA 143
      I=NPHI+K                                                          INITA 144
      N=NPHI-K                                                          INITA 145
      RS(K)=RS(M)                                                       INITA 146
      RS(I)=RS(N)                                                       INITA 147
      RSZ(K)=RSZ(M)                                                     INITA 148
      RSZ(I)=RSZ(N)                                                     INITA 149
      RSPHI(K)=-RSPHI(M)                                                INITA 150
      RSPHI(I)=-RSPHI(N)                                                INITA 151
25    CONTINUE                                                          INITA 152
C.........WRITE TAPE = INITIAL DATA AND BODY SHAPE                      INITA 153
      ZM=P(3,3)/ABS(RHO(3,3))**GAMMA                                    INITA 154
      ZM=ABS(ZM)                                                        INITA 155
      DO 38 K=3,NPHI                                                    INITA 156
      S(K)=P(3,K)/ABS(RHO(3,K))**GAMMA                                  INITA 157
      S(K)=ABS(S(K))                                                    INITA 158
38    CONTINUE                                                          INITA 159
      S(2)=S(4)                                                         INITA 160
      S(NPHI1)=S(NPHM1)                                                 INITA 161
      CALL GEOM3(1,PHIP,NPH1,Z,RB,RBZ,KBPH,IPRNT)                       INITA 162
      CALL GEOM1(2)                                                     INITA 163
      RETURN                                                            INITA 164
      END                                                               INITA 165

      SUBROUTINE IOCON(K1)                                              IOCON   2
      LEVEL 2,ETEMP,E3,F3,G3,H3                                         CVARB   2
      COMMON/LARGE/ETEMP(4,24,41),E3(4,24,41),                          CVARB   3
     *       F3(4,24,41)    , G3(4,24,41) , H3(4,24,41)                CVARB   4
      LEVEL 2, RHO,P,U,V,W,ROB,ROBZ,VINF,WINF,ROBPH,RB,RBZ,KBPH,DTDPH,  PVARB   2
     *  BCT,DTDZ,DTDR,AC1,ICONST,GAM,CONST,NREGON,RS,RSZ,RSPHI,RST,RSZT,  PVARB   3
     *  RSPHIT                                                         PVARB   4
      COMMON /PVARB/RHO(24,41) , P(24,41) , U(24,41) , V(24,41) , W(24,4  PVARB   5
     *3) ,                                                              PVARB   6
     *       ROB(41)    , ROBZ(41) , VINF(41) , WINF(41) ,             PVARB   7
     *       ROBPH(41)  ,            KB(41)   , RBZ(41)  , RBPH(41) ,  PVARB   8
     *      DTDPH(24,41), BC1(41)  ,DTDZ(24,41),DTDR(41) , AC1(41)  ,  PVARB   9
     *       ICONST(50) , GAM(20)  , CONST(50) ,NREGON   , RS(41)    ,  PVARB  10
     *       RSZ(41)    , RSPHI(41), RST(41)   , RSZT(41), RSPHIT(41)  PVARB  11
      COMMON /IDVARB/RK,ETA(41),PHIP(41),DTIL(41),DTILE(41),DETA,TP(24)  IDVARB   2
      COMMON/SVARB/T,Z      , PHI    , DT     , DZ     , DPHI   , ZINT  ,  SVARB   2
     *       ZERO  , PI     , ALPHA  , GAMMA  , SIGMA  , XMACH  , TAPE1 ,  SVARB   3
     *       TAPE2 , DISK1  , ALPH   , DISK2  , SIGM   , NPRNT  , DZDT  ,  SVARB   4
     *       DZDPH , ZM     , TMAD   , TMLD   , TMW    , TML    , TTMW  ,  SVARB   5
     *       TTML  , RZ     , SZ     , NIPHI  , NIT    , KPHI   , NITER ,  SVARB   6
     *       NPHI  , NPHI1  , NPHI2  , NPHI3  , NPHM1  , NPHM2  , NPHM3 ,  SVARB   7
     *       NT    , NT1    , NT2    , NT3    , PHIFD  , NCONE  , RADI  ,  SVARB   8
     *       PHIF  , METHOD , LAG    , NBC    , PINF   , RHOIN  , UINF  ,  SVARB   9
     *QINF,GASCON,NREAL,NPUNCH                                          SVARB  10
      INTEGER DISK1,DISK2,TAPE1,TAPE2                                   SVARB  11
      COMMON/WRDKOD/AWR,BWR,CWR,DWR,UWR,VWR,WWR,PWR,RHJWR,WRC1,NEWR,NTAL  WRDKOD   2
      COMMON/WRDKO1/K,J,NPHIWR,NT2WR,WRMACH,PKKFUN,WRZ                  WRDKO1   2
      COMMON /ACCNT/NRECA(51)                                           ACCNT   2
      COMMON/CLUSTR/RJ,XI(24),TXI(24),TXIT(24)                          CLUSTR   2
      DIMENSION I(4),F(4),G(4),H(4)                                     IOCON  11
      COMMON /L1800/NTEST(11),NCRTA,NOSKY(4),NCRTB(3),NSWCH(32),IN(32)  L1800   2
      II=K1+1                                                           IOCON  13
      GO TO (5,5,2),II                                                  IOCON  14
      CONTINUE                                                          IOCON  15
      DO 3 K=2,NPHI1                                                    IOCON  16
      DO 3 J=3,NT2                                                      IOCON  17
      T=XI(J)                                                           IOCON  18
C..CODE CONSERVATIVE VARIABLES                                          IOCON  19
```

```
      R=T4(ROB(K)-RB(K))+RB(K)                                          IOCON  20
      RR=RHO(J,K)                                                       IOCON  21
      PP=P(J,K)                                                         IOCON  22
      UU=U(J,K)                                                         IOCON  23
      VV=V(J,K)                                                         IOCON  24
      WW=W(J,K)                                                         IOCON  25
      IF(NSWCH(11) .GT. 0)GO TO 10                                      IOCON  26
      A=DTDZ(J,K)                                                       IOCON  27
      B=DTDR(K)                                                         IOCON  28
      C=DTDPH(J,K)                                                      IOCON  29
      D=ROB(K)-RB(K)                                                    IOCON  30
      IF(J .EQ. 2)GO TO 4C75                                            IOCON  31
      RJM1=R(J-1)*D+RB(K)                                               IOCON  32
      VSTAR1=DTDZ(J-1,K)*U(J-1,K)+B*V(J-1,K)+DTDPH(J-1,K)*W(J-1,K)/RJM1 IOCON  33
      VSTAR1=VSTAR1*UU/U(J-1,K)                                         IOCON  34
4C75  VSTAR2=A*UU+B*VV+C*WW/R                                           IOCON  35
      IF(J .EQ. NT2)GO TO 4775                                          IOCON  36
      RJP1=R(J+1)*D+RB(K)                                               IOCON  37
      VSTAR3=DTDZ(J+1,K)*U(J+1,K)+B*V(J+1,K)+DTDPH(J+1,K)*W(J+1,K)/RJP1 IOCON  38
      VSTAR3=VSTAR3*UU/U(J+1,K)                                         IOCON  39
4775  CONTINUE                                                          IOCON  40
      IF(II-1)4,4,6                                                     IOCON  41
4     VBAR=VSTAR2                                                       IOCON  42
      IF(J .LE. 4)GO TO 14                                              IOCON  43
      IF(J .GE. (NT2-1))GO TO 14                                        IOCON  44
      IF(VSTAR2 .GT. VSTAR1)VBAR=0.5*(VSTAR2+VSTAR1)                    IOCON  45
14    WBAR=WW                                                           IOCON  46
      IF(WW .GT. W(J,K-1))WBAR=0.5*(WW+W(J,K-1))                        IOCON  47
      GO TO 10                                                          IOCON  48
6     VBAR=VSTAR2                                                       IOCON  49
      IF(J .LT. 4)GO TO 10                                              IOCON  50
      IF(J .GE. (NT2-1))GO TO 10                                        IOCON  51
      IF(VSTAR3 .GT. VSTAR2)VBAR=0.5*(VSTAR3+VSTAR2)                    IOCON  52
10    WBAR=WW                                                           IOCON  53
      IF(W(J,K+1) .GT. WW)WBAR=0.5*(W(J,K+1)+WW)                        IOCON  54
10    CONTINUE                                                          IOCON  55
      E(1)=RR*UU                                                        IOCON  56
      E(2)=GAM(2)*PP+E(1)*UU                                            IOCON  57
      E(3)=E(1)*VV                                                      IOCON  58
      E(4)=E(1)*WW                                                      IOCON  59
      IF(NSWCH(11))11,11,12                                             IOCON  60
12    CONTINUE                                                          IOCON  61
      F(1)=RR*VV                                                        IOCON  62
      F(2)=F(1)*UU                                                      IOCON  63
      F(3)=GAM(2)*PP+F(1)*VV                                            IOCON  64
      F(4)=F(1)*WW                                                      IOCON  65
      G(1)=RR*WW/R                                                      IOCON  66
      G(2)=G(1)*UU                                                      IOCON  67
      G(3)=G(1)*VV                                                      IOCON  68
      G(4)=((GAM(2)*PP+RR*WW**2)/R                                      IOCON  69
      H(1)=F(1)/R                                                       IOCON  70
      H(2)=H(1)*UU                                                      IOCON  71
      H(3)=RR*(VV**2-WW**2)/R                                           IOCON  72
      H(4)=2.0*G(3)                                                     IOCON  73
      GO TO 13                                                          IOCON  74
11    CONTINUE                                                          IOCON  75
      D=GAM(2)*PP                                                       IOCON  76
      F(1)=RR*VBAR                                                      IOCON  77
      F(2)=A*D+F(1)*UU                                                  IOCON  78
      F(3)=B*D+F(1)*VV                                                  IOCON  79
      F(4)=C*D/R+F(1)*WW                                                IOCON  80
      G(1)=RR*WBAR/R                                                    IOCON  81
      G(2)=G(1)*UU                                                      IOCON  82
      G(3)=G(1)*VV                                                      IOCON  83
      G(4)=D/R+G(1)*WW                                                  IOCON  84
      H(1)=RR*VV/R-ACT(K)*E(1)-(BCT(K)+DTILE(K))*RR*WW/R-TXIT(J)*RR*VV  IOCON  85
      HH=RR*VV*UU/R                                                     IOCON  86
      H(2)=HH      -ACT(K)*E(2)-(BCT(K)+DTILE(K))*RR*WW*UU/R-TXIT(J)*RR*V IOCON  87
     $V*UU                                                              IOCON  88
      H(3)=RR*(VV**2-WW**2)/R-ACT(K)*E(3)-(BCT(K)+DTILE(K))*RR*WW*VV/R- IOCON  89
     $TXIT(J)*(GAM(2)*PP+RR*VV**2)                                      IOCON  90
      H(4)=2.0*RR*VV*WW/R-ACT(K)*F(4)-(BCT(K)+DTILE(K))*(GAM(2)*PP+RR*WW IOCON  91
     $**2)/R-TXIT(J)*RR*VV*WW                                           IOCON  92
13    CONTINUE                                                          IOCON  93
      DO 16 N=1,4                                                       IOCON  94
      IF(NSWCH(11) .GT. 0)F(N)=DTDZ(J,K)*E(N)+DTDR(K)*F(N)+DTDPH(J,K)*  IOCON  95
     *  G(N)                                                            IOCON  96
      IF(NSWCH(11) .GT. 0)H(N)=H(N)-ACT(K)*E(N)-(BCT(K)+DTILE(K))*G(N)- IOCON  97
     $TXIT(J)*F(N)                                                      IOCON  98
      H(N)=H(N)                                                         IOCON  99
      F(N)=F(N)*TXI(J)                                                  IOCON 100
      G(N)=G(N)*DTIL(K)                                                 IOCON 101
16    CONTINUE                                                          IOCON 102
      GO TO (17,18),II                                                  IOCON 103
17    CONTINUE                                                          IOCON 104
C..SET CONSERVATIVE VARIABLES AT N                                      IOCON 105
      DO 20 N=1,4                                                       IOCON 106
      EO(N,J,K)=E(N)                                                    IOCON 107
      FO(N,J,K)=F(N)                                                    IOCON 108
      GO(N,J,K)=G(N)                                                    IOCON 109
      HO(N,J,K)=H(N)                                                    IOCON 110
20    CONTINUE                                                          IOCON 111
      GO TO 3                                                           IOCON 112
25    CONTINUE                                                          IOCON 113
C..SET CONSERVATIVE VARIABLES AT N+1 TILDE                              IOCON 114
      DO 5 N=1,4                                                        IOCON 115
      IF(J.EQ.NT2) GO TO 30                                             IOCON 116
      GO TO 35                                                          IOCON 117
C..RESET ETEMP AT BODY AND SHOCK                                        IOCON 118
30    ETEMP(N,J,K)=E(N)                                                 IOCON 119
35    CONTINUE                                                          IOCON 120
      EO(N,J,K)=E(N)                                                    IOCON 121
      GO(N,J,K)=G(N)                                                    IOCON 122
      HO(N,J,K)=H(N)                                                    IOCON 123
5     CONTINUE                                                          IOCON 124
3     CONTINUE                                                          IOCON 125
      RETURN                                                            IOCON 126
2     CONTINUE                                                          IOCON 127
C..EXCEEDS CONSERVATIVE VARIABLES--PERFECT GAS.                         IOCON 128
      AA=1.4-GAM(2)                                                     IOCON 129
      DO 1 K=3,NPHI                                                     IOCON 130
      DO 1 J=3,NT2                                                      IOCON 131
      A=ETEMP(1,J,K)                                                    IOCON 132
      B=ETEMP(2,J,K)                                                    IOCON 133
      C=ETEMP(3,J,K)                                                    IOCON 134
      D=ETEMP(4,J,K)                                                    IOCON 135
      RR=-B/A                                                           IOCON 136
      V(J,K)=C/A                                                        IOCON 137
      W(J,K)=D/A                                                        IOCON 138
      CC=GAM(2)*(1.0-V(J,K)**2-W(J,K)**2)                               IOCON 139
      DD=BB**2-4.0*AA*CC                                                IOCON 140
      IF(DD) 7,8,8                                                      IOCON 141
7     CONTINUE                                                          IOCON 142
      DD=0.0                                                            IOCON 143
8     CONTINUE                                                          IOCON 144
      U(J,K)=(-BB+SQRT(DD))/(2.0*AA)                                    IOCON 145
      RHO(J,K)=A/U(J,K)                                                 IOCON 146
      P(J,K)=RHO(J,K)*(1.0-U(J,K)**2-V(J,K)**2-W(J,K)**2)               IOCON 147
1     CONTINUE                                                          IOCON 148
      RETURN                                                            IOCON 149
      END                                                               IOCON 150

      SUBROUTINE OUTPTM                                                 OUTPTM   2
      LEVEL 2,ETEMP,EO,FO,GO,HO                                         CVARB    2
      COMMON/LARGE/ETEMP(4,24,41),EO(4,24,41),                          CVARB    3
     *       FO(4,24,41)    , GO(4,24,41) , HO(4,24,41)                CVARB    4
      LEVEL 2, RHO,P,U,V,W,ROB,ROBZ,VINF,WINF,ROBPH,RB,RBZ,RBPH,DTDPH,  PVARB    2
     *  BCT,DTDZ,DTDR,ACT,ICONST,GAM,CONST,NREGON,RS,RSZ,KSPHI,KST,RSZI,  PVARB    3
     *  KSPHIT                                                         PVARB    4
      COMMON /PVARB/RHO(24,41) , P(24,41) , U(24,41) , V(24,41) , W(24,4 PVARB    5
     *1) ,                                                              PVARB    6
     *       ROB(41)    , ROBZ(41) , VINF(41) , WINF(41) ,             PVARB    7
     *       ROBPH(41)  ,            RB(41)   , RBZ(41)   , RBPH(41) , PVARB    8
     *      DTDPH(24,41), BCT(41)  ,DTDZ(24,41),DTDR(41) , ACT(41)  , PVARB    9
     *       ICONST(50) , GAM(20)  , CONST(50) ,NREGON   , RS(41)   , PVARB   10
     *       RSZ(41)    , KSPHI(41), RST(41)   , RSZT(41), RSPHIT(41) PVARB   11
      COMMON /IDVARB/RK,ETA(41),PHIP(41),DTIL(41),DTILE(41),DETA,TP(24) IDVARB   2
      COMMON/SVARB/T,Z      , PHI    , DT    , DZ    , DPHI  , ZINT  , SVARB    2
     *       ZEND   , PI     , ALPHA , GAMMA , SIGMA , XMACH , TAPE1 , SVARB    3
     *       TAPE2 , DISK1 , ALPH  , DISK2 , SIGM  , NPRNT , DZDT  , SVARB    4
     *       DZDPH , ZM     , TMWD  , TMLD  , TMW   , TML   , TTMW  , SVARB    5
     *       TTML  , RZ     , BZ    , NIPHI , NIT   , KPHI  , NITER , SVARB    6
     *       NPHI  , NPHI1 , NPHI2 , NPHI3 , NPHM1 , NPHM2 , NPHM3 , SVARB    7
     *       NT    , NT1    , NT2   , NT3   , PHIFD , NCONE , RAD1  , SVARB    8
     *       PHIF  , METHOD, LAG   , NBC   , PINF  , RHOIN , UINF  , SVARB    9
     *QINF,GASCON,NREAL,NPUNCH                                          SVARB   10
      INTEGER DISK1,DISK2,TAPE1,TAPE2                                   SVARB   11
      COMMON/ENTRO/S(41),ZBS,ZFLD,ITPRTB,ITPRTF,NCASE,NTDSOS            ENTRO    2
      COMMON/CLUSTR/KJ,X1(24),TXI(24),TXIT(24)                          CLUSTR   2
      COMMON/DNSTRM/ IPLOT,NBLUNT,NZEND,NZADD                           DNSTRM   2
      COMMON/CONT/ KVCON,VCON(20),KRCON,RCON(20),XC(20,100),YC(20,100), TRANSF1  2
     *  VF(20,100),KADF(20,100)                                         TRANSF1  3
      COMMON /BLUNT/ THETA(25),RP(20,25)                                TRANSF1  4
      LEVEL 2, VX,VY                                                    TRANSF1  5
      COMMON/VCOMP/ VX(20,100),VY(20,100)                               TRANSF1  6
      COMMON /SHOCKS/ DRSDX(100),DST(50)                                SHOCKS   2
C                                                                       OUTPTM  12
      IF (NZEND .EQ. 0) WRITE(6,610)                                    OUTPTM  13
      NZEND=NZEND+1                                                     OUTPTM  14
```

```
      KZ=NZEND+NBLUNT                                                   OUTPTM  15
      IF(Z.LT.ZPLOT+1.0) NZADD=NZEND                                    OUTPTM  16
      ZC=1.0-Z                                                          OUTPTM  17
      WRITE(6,600) NZEND,ZC,RB(3),ROB(3)                                OUTPTM  18
      DRSDX(KZ)=ROBZ(3)                                                 OUTPTM  19
      WRITE(9) NZEND,Z,DRSDX(KZ)                                        OUTPTM  20
      DO 20 J=3,NTZ                                                     OUTPTM  21
      T=X1(J)                                                           OUTPTM  22
      Y=T*(RGB(3)-RB(3))+RB(3)                                          OUTPTM  23
      WRITE(9) J,Y,U(J,3),V(J,3),RHO(J,3)                               OUTPTM  24
      JZ=J-2                                                            OUTPTM  25
      XC(JZ,KZ)=Z-1.0                                                   OUTPTM  26
      YC(JZ,KZ)=Y                                                       OUTPTM  27
      VX(JZ,KZ)=U(J,3)                                                  OUTPTM  28
      VY(JZ,KZ)=V(J,3)                                                  OUTPTM  29
      RHOF(JZ,KZ)=RHO(J,3)                                              OUTPTM  30
   20 CONTINUE                                                          OUTPTM  31
      RETURN                                                            OUTPTM  32
C                                                                       OUTPTM  33
  600 FORMAT(5X,13,3(10X,F10.4))                                        OUTPTM  34
  610 FORMAT(1H1//54X,20HMARCHING CALCULATION//54X,20(1H*)///           OUTPTM  35
     *     2X,8HSTEP NO.,4X,19HDOWNSTREAM LOCATION,4X,13HBODY ORDINATE, OUTPTM  36
     *     6X,14HSHOCK ORDINATE//)                                      OUTPTM  37
      END                                                               OUTPTM  38


      SUBROUTINE PMTURN(M1,DNU,P2P1,NITS,GAMMA)                         PMTURN   2
      REAL M1,M2,M                                                      PMTURN   3
            XNU(M) = ATAN(SQRT(C*(M*M-1.)))/SQRTC-ATAN(SQRT(M*M-1.))    PMTURN   4
            XNUP(M) = SQRT(M*M-1.)/(1.+(GM-1.)/2.*M*M)/M                PMTURN   5
      GP=GAMMA                                                          PMTURN   6
      C = (GM-1.)/(GM+1.)                                               PMTURN   7
      SQRTC = SQRT(C)                                                   PMTURN   8
      EPS = 0.1E-6                                                      PMTURN   9
      XNU1 = XNU(M1)                                                    PMTURN  10
      XNU2 = XNU1+DNU                                                   PMTURN  11
      XM = M1                                                           PMTURN  12
      DO 10 I=1,20                                                      PMTURN  13
      NITS = I                                                          PMTURN  14
      M2 = XM -(XNU(XM)-XNU2)/XNUP(XM)                                  PMTURN  15
      IF (M2 .GT. 100.0) GO TO 20                                       PMTURN  16
      IF (ABS((M2-XM)/XM) .LT. EPS) GO TO 30                            PMTURN  17
   10 XM = M2                                                           PMTURN  18
   20 CONTINUE                                                          PMTURN  19
      M2 = 100.0                                                        PMTURN  20
      WRITE(6,1)                                                        PMTURN  21
   30 CONTINUE                                                          PMTURN  22
      P2P1 = ((1.+(GM-1.)/2.*M1*M1)/(1.+(GM-1.)/2.*M2*M2))**(GM/(GM-1.)) PMTURN  23
      RETURN                                                            PMTURN  24
    1 FORMAT(10X,47H- - - - BODY TURN STOPPED AT M2 = 100.0 - - - -)    PMTURN  25
      END                                                               PMTURN  26


      SUBROUTINE SETDAT                                                 SETDAT   2
      LEVEL 2,ET,MP,EL,FJ,GO,MO                                         CVARB    2
      COMMON/LAGS/ET(MP,24,41),EJ(4,24,41),                             CVARB    3
     *     FO(4,24,41)    , GO(4,24,41) , MO(4,24,41)                   CVARB    4
      LEVEL 2, PMO,P,U,V,W,RUM,RCBZ,VINF,WINF,RUPPM,RB,RBZ,RBPM,DTDPM,  PVARB    2
     *  BCT,DTDZ,DTDR,ACT,ICONST,GAM,CONST,NREGUN,RS,RSZ,RSPHI,RST,RSZT,  PVARB    3
     *  PSPHI)                                                         PVARB    4
      COMMON /PVARB/RHO(24,41) , P(24,41) , U(24,41) , V(24,41) , W(24,4 PVARB    5
     *1) ,                                                            PVARB    6
     *       RUP(41)    , RUBZ(41) , VINF(41) , WINF(41) ,            PVARB    7
     *       RUPPM(41)  ,          RB(41)    , RBZ(41)  , RBPM(41) ,  PVARB    8
     *      DTDPM(24,41), BCT(41)  ,DTDZ(24,41),DTDR(41) , ACT(41)  ,  PVARB    9
     *       ICONST(20) , GAM(20)  , CONST(50) ,NREGUN   , RS(41)    ,  PVARB   10
     *       RSZ(41)    , RSPHI(41), RST(41)   , RSZT(41), RSPHIT(41)  PVARB   11
      COMMON /ILVARB/EK,ETA(41),PHIP(41),DTIL(41),DTILC(41),DETA,TP(24) IUVARB   2
      COMMON/SVARB/T,Z      , PHI   , DT    , DZ    , DPHI  , ZINT  ,  SVARB    2
     *       ZERO   , PI    , ALPHA , GAMMA , SIGMA , AMACH , TAPE1 ,  SVARB    3
     *       TAPE2  , DISK1 , ALPH  , DISK2 , SIGM  , NPRNT , DZOUT ,  SVARB    4
     *       DZDPM  , ZM    , TMWO  , TMLO  , TML   , TIMW  ,          SVARB    5
     *       TIML   , RZ    , DZ    , NIPM1 , NIT   , KPM1  , NITCK ,  SVARB    6
     *       KPM1   , KP41A , NPM12 , NPM13 , KPMM1 , NPMM2 , NPMM3 ,  SVARB    7
     *       NT     , NT1   , NT2   , NT3   , PHIFO , NCONE , KAUI  ,  SVARB    8
     *       PHIF   , METHOD, LAG   , NOL   , PINF  , KNOIN , UINF  ,  SVARB    9
     *QINF,QESCON,NKUAL,NPUNCH                                          SVARB   10
      INTEGER DISK1,DISK2,TAPE1,TAPE2                                   SVARB   11
      COMMON /LIMITS/NFIST(12),NCRTA,NCRTB(3),NSWCH(32),IN(32)          LIMITS   2
      COMMON/XYZ/NXYXY(3),X1(160),X2(160),X3(160),Y1(160),Y2(160),Y3(160 XYZ      2
     *),Z1(160),Z2(160),Z3(160)                                         XYZ      3
      COMMON/JOE/ZERO,CF1,CF2,ZLF,ZTRAN,DZTRAN                          JOE      2
      COMMON/EPSLC/EPSL1,EPSL2                                          EPSL     2
      COMMON/CNTRL/ISTART,ZFS,ZFLO,ITPNTS,ITPNTF,NCASE,NTSSJS           CNTRL    2
      COMMON/CFS/NCF                                                    CFS      2

      COMMON/CLUSTR/RJ,XI(24),TXI(24),TXIT(24)                          CLUSTR   2
      COMMON /ACCNT/NRECA(51)                                           ACCNT    2
      COMMON/WRDKOI/K,J,NPHIWR,NT2WR,WRMACH,PKKFUN,WRI                  WRDKOI   2
C                                                                       SETDAT  16
C     INITIALISE ARRAYS                                                 SETDAT  17
C                                                                       SETDAT  18
      ICONST(1)=0                                                       SETDAT  19
      ICONST(2)=2                                                       SETDAT  20
      ICONST(3)=0                                                       SETDAT  21
      ICONST(4)=1                                                       SETDAT  22
      DO 1 I=5,50                                                       SETDAT  23
      ICONST(I)=0                                                       SETDAT  24
    1 CONTINUE                                                          SETDAT  25
      NBC=1                                                             SETDAT  26
      METHOD=2                                                          SETDAT  27
      NPRNT=5                                                           SETDAT  28
      NWRPRT=0                                                          SETDAT  29
      NWPOUT=5                                                          SETDAT  30
      DZDT=0.0                                                          SETDAT  31
      WRMACH=0.0                                                        SETDAT  32
      DO 2 I=1,32                                                       SETDAT  33
      IN(I)=0                                                           SETDAT  34
      NSWCH(I)=0                                                        SETDAT  35
    2 CONTINUE                                                          SETDAT  36
      DO 3 I=1,160                                                      SETDAT  37
      X1(I)=0.0                                                         SETDAT  38
      X2(I)=0.0                                                         SETDAT  39
      X3(I)=0.0                                                         SETDAT  40
      Y1(I)=0.0                                                         SETDAT  41
      Y2(I)=0.0                                                         SETDAT  42
      Y3(I)=0.0                                                         SETDAT  43
      Z1(I)=0.0                                                         SETDAT  44
      Z2(I)=0.0                                                         SETDAT  45
      Z3(I)=0.0                                                         SETDAT  46
    3 CONTINUE                                                          SETDAT  47
      DO 4 I=1,11                                                       SETDAT  48
      NTEST(I)=0                                                        SETDAT  49
    4 CONTINUE                                                          SETDAT  50
      NCRTA=0                                                           SETDAT  51
      DO 5 I=1,4                                                        SETDAT  52
      NCSXY(I)=0                                                        SETDAT  53
    5 CONTINUE                                                          SETDAT  54
      DO 6 I=1,3                                                        SETDAT  55
      NCRTB(I)=0                                                        SETDAT  56
      NXYXY(I)=0                                                        SETDAT  57
    6 CONTINUE                                                          SETDAT  58
      DO 7 I=1,51                                                       SETDAT  59
      NRECA(I)=0                                                        SETDAT  60
    7 CONTINUE                                                          SETDAT  61
      DO 8 I=1,20                                                       SETDAT  62
      GAM(I)=0.0                                                        SETDAT  63
    8 CONTINUE                                                          SETDAT  64
      DO 9 I=1,10                                                       SETDAT  65
      CONST(I)=0.0                                                      SETDAT  66
    9 CONTINUE                                                          SETDAT  67
      DO 10 I=1,24                                                      SETDAT  68
      TP(I)=0.0                                                         SETDAT  69
   10 CONTINUE                                                          SETDAT  70
      DO 11 I=1,41                                                      SETDAT  71
      PGB(I)=0.0                                                        SETDAT  72
      RGBZ(I)=0.0                                                       SETDAT  73
      VINF(I)=0.0                                                       SETDAT  74
      WINF(I)=0.0                                                       SETDAT  75
      RGBPM(I)=0.0                                                      SETDAT  76
      PHI(I)=0.0                                                        SETDAT  77
      RBZ(I)=0.0                                                        SETDAT  78
      RBPM(I)=0.0                                                       SETDAT  79
      BCT(I)=0.0                                                        SETDAT  80
      DTDR(I)=0.0                                                       SETDAT  81
      ACT(I)=0.0                                                        SETDAT  82
      RS(I)=0.0                                                         SETDAT  83
      RSZ(I)=0.0                                                        SETDAT  84
      RSPHI(I)=0.0                                                      SETDAT  85
      RST(I)=0.0                                                        SETDAT  86
      RSZT(I)=0.0                                                       SETDAT  87
      RSPHIT(I)=0.0                                                     SETDAT  88
      ETA(I)=0.0                                                        SETDAT  89
      PHIP(I)=0.0                                                       SETDAT  90
      DTIL(I)=0.0                                                       SETDAT  91
      DTILC(I)=0.0                                                      SETDAT  92
      DO 12 J=1,24                                                      SETDAT  93
      RHO(J,I)=0.0                                                      SETDAT  94
      P(J,I)=0.0                                                        SETDAT  95
      U(J,I)=0.0                                                        SETDAT  96
      V(J,I)=0.0                                                        SETDAT  97
```

```
      W(J,I)=0.0
      DTDPH(J,I)=0.0
      DTDZ(J,I)=0.0
   11 CONTINUE
      RETURN
      END


      SUBROUTINE SHOCKM(K4)
      LEVEL 2,(TEMP,EU,FU,GU,HU)
      COMMON/LARGE/ETEMP(4,24,41),EU(4,24,41),
     *      FU(4,24,41)     , GU(4,24,41) , HU(4,24,41)
      LEVEL 2, RHO,P,U,V,W,RO9,ROBZ,VINF,WINF,ROBPH,RB,RBZ,RBPH,DTDPH,
     *  BCT,DTDZ,DTDR,ACT,ICONST,GA4,CONST,NREGON,RS,RSZ,RSPHI,KST,RSZT,
     *  RSPHIT
      COMMON /PVARB/RHO(24,41) , P(24,41) , U(24,41) , V(24,41) , W(24,4
     *1) ,
     *       RO9(41)     , ROBZ(41) , VINF(41) , WINF(41) ,
     *       ROBPH(41)   ,            RB(41)   , RBZ(41)  , RBPH(41) ,
     *      DTDPH(24,41), BCT(41)  ,DTDZ(24,41),DTDR(41) , ACT(41)  ,
     *       ICONST(50) , GAM(20)  , CONST(50) ,NREGON   , RS(41)   ,
     *       RSZ(41)    , RSPHI(41), KST(41)   , RSZT(41), RSPHIT(41)
      COMMON /IDVARB/RK,ETA(41),PHIP(41),DTIL(41),DTILE(41),DETA,IP(24)
      COMMON/SVARB/T,Z      , PHI    , DT     , DZ     , DPHI   , ZINT   ,
     *       ZEND   , PI     , ALPHA  , GAMMA  , SIGMA  , XMACH  , TAPE1  ,
     *       TAPE2  , DISK1  , ALPH   , DISKZ  , SIGM   , NPRNT  , DZDT   ,
     *       DZDPH  , ZM     , TMWD   , TMLD   , TMW    , TML    , TTMW   ,
     *       TTML   , RZ     , BZ     , NIPHI  , NIT    , KPHI   , NITER  ,
     *       NPHI   , NPHI1  , NPHI2  , NPHI3  , NPHM1  , NPHM2  , NPHM3  ,
     *       NT     , NT1    , NT2    , NT3    , PHIFD  , NCONE  , RADI   ,
     *       PHIF   , METHOD, LAG    , NBC    , PINF   , RHOIN  , UINF   ,
     *QINF,GAS CON,NREAL,NPUNCH
      INTEGER DISK1,DISK2,TAPE1,TAPE2
C..RANKINE-HUGONIOT FUNCTIONS
      U1TILD(A)=SQRT(C1+C2*(A-1.))
      DRSDZ(A,B,C,D)=(UINF*C+A*SQRT(D*(1.+B*B)+C*C))/D
      RHOS(A)=(A+GRATIO)/(1.+GRATIO*A)
      ABAR(A,B,D,E,F)=(F-1.)*ABS(-UINF*D+A-B*E)/((D*D+1.+E*E)*F)
      US(A,B,C)=A-C*B
      VS(A,B)=A+B
      WS(A,B,C)=A-B*C
      GAMP1=GAMMA+1.
      GAMM1=GAMMA-1.
      GRATIO=GAMP1/GAMP1
      GAMP1G=GAMP1/GAMMA
      C1=0.5*GAMM1*PINF/RHOIN
      C2=0.5*GAMP1G*C1
      GO TO (1,2),K4
    1 CONTINUE
C..SHOCK PREDICTOR
      DO 3 K=3,NPHI
    3 RST(K)=RS(K)+DZ*RSZ(K)
      DO 4 K=1,2
      M=6-K
      I=NPHI+K
      N=NPHI-K
      RST(K)=RST(M)
      RST(I)=RST(N)
    4 CONTINUE
      DO 5 K=3,NPHI
      RSPHIT(K)=(RST(K+1)-RST(K-1))/(2.0*DETA)*DTIL(K)
      PS=P(NT2,K)
      PSRAT=PS/PINF
      U1T=U1TILD(PSRAT)
      RHRAT=RHOS(PSRAT)
      RS1=RST(K)
      RSPH=RSPHIT(K)
      RSPHR=RSPH/RS1
      FACT1=VINF(K)-WINF(K)*RSPHR
      FACT2=UINF*UINF-U1T*U1T
      IF(FACT2 .LT. 0.)U1T=-U1T
      RSZT1=DRSDZ(U1T,RSPHR,FACT1,FACT2)
      RSZT(K)=RSZT1
      ABART=ABAR(VINF(K),WINF(K),RSZT1,RSPHR,RHRAT)
      UST=US(UINF,ABART,RSZT1)
      VST=VS(VINF(K),ABART)
      WST=WS(WINF(K),ABART,RSPHR)
      IF (NREAL.EQ.0) ROST=RHRAT*RHOIN
      RHO(NT2,K)=ROST
      U(NT2,K)=UST
      V(NT2,K)=VST
      W(NT2,K)=WST
    5 CONTINUE
      DO 6 K=1,2
      M=6-K
      I=NPHI+K
      N=NPHI-K
      RSZT(K)=RSZT(M)
      RSZT(I)=RSZT(N)
      PSPHIT(K)=-RSPHIT(M)
      RSPHIT(I)=-RSPHIT(N)
    6 CONTINUE
      RETURN
    2 CONTINUE
C..SHOCK CORRECTOR
      DO 8 K=3,NPHI
    8 RS(K)=RS(K)+0.5*(RSZ(K)+RSZT(K))*DZ
      DO 9 K=1,2
      M=6-K
      I=NPHI+K
      N=NPHI-K
      RS(K)=RS(M)
      RS(I)=RS(N)
    9 CONTINUE
      DO 10 K=3,NPHI
      RSPHI(K)=(RS(K+1)-RS(K-1))/(2.0*DETA)*DTIL(K)
      PS=P(NT2,K)
      PSRAT=PS/PINF
      U1T=U1TILD(PSRAT)
      RHRAT=RHOS(PSRAT)
      PS1=PS(K)
      RSPH=RSPHI(K)
      RSPHR=RSPH/RS1
      FACT1=VINF(K)-WINF(K)*RSPHR
      FACT2=UINF*UINF-U1T*U1T
      IF(FACT2 .LT. 0.)U1T=-U1T
      RSZ1 =DRSDZ(U1T,RSPHR,FACT1,FACT2)
      RSZ(K)=RSZ1
      ABART=ABAR(VINF(K),WINF(K),RSZ1 ,RSPHR,RHRAT)
      USF=US(UINF,ABART,RSZ1)
      VSF=VS(VINF(K),ABART)
      WSF=WS(WINF(K),ABART,RSPHR)
      IF (NREAL.EQ.0) ROSF=RHRAT*RHOIN
      RHO(NT2,K)=ROSF
      U(NT2,K)=USF
      V(NT2,K)=VSF
      W(NT2,K)=WSF
   10 CONTINUE
      DO 11 K=1,2
      M=6-K
      I=NPHI+K
      N=NPHI-K
      RSZ(K)=RSZ(M)
      RSZ(I)=RSZ(N)
      RSPHI(K)=-RSPHI(M)
      RSPHI(I)=-RSPHI(N)
   11 CONTINUE
      RETURN
      END
```

REFERENCES

1. Spreiter, J. R. and Jones, W. P.: On the Effect of a Weak Interplanetary Magnetic Field on the Interaction Between the Solar Wind and the Geomagnetic Field. Jour. Geophy. Res., vol. 68, 1963, pp. 3555-3564.

2. Spreiter, J. R., Alksne, A. Y., and Summers, A. L.: Hydromagnetic Flow Around the Magnetopause. Plan. & Space Sci., vol. 14, 1966, pp. 223-253.

3. Spreiter, J. R., Alksne, A. Y., and Summers, A. L.: External Aerodynamics of the Magnetosphere. Physics of the Magnetosphere (Ed. R. L. Carovillano, J. F. McClay, and H. R. Radoski), D. Reidel Pub. Co., 1968, pp. 304-378 (also NASA TN 4482, 1968).

4. Spreiter, J. R. and Alksne, A. Y.: Plasma Flow Around the Magnetosphere. Rev. Geophys., vol. 7, 1969, pp. 11-50.

5. Spreiter, J. R., Summers, A. L., and Rizzi, A. W.: Solar Wind Flow Past Nonmagnetic Planets — Venus and Mars. Plan. & Space Sci., vol. 18, 1970, pp. 1281-1289.

6. Spreiter, J. R. and Alksne, A. Y.: Solar Wind Flow Past Objects in the Solar System. Ann. Rev. Fluid Mech., vol. 2, pp. 313-354 (Ed. W. R. Sears, M. D. Van Dyke, and W. G. Vincenti), Annual Review, Inc., Palo Alto, CA, 1970.

7. Spreiter, J. R., March, C. M., Summers, A.: Hydromagnetic Aspects of Solar Wind Flow Past the Moon. Cosmic Electrodynamics, vol. 1, no. 1, 1970, pp. 5-50.

8. Spreiter, J. R. and Rizzi, A. W.: Aligned Magnetohydrodynamics Solution for Solar Wind Flow Past the Earth's Magnetosphere. Acta Astronautica, vol. 1, 1974, pp. 15-35.

9. Briggs, B. R. and Spreiter, J. R.: Theoretical Determination of the Boundary and Distortion of the Geomagnetic Field in a Steady Solar Wind. NASA TR R-178, 1963.

10. Chapman, S. and Ferraro, V. C. A.: An Outline of a Theory of Magnetic Storms. Terrest. Magnetism Atmospheric Elec., vol. 36, 1931, pp. 77-97, 171-186.

11. Midgley, J. E. and Davis, Jr., L.: Calculation by a Moment Technique of the Perturbation of the Geomagnetic Field by the Solar Wind. Jour. Geophy. Res., vol. 68, 1963, pp. 5111-5123.

12. Mead, G. D. and Beard, D. B.: Shape of the Geomagnetic Field Solar Wind Boundary. Jour. Geophy. Res., vol. 69, 1964, pp. 1169-1179.

13. Olsen, W. P.: The Shape of the Titled Magnetopause. Jour. Geophy. Res., Vol. 74, 1969, pp. 5642-5651.

14. Beard, D. B.: The Interaction of the Terrestrial Magnetic Field with the Solar Corpuscular Radiation. Jour. Geophy. Res., vol. 65, 1960, pp. 3559-3568.

15. Spreiter, J. R. and Briggs, B. R.: Theoretical Determination of the Form of the Hollow Produced in the Solar Corpuscular Stream by Interaction with the Magnetic Dipole Field of the Earth. NASA TR R-120, 1961.

16. Spreiter, J. R. and Briggs, B. R.: Theoretical Determination of the Form of the Boundary of the Solar Corpuscular Stream Produced by Interaction with the Magnetic Dipole Field of the Earth. Jour. Geophys. Res., vol. 67, no. 1, Jan. 1962, pp. 37-51.

17. Spreiter, J. R. and Briggs, B. R.: On the Choice of Condition to Apply at the Boundary of the Geomagnetic Field in the Steady-State Chapman-Ferraro Problem. Jour. Geophys. Res., vol. 67, no. 7, Jul. 1962, pp. 2983-2985.

18. Davis, Jr., L. and Beard, D. B.: A Correction to the Approximate Condition for Locating the Boundary Between a Magnetic Field and a Plasma. Jour. Geophys. Res., vol. 67, 1962, pp. 4505-4507.

19. Spreiter, J. R. and Summers, A. L.: On Conditions Near the Neutral Points on the Magnetosphere Boundary. Plan. & Space Sci., vol. 15, 1967, pp. 787-798.

20. Kuhn, G. D., Goodwin, F. K., and Perkins, Jr., S. C.: User's Manual for Space-Shuttle Computer Programs. NEAR TR 110, Apr. 1976.

21. Chaussee, D. S., Stahara, S. S., and Spreiter, J. R.: Application of an Axisymmetric Implicit Blunt Body Procedure: Computation of Solar Wind Flows Past Terrestrial Planets. AIAA Paper No. 77-700, Jun. 1977.

22. Beam, R. M. and Warming, R. F.: An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form. J. Comp. Phys., vol. 22, no. 1, Sep. 1976.

23. Kutler, P., Reinhardt, W. A., and Warming, R. F.: Multi-Shocked, Three-Dimensional Supersonic Flow Fields with Real Gas Effects. AIAA Journal, vol. 11, May 1973, pp. 657-664.

24. Kutler, P., Reinhardt, W. A., and Warming, R. F.: Numerical Computations of Multi-Shocked Three-Dimensional Supersonic Flow Fields with Real Gas Effects. AIAA Paper No. 72-702, Jun. 1972.

25. Chaussee, D. S., Holtz, T., and Kutler, P.: Inviscid Supersonic/ Hypersonic Body Flow Fields and Aerodynamics from Shock-Capturing Technique Calculations. AIAA Paper No. 75-837, Jun. 1975.

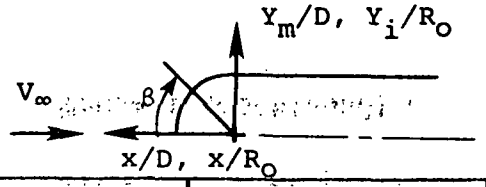
26. MacCormack, R. W.: The Effect of Viscosity in Hypervelocity Impact Cratering. AIAA Paper No. 69-354, 1969.

27. Alksne, A. Y. and Webster, D. L.: Magnetic and Electric Fields in the Magnetosheath. Plan. & Space Sci., vol. 18, 1970, pp. 1203-1212.

28. Grad, H.: Reducible Problems in Magneto-Fluid Dynamics Steady Flows. Rev. Mod. Phys., vol. 32, 1960, pp. 830-847.

29. Imai, I.: On Flows of Conducting Fluids Past Bodies. Rev. Mod. Phys., vol. 32, 1960, pp. 992-999.

30. Inouye, M. and Lomax, H.: Comparison of Experimental and Numerical Results for the Flow of a Perfect Gas About Blunt-Nosed Bodies. NASA TD-1426, Sep. 1962.



	MAGNETOPAUSE		IONOPAUSE		IONOPAUSE		IONOPAUSE		IONOPAUSE	
	Equatorial Trace		H/R_o = 0.01		H/R_o = 0.1		H/R_o = 0.2		H/R_o = 0.5	
β	x/D	Y_m/D	x/R_o	Y_i/R_o	x/R_o	Y_i/R_o	x/R_o	Y_i/R_o	x/R_o	Y_i/R_o
0°	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
2°	0.9995	0.0349	0.9994	0.0348	0.9995	0.0348	0.9995	0.0349	0.9996	0.0349
6°	0.9957	0.1046	0.9946	0.1045	0.9954	0.1046	0.9958	0.1047	0.9966	0.1047
10°	0.9879	0.1742	0.9851	0.1737	0.9870	0.1740	0.9883	0.1743	0.9906	0.1747
14°	0.9764	0.2434	0.9709	0.2421	0.9745	0.2430	0.9771	0.2436	0.9814	0.2447
18°	0.9610	0.3122	0.9520	0.3093	0.9580	0.3113	0.9621	0.3126	0.9692	0.3149
22°	0.9417	0.3805	0.9285	0.3751	0.9373	0.3787	0.9434	0.3812	0.9537	0.3853
26°	0.9185	0.4480	0.9006	0.4393	0.9125	0.4451	0.9209	0.4491	0.9349	0.4560
30°	0.8915	0.5147	0.8684	0.5014	0.8838	0.5103	0.8945	0.5164	0.9127	0.5269
34°	0.8607	0.5805	0.8320	0.5612	0.8510	0.5740	0.8643	0.5830	0.8869	0.5982
38°	0.8260	0.6453	0.7916	0.6185	0.8143	0.6362	0.8302	0.6486	0.8573	0.6698
42°	0.7874	0.7089	0.7473	0.6729	0.7738	0.6967	0.7921	0.7132	0.8238	0.7418
46°	0.7448	0.7713	0.6995	0.7243	0.7293	0.7553	0.7501	0.7767	0.7862	0.8141
50°	0.6984	0.8323	0.6481	0.7724	0.6811	0.8118	0.7040	0.8390	0.7441	0.8868
54°	0.6480	0.8919	0.5936	0.8171	0.6292	0.8660	0.6538	0.8999	0.6974	0.9598
58°	0.5935	0.9499	0.5362	0.8581	0.5736	0.9180	0.5995	0.9593	0.6456	1.0332
62°	0.5350	1.0062	0.4760	0.8953	0.5144	0.9675	0.5408	1.0172	0.5885	1.1068
66°	0.4723	1.0609	0.4134	0.9285	0.4516	1.0143	0.4778	1.0733	0.5256	1.1806
70°	0.4054	1.1137	0.3486	0.9578	0.3852	1.0585	0.4104	1.1275	0.4566	1.2546
74°	0.3340	1.1648	0.2819	0.9829	0.3154	1.0998	0.3383	1.1797	0.3809	1.3285

Table 1.- Ordinates of Various Magneto/Ionopause Shapes

β	MAGNETOPAUSE Equatorial Trace		IONOPAUSE $H/R_o = 0.01$		IONOPAUSE $H/R_o = 0.1$		IONOPAUSE $H/R_o = 0.2$		IONOPAUSE $H/R_o = 0.5$	
	x/D	Y_m/D	x/R_o	Y_i/R_o	x/R_o	Y_i/R_o	x/R_o	Y_i/R_o	x/R_o	Y_i/R_o
78°	0.2580	1.2138	0.2134	1.0041	0.2419	1.1382	0.2614	1.2298	0.2981	1.4024
82°	0.1772	1.2610	0.1435	1.0212	0.1650	1.1737	0.1796	1.2777	0.2074	1.4759
86°	0.0913	1.3060	0.0723	1.0345	0.0843	1.2062	0.0925	1.3232	0.1083	1.5491
90°	0.0	1.3491	0.0	1.0442	0.0	1.2356	0.0	1.3661	0.0	1.6216
94°	-0.0972	1.3900	-0.0735	1.0507	-0.0882	1.2620	-0.0983	1.4065	-0.1184	1.6932
98°	-0.2008	1.4289	-0.1482	1.0547	-0.1806	1.2854	-0.2029	1.4441	-0.2479	1.7636
102°	-0.3115	1.4656	-0.2246	1.0568	-0.2776	1.3058	-0.3143	1.4788	-0.3895	1.8325
106°	-0.4302	1.5002	-0.3033	1.0577	-0.3795	1.3233	-0.4332	1.5106	-0.5447	1.8997
110°	-0.5578	1.5327	-0.3851	1.0580	-0.4870	1.3380	-0.5603	1.5394	-0.7150	1.9646
114°	-0.6959	1.5630	-0.4711	1.0581	-0.6011	1.3501	-0.6968	1.5651	-0.9024	2.0269
118°	-0.8461	1.5913	-0.5626	1.0581	-0.7230	1.3598	-0.8442	1.5877	-1.1092	2.0861
122°	-0.0107	1.6175	-0.6612	1.0581	-0.8543	1.3672	-1.0043	1.6072	-1.3384	2.1419
126°	-1.1927	1.6416	-0.7688	1.0581	-0.9973	1.3727	-1.1797	1.6237	-1.5938	2.1936
130°	-1.3961	1.6638	-0.8879	1.0581	-1.1550	1.3765	-1.3737	1.6371	-1.8803	2.2409
134°	-1.6262	1.6840	-1.0218	1.0581	-1.3317	1.3790	-1.5911	1.6476	-2.2048	2.2832
138°	-1.8905	1.7022	-1.1752	1.0581	-1.5332	1.3805	-1.8387	1.6555	-2.5767	2.3200
142°	-2.1997	1.7186	-1.3544	1.0581	-1.7679	1.3812	-2.1261	1.6611	-3.0093	2.3511
146°	-2.5696	1.7332	-1.5688	1.0581	-2.0483	1.3816	-2.4679	1.6646	-3.5226	2.3761
150°	-3.0242	1.7460	-1.8328	1.0581	-2.3932	1.3817	-2.8866	1.6666	-4.1480	2.3948
154°	-3.6026	1.7571	-2.1695	1.0581	-2.8330	1.3817	-3.4189	1.6675	-4.9365	2.4077

Table 1 (Continued)

β	MAGNETOPAUSE Equatorial Trace		IONOPAUSE $H/R_o = 0.01$		IONOPAUSE $H/R_o = 0.1$		IONOPAUSE $H/R_o = 0.2$		IONOPAUSE $H/R_o = 0.5$	
	x/D	Y_m/D	x/R_o	Y_i/R_o	x/R_o	Y_i/R_o	x/R_o	Y_i/R_o	x/R_o	Y_i/R_o
158°	-4.3723	1.7665	-2.6190	1.0581	-3.4199	1.3817	-4.1281	1.6678	-5.9779	2.4152
162°	-5.4608	1.7743	-3.2567	1.0582	-4.2526	1.3817	-5.1333	1.6679	-7.4439	2.4187
166°	-7.1412	1.7805	-4.2440	1.0582	-5.5419	1.3817	-6.6897	1.6679	-9.7048	2.4197
170°	-10.1239	1.7851	-6.0011	1.0582	-7.8363	1.3818	-9.4594	1.6679	-13.7235	2.4198
174°	-17.0135	1.7882	-10.0678	1.0582	-13.1466	1.3818	-15.8695	1.6680	-23.0232	2.4198

Table 1 (Concluded)

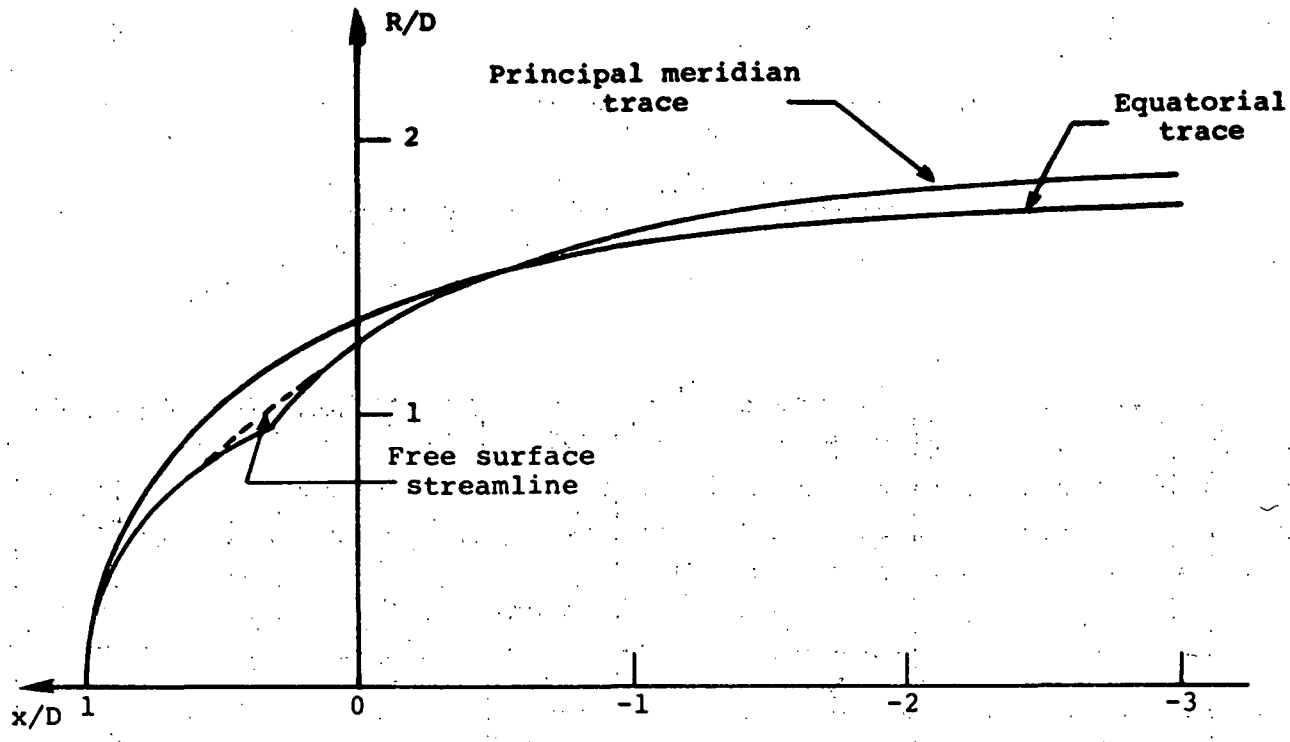


Figure 1. - Comparison of the equatorial and principal meridian traces of the magnetosphere boundary as provided by the simplified theory of equation 16.

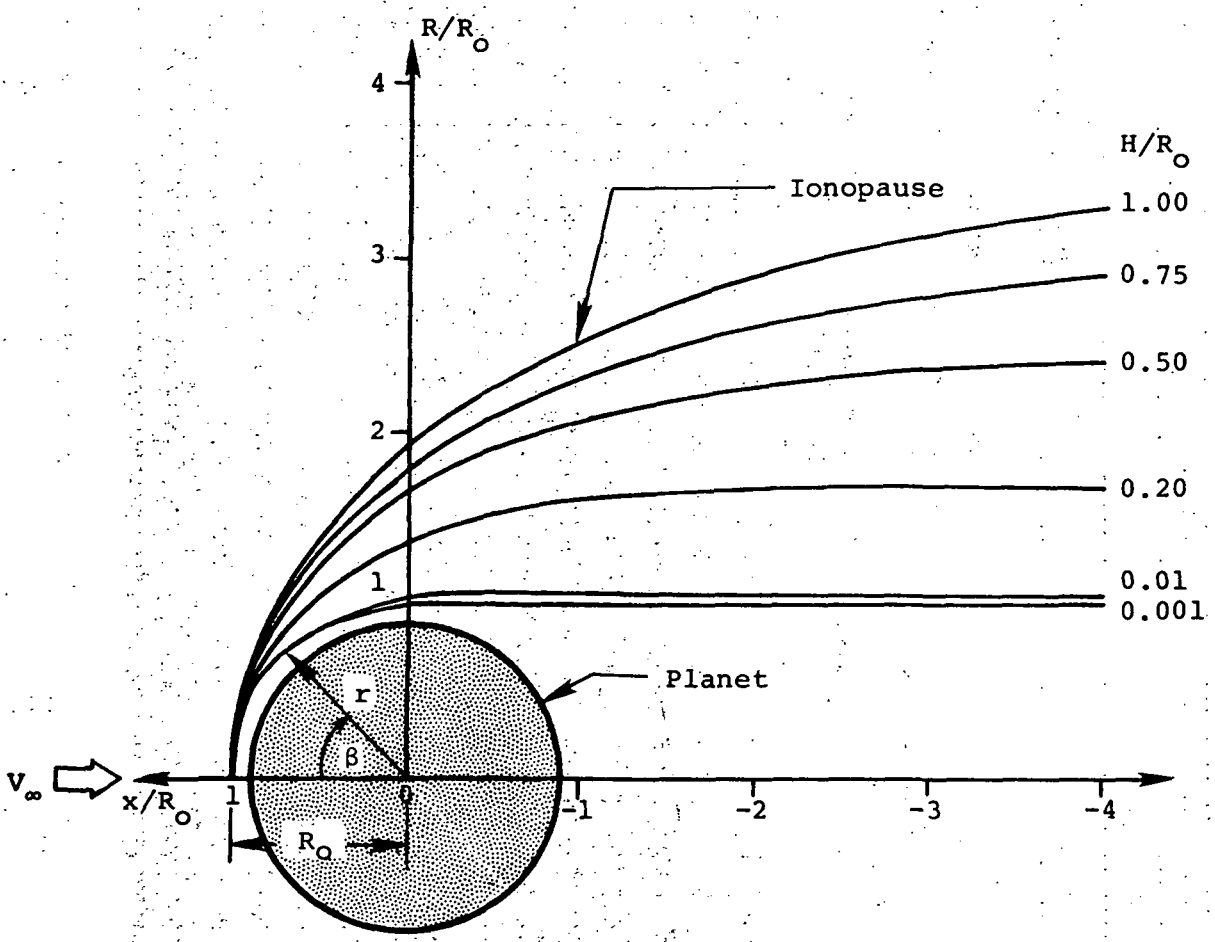


Figure 2. - Illustration of ionopause shapes for various values of the ionosphere scale height to shock standoff distance ratio H/R_o.

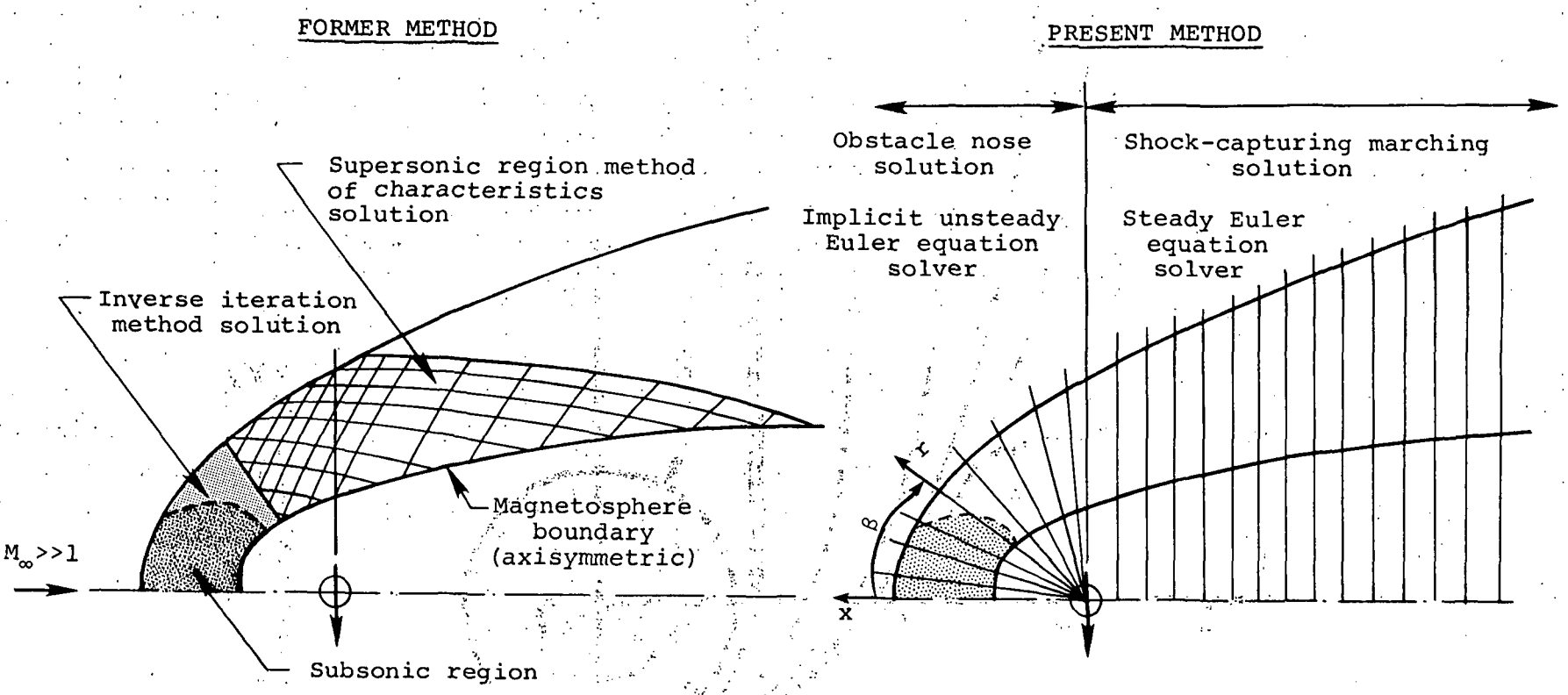


Figure 3. - Comparison of former and present computational procedures for determining the gasdynamic flow properties of solar wind-magneto/ionopause interactions.

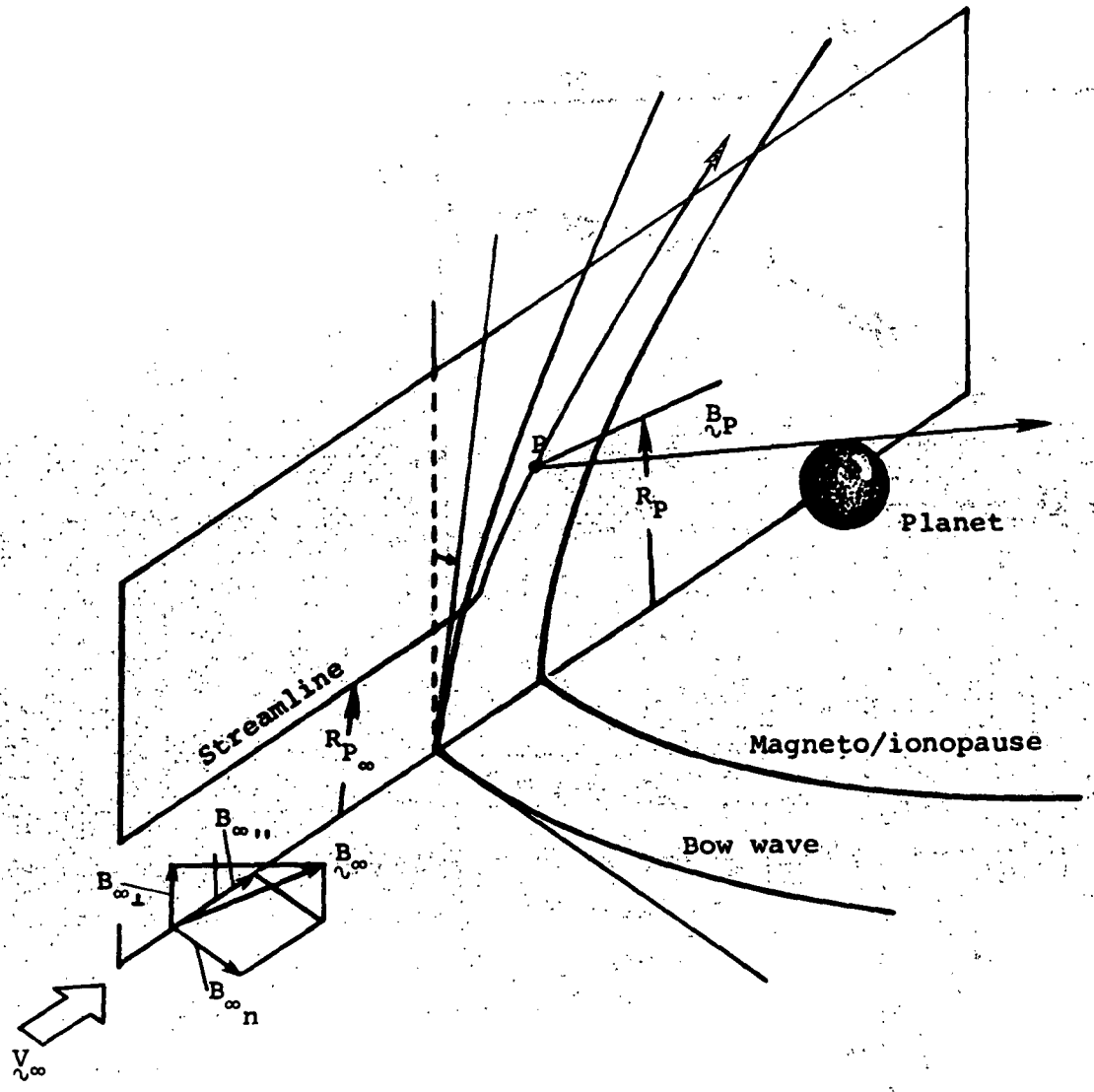


Figure 4. - Illustration of the components of the three-dimensional magnetic field.

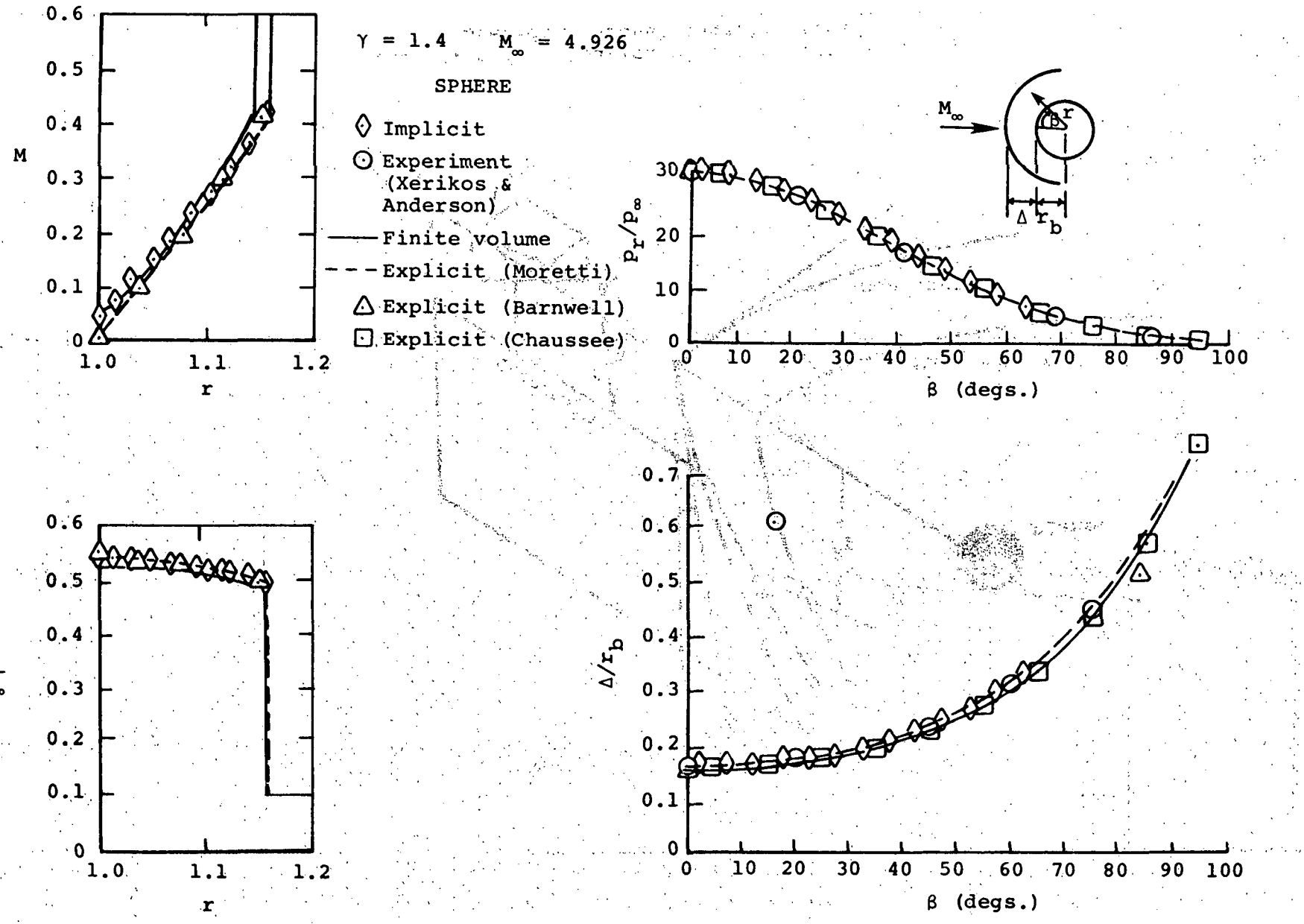
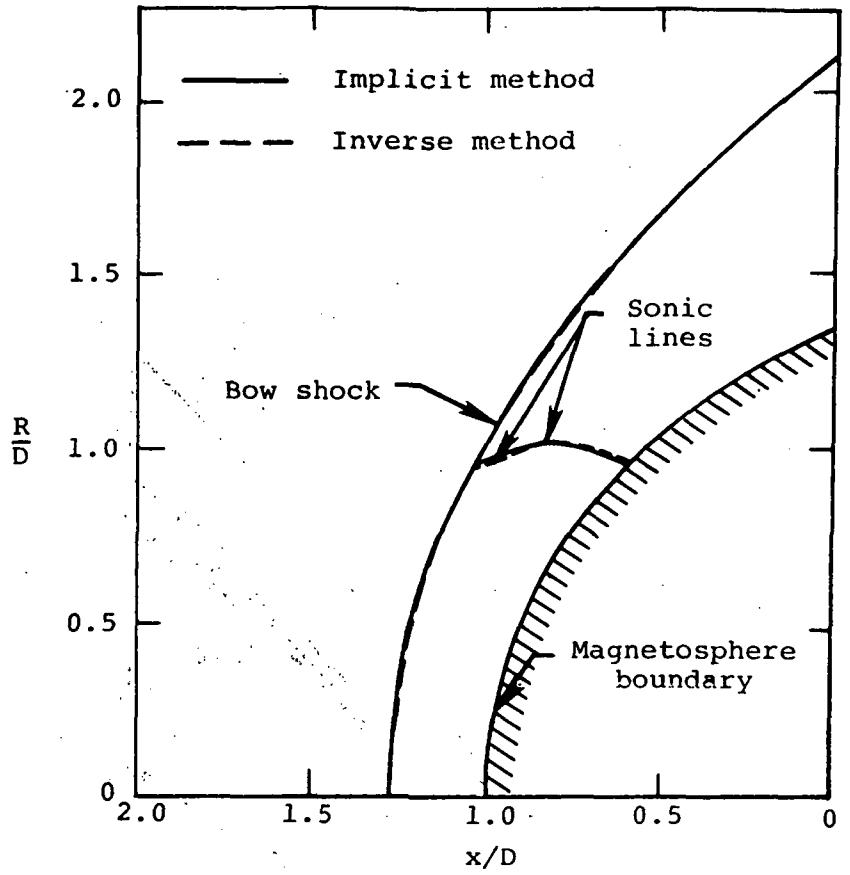
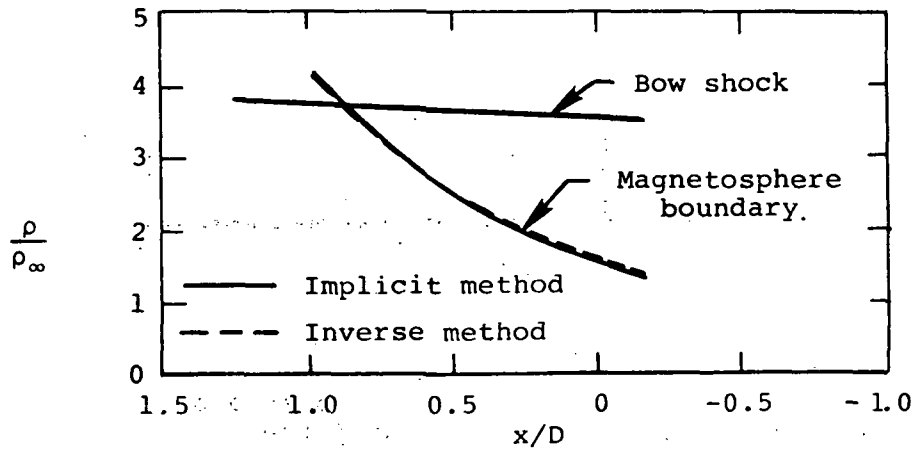


Figure 5. - Comparison of flow properties predicted by the present implicit method with other techniques and experiment for supersonic flow past a sphere; $M_\infty = 4.926$, $\gamma = 1.4$.



(a) Shock shape and sonic line location.



(b) Density distribution.

Figure 6. - Comparison of implicit and inverse methods for shock shape and sonic line location, and density distribution along bow shock and magnetosphere boundary for $M_\infty = 8$, $\gamma = 5/3$ flow past the rotated equatorial trace of the magnetopause.

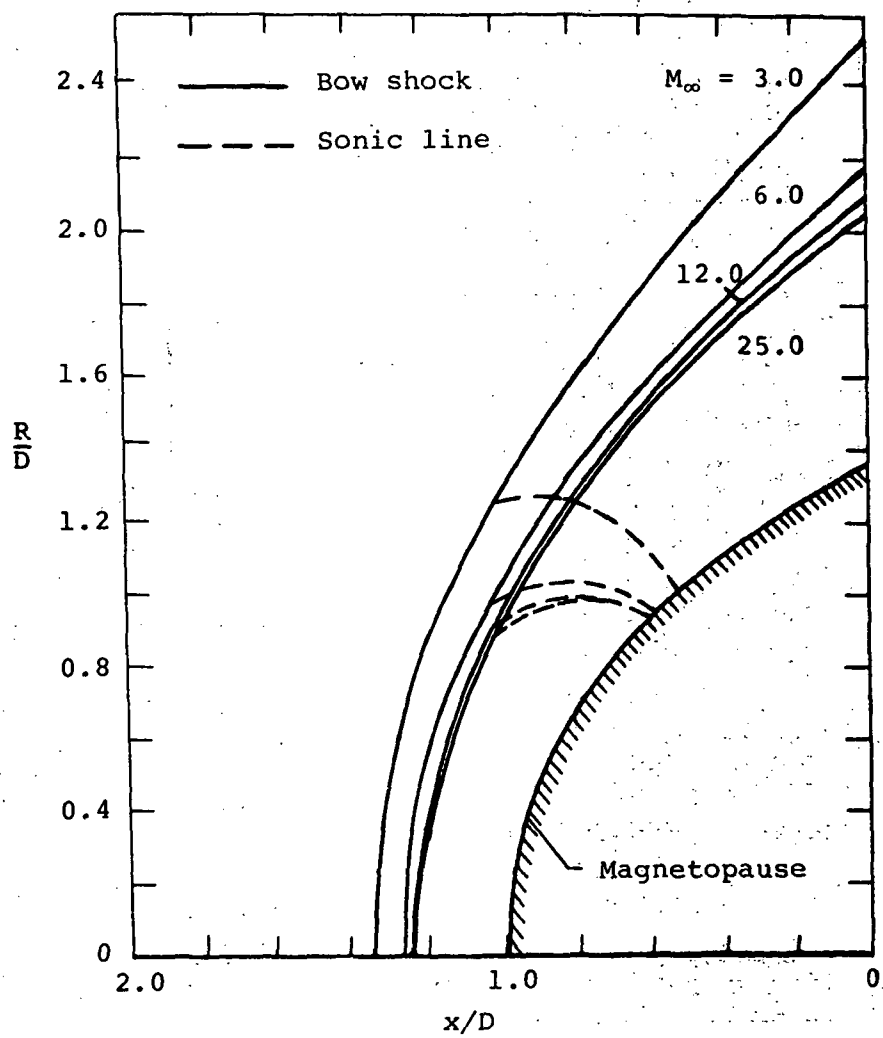


Figure 7. - Bow wave and sonic line locations for various supersonic flows past the rotated equatorial trace of the magnetopause with $\gamma = 5/3$.

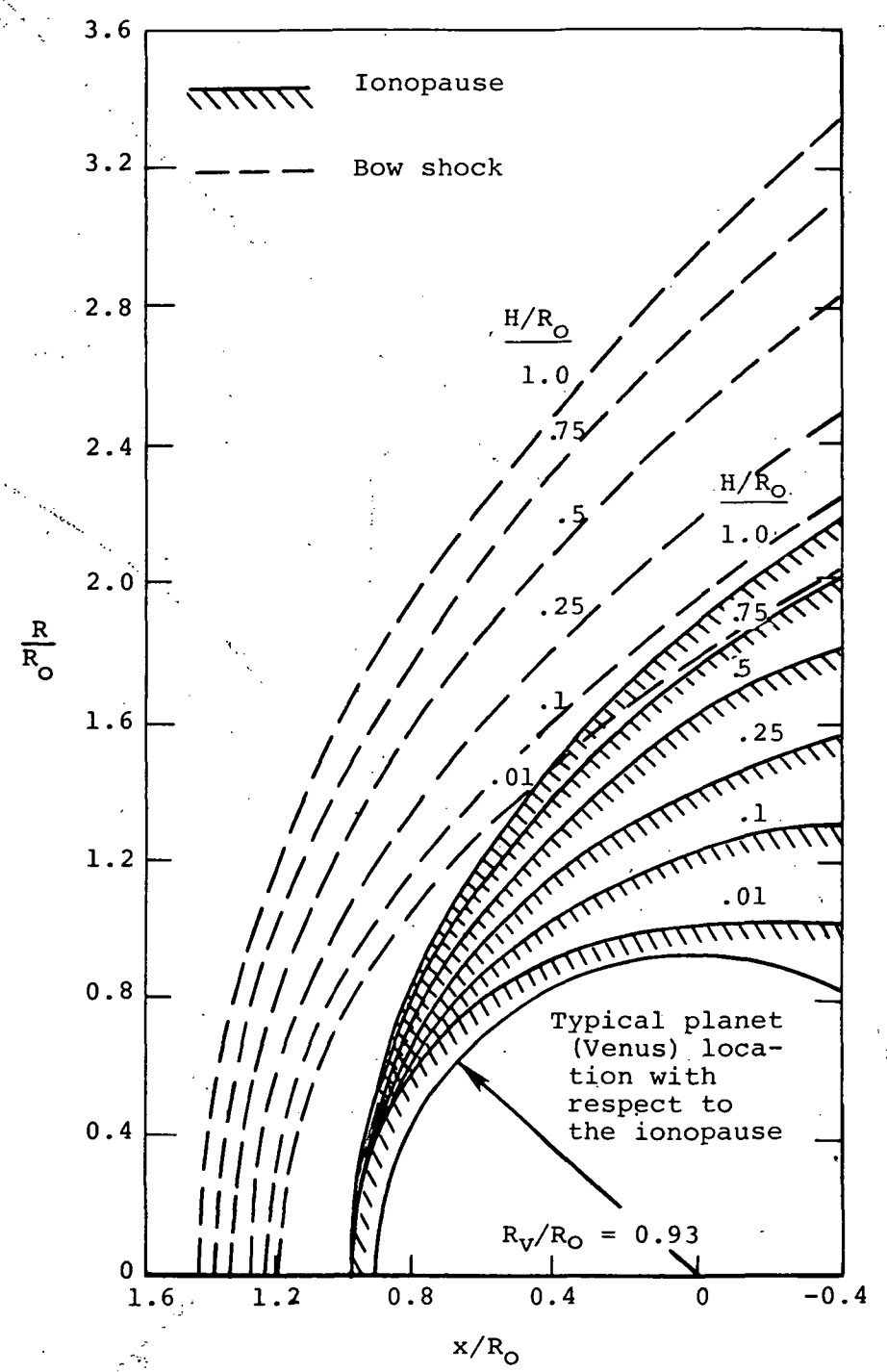


Figure 8. - Bow shock location for solar wind flow with $M_\infty = 8$, $\gamma = 5/3$ past various ionopause shapes.

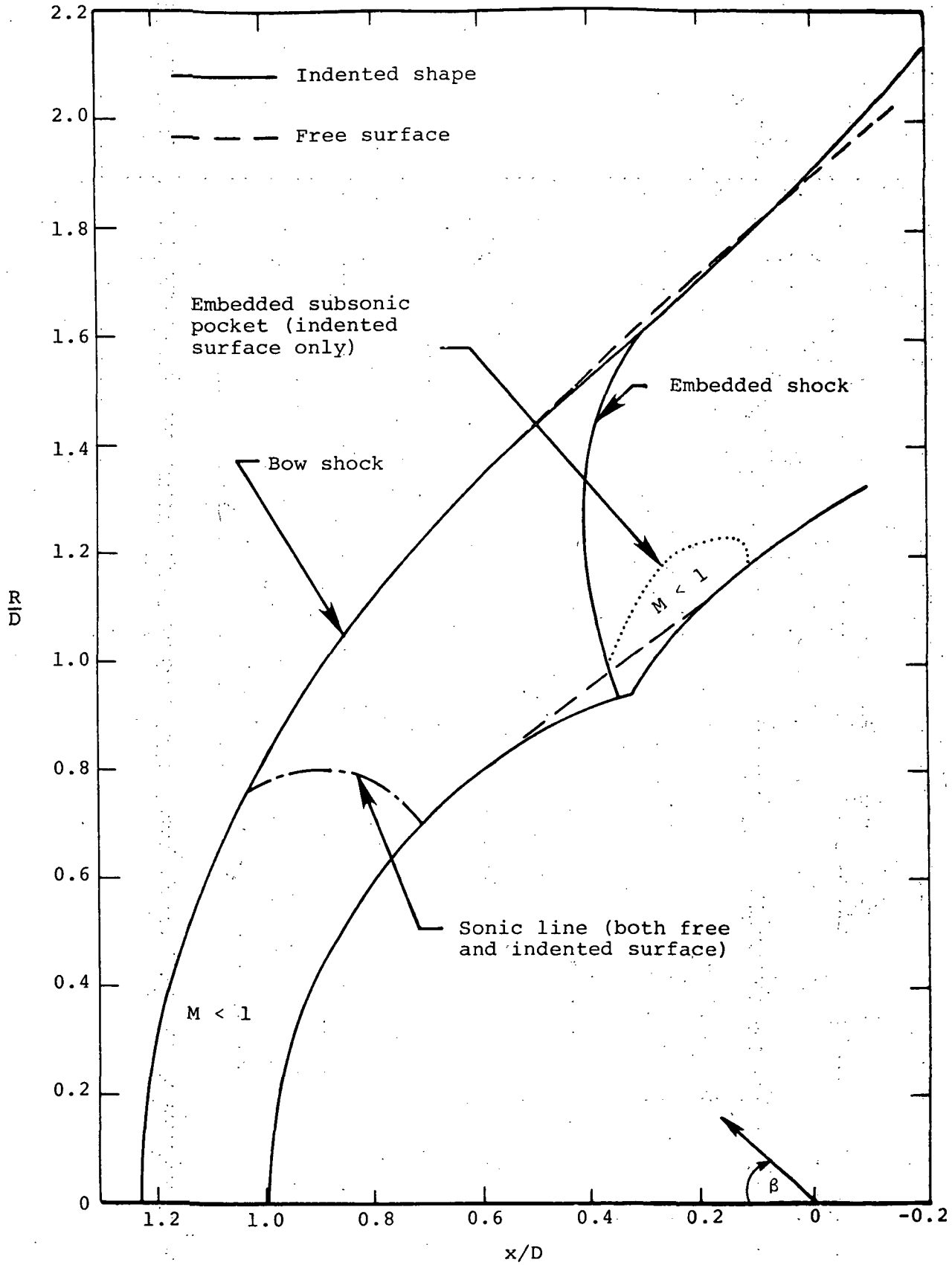


Figure 9. - Bow shock and embedded shock locations for solar wind flow with $M_\infty = 5$, $\gamma = 5/3$ past the rotated principal meridian of the magnetosphere.

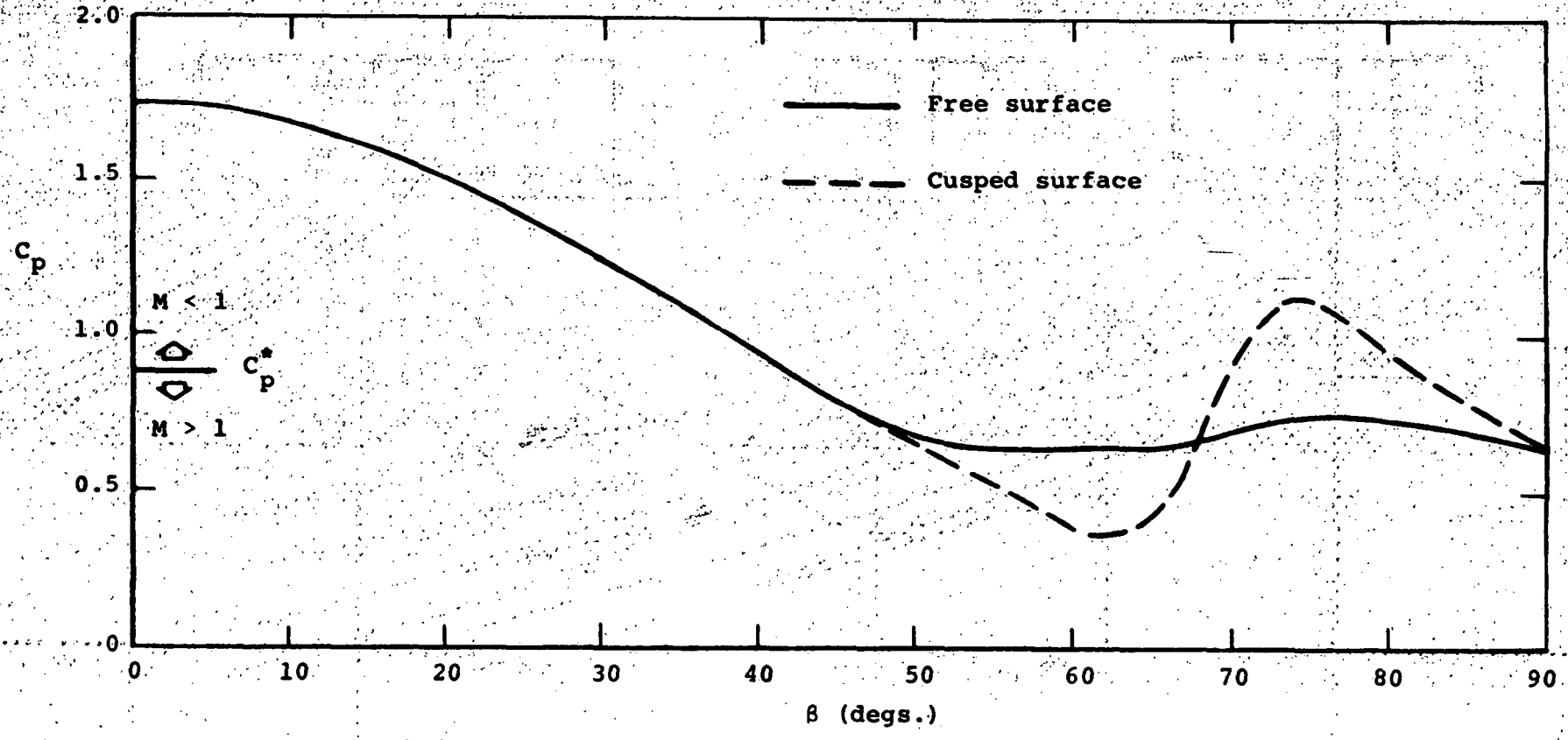
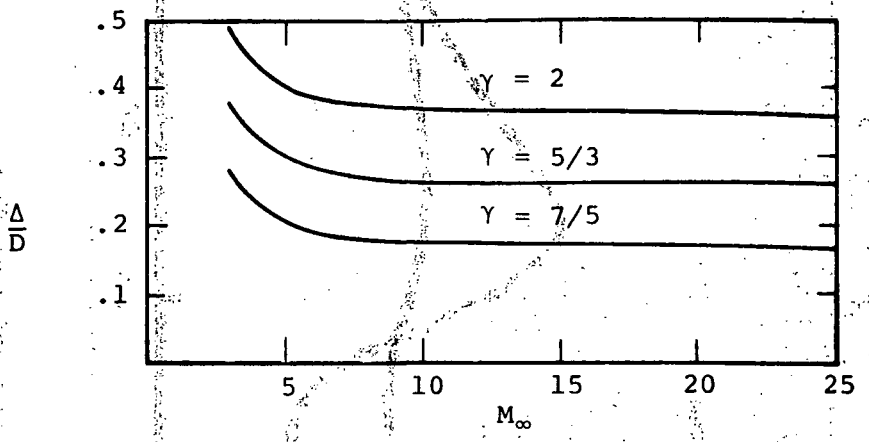
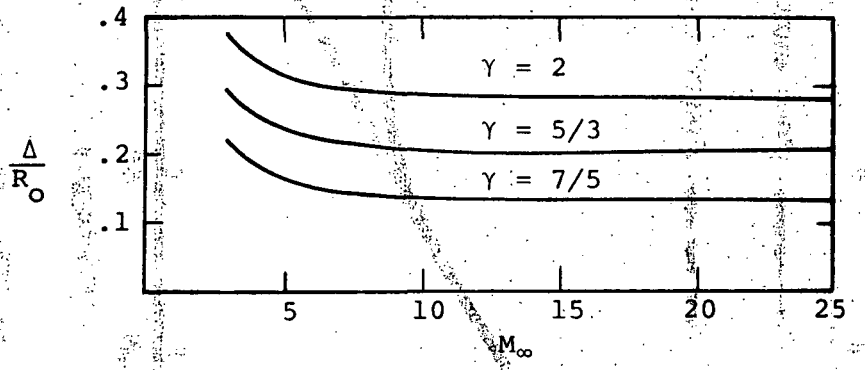


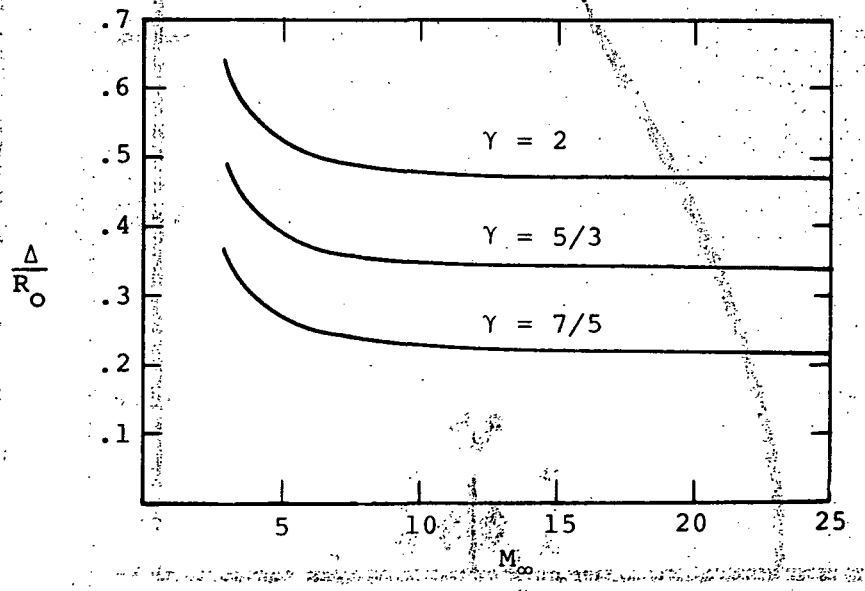
Figure 10. - Magnetopause pressure coefficients for the principal meridian magnetopause shapes shown in figure 9.



(a) Magnetopause equatorial trace



(b) Ionopause trace - $H/R_o = .01$



(c) Ionopause trace - $H/R_o = .5$

Figure 11. - Variation of shock stand-off distance with oncoming Mach number and ratio of specific heats for various magneto/ionopause traces as determined by the present implicit procedure.

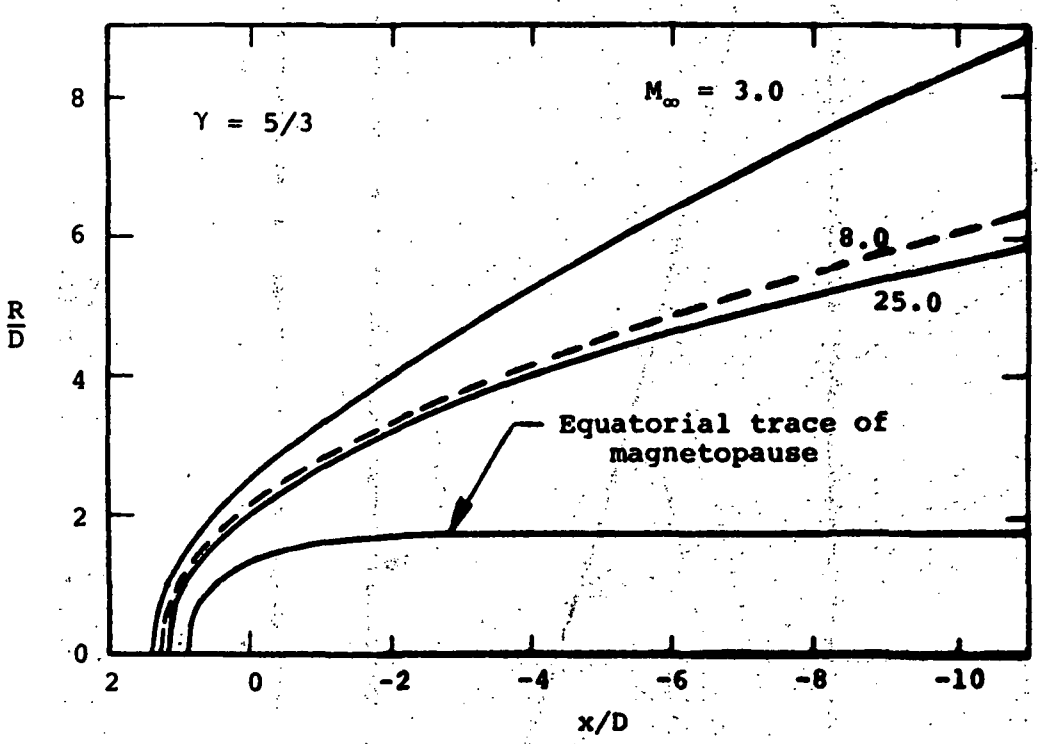


Figure 12. - Shock shapes for various supersonic flows past the rotated equatorial trace of the magnetopause; combined near (blunt body) and far (marching) solutions.

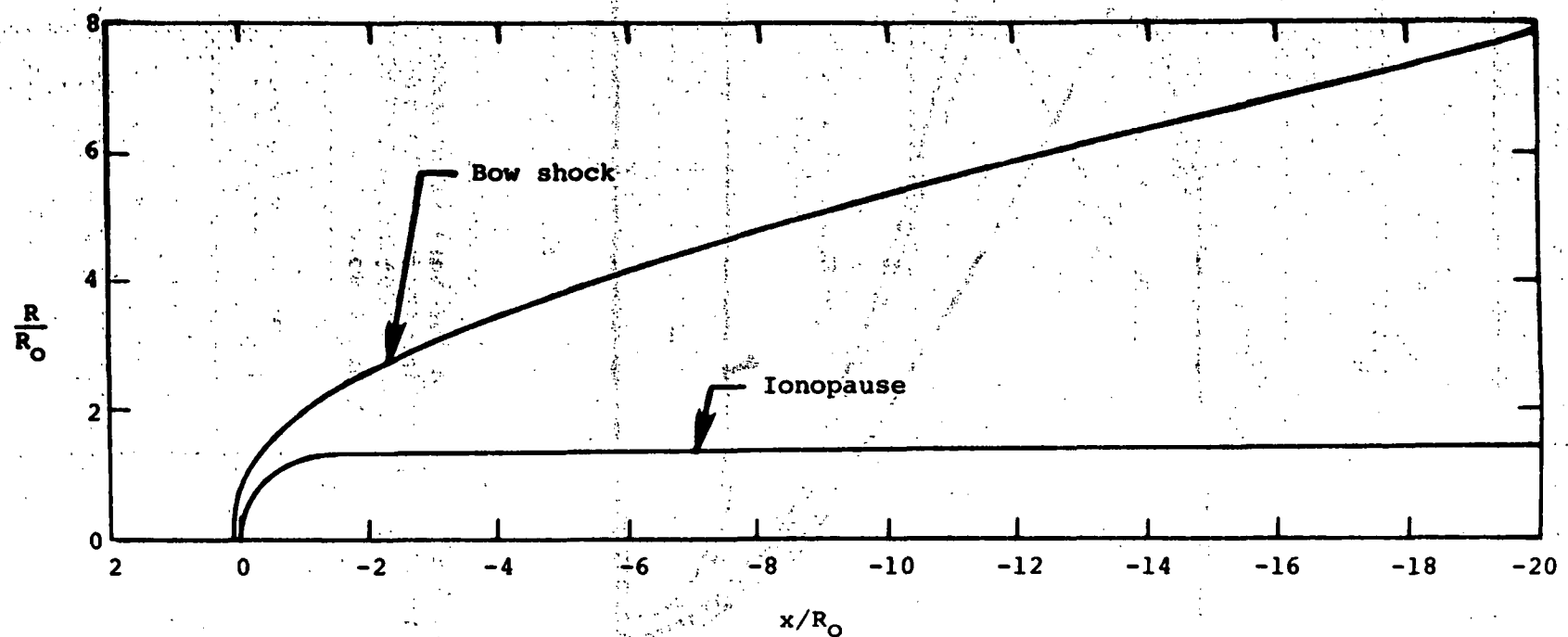
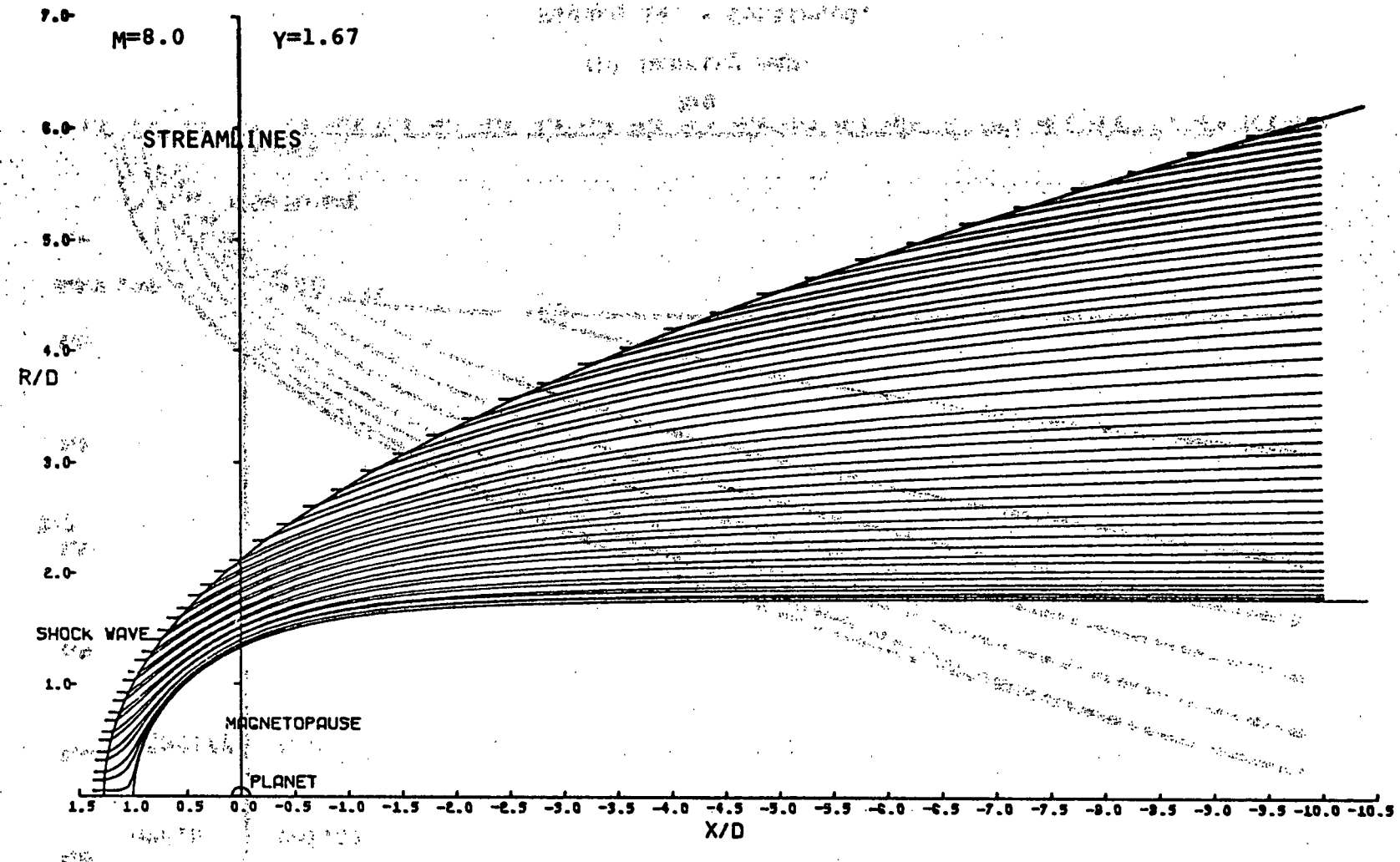
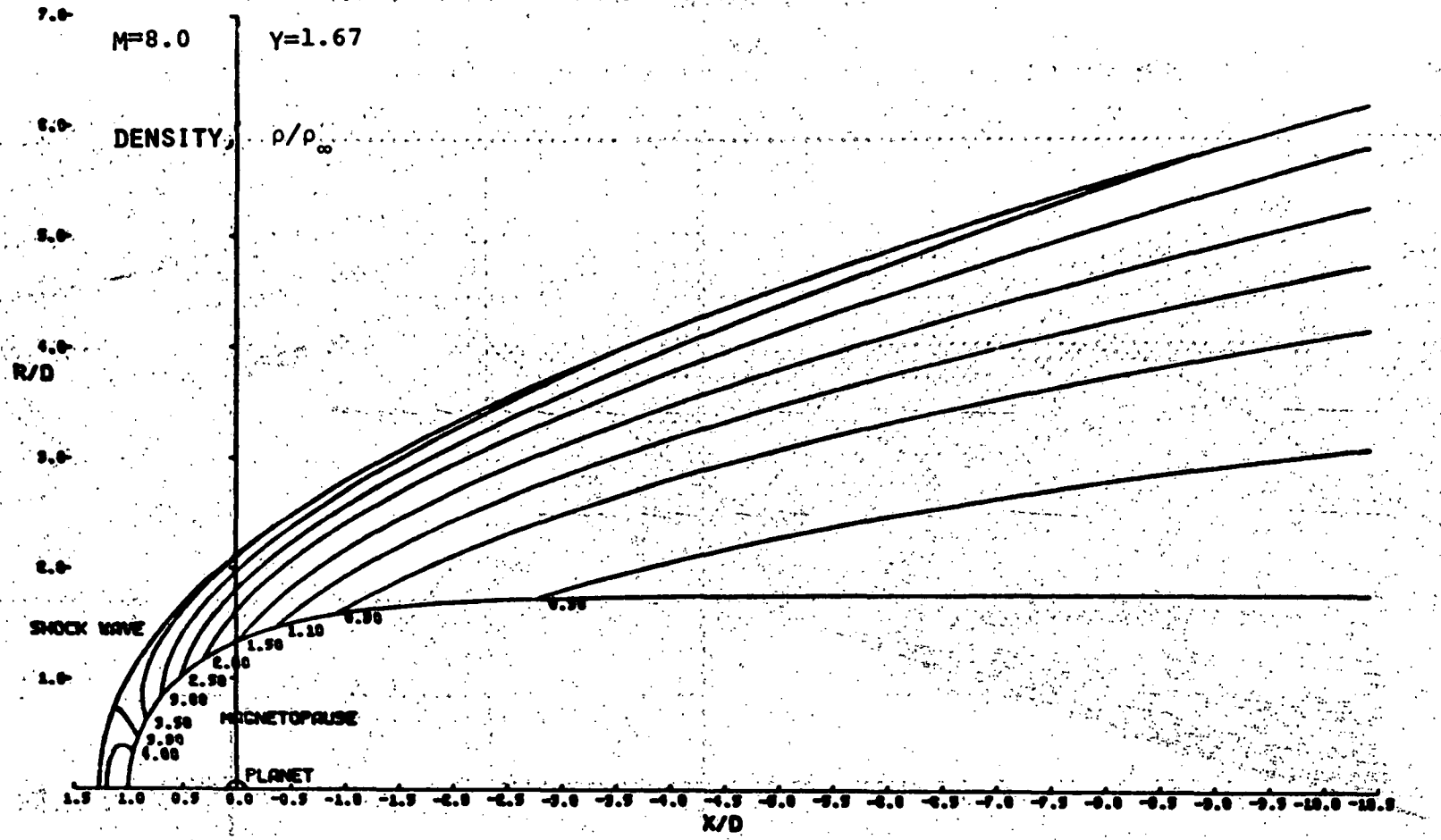


Figure 13. - Shock shape for $M_\infty = 8$, $\gamma = 5/3$ flow past an ionopause shape with $H/R_o = 0.1$; combined near (blunt body) and far (marching) field solutions.



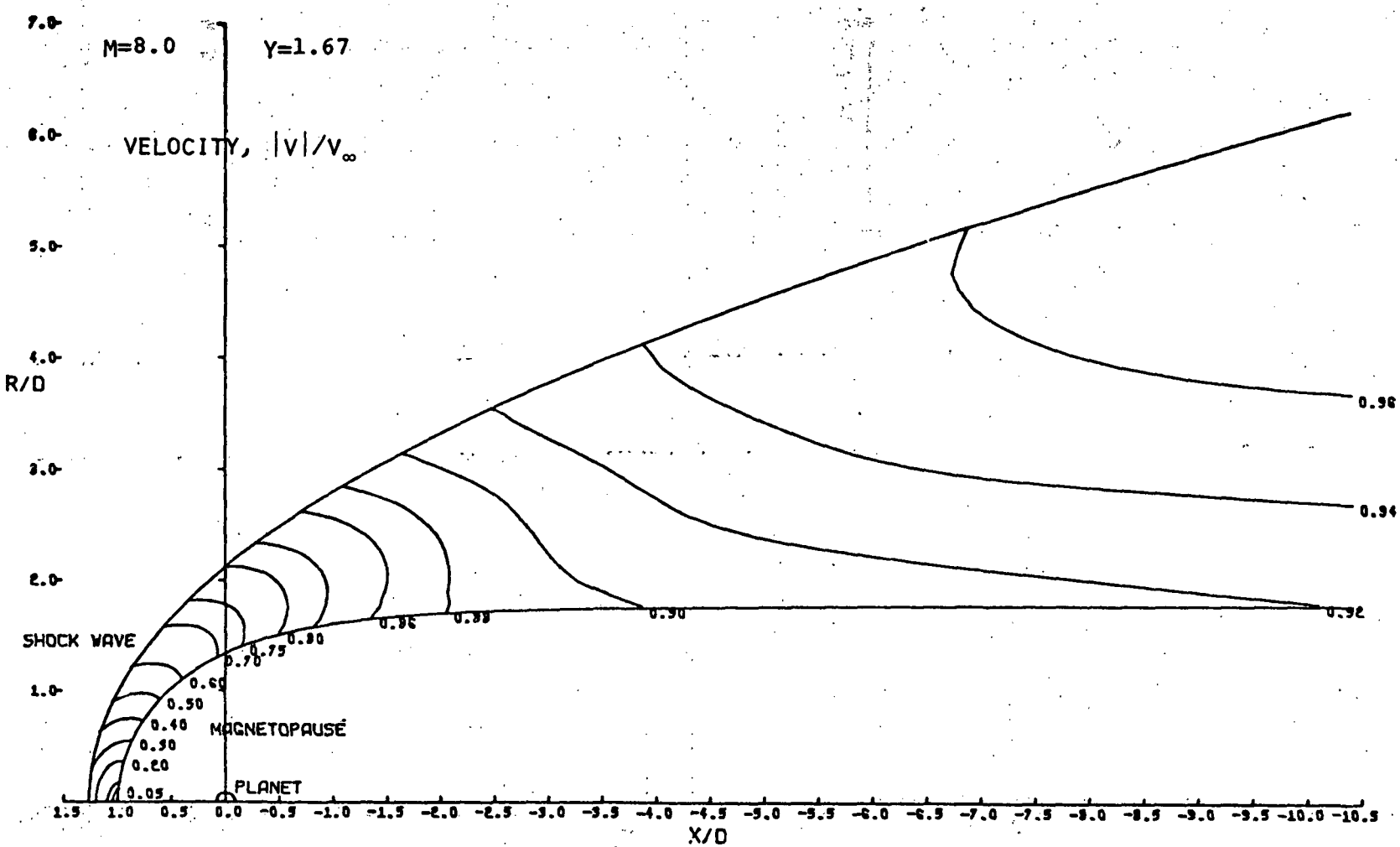
(a) Streamline Map.

Figure 14. - Streamline, density, and velocity maps for $M_\infty = 8.0$, $\gamma = 5/3$ flow past the rotated equatorial trace of the magnetopause; combined blunt body and marching flow field.



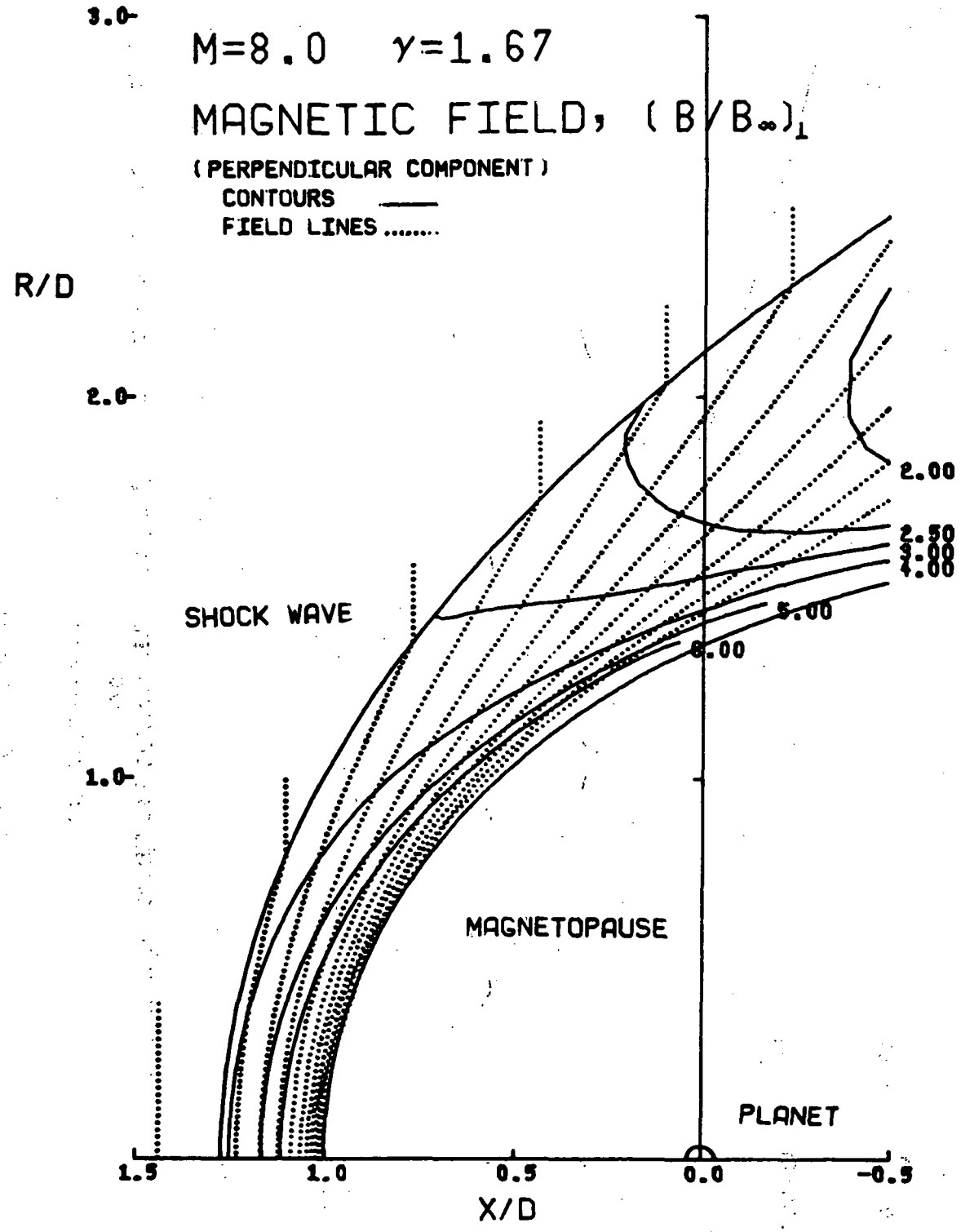
(b) Density map.

Figure 14. - Continued.



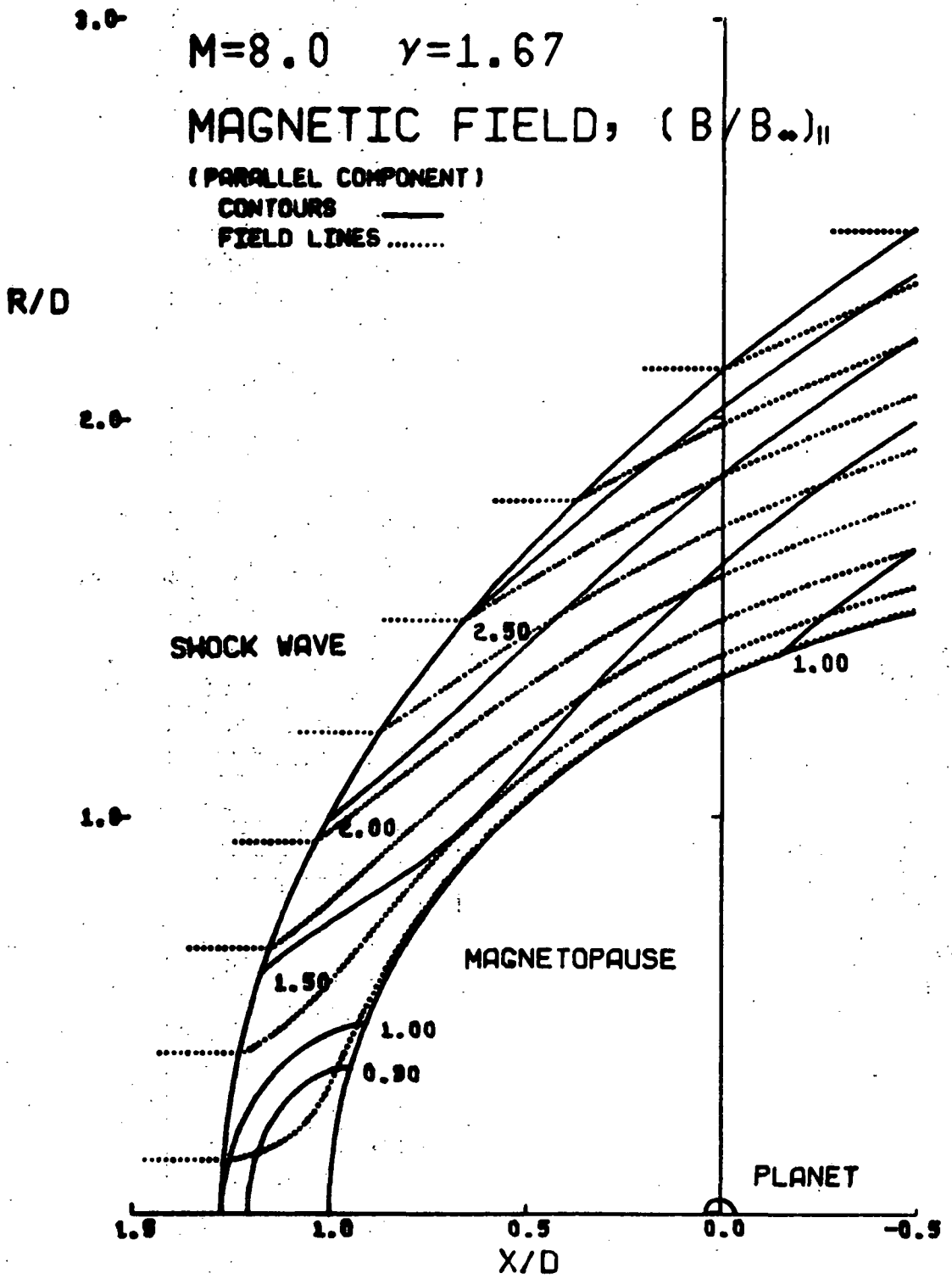
(c) Velocity map.

Figure 14. - Concluded.



(a) Perpendicular component $(\underset{\sim}{B}/B_\infty)_\perp$

Figure 15. - Contours and field line locations of the in-plane magnetic field components $(\underset{\sim}{B}/B_\infty)_\perp$ and $(\underset{\sim}{B}/B_\infty)_{\parallel}$ for $M_\infty = 8$ and $\gamma = 5/3$ flow past the rotated equatorial trace of the magnetopause.



(b) Parallel component $(\underset{\sim}{B}/B_\infty)_{\parallel}$

Figure 15. - Concluded.

*U.S. GOVERNMENT PRINTING OFFICE: 1977 - 735-078/51

National Aeronautics and
Space Administration

Washington, D.C.
20546

Official Business
Penalty for Private Use, $300

THIRD-CLASS BULK RATE

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

NASA

POSTMASTER: If Undeliverable (Section 158
Postal Manual) Do Not Return