NUMERICAL COMPUTATION OF RING-SYMMETRIC SPACECRAFT
EXHAUST LUMES(U) FALCOVITZ (JOSEPH) HAIFA (ISRAEL)
J FALCOVITZ JAN 87 NPS-72-87-001CR N62271-86-M-0214
UNCLASSIFIED
CONTRACTOR REPORT

NUMERICAL COMPUTATION OF RING-SYMMETRIC SPACECRAFT EXHAUST PLUMES

by

Joseph Falcovitz

January 1987

Approved for public release; distribution unlimited.

Prepared for: Strategic Defense Initiative Office
The Pentagon
Washington, DC 20301-7100
The work reported herein was performed for the Naval Postgraduate School by Dr. Joseph Falcovitiz under contract N62271-86-M-0214. The work presented in this report is in support of "Rarefied Gas Dynamics of Laser Exhaust Plume" sponsored by the Strategic Defense Initiative Office/Directed Energy Office. This is a partial report for that contract. The work provides information concerning numerical computation of the flow in spacecraft exhaust plumes. The project at the Naval Postgraduate School is under the cognizance of Distinguished Professor A. E. Fuhs who is principal investigator.

Reproduction of all or part of this report is authorized.

Prepared by:

Dr. Joseph Falcovitiz
Research Contractor

Reviewed by:

Allen E. Fuhs
Distinguished Professor & Chairman
Space Systems Academic Group

Released by:

G. E. Schacher
Dean of Science and Engineering
This report supplements report NPS72-86-003CR. It provides further details about the code JET and the numerical schemes on which it is based: inverse marching characteristic and semi-inverse marching characteristic (SIMA) schemes. The computational procedure is described in some detail. The principles of operation of the code JET are outlined, including a glossary of all major arrays, variables and subroutines. Finally, the full listing of the JET code is reproduced.

**Abstract (Continue on reverse if necessary and identify by block number)**

Laser Exhaust, Exhaust Plume, Method of Characteristics, Inverse Marching, Ring Plumes

---

**DD FORM 1473, 84 MAR**

83 APR edition may be used until exhausted. All other editions are obsolete.
ABSTRACT

This report supplements report NPS72-86-003CR. It provides further details about the code JET and the numerical schemes on which it is based: inverse marching characteristic and semi-inverse marching characteristic (SIMA) schemes. The computational procedure is described in some detail. The principles of operation of the code JET are outlined, including a glossary of all major arrays, variables and subroutines. Finally, the full listing of the JET code is reproduced.
ACKNOWLEDGEMENT

This work was conducted as part of a laser exhaust study under the cognizance of Distinguished Professor Allen E. Fuhs. I deeply appreciate and wish to thank Professor Fuhs for his continuous support and guidance.
# TABLE OF CONTENTS

1. INTRODUCTION ............................................................................................................. 1

2. THE COMPUTATIONAL SCHEME ...................................................................................... 3  
   2.1 Riemann Invariants .................................................................................................. 3  
   2.2 The Integration Scheme for a New Grid Point ......................................................... 4  
   2.3 Boundary Conditions ............................................................................................... 5  
   2.4 Continuum Breakdown Surface ............................................................................... 6  

3. THE JET CODE ................................................................................................................ 7  
   3.1 Main Variables ......................................................................................................... 7  
   3.2 Auxiliary Variables .................................................................................................. 8  
   3.3 Major Parameters ..................................................................................................... 10  
   3.4 Description of JET subroutines .............................................................................. 12  

4. THE JET CODE LISTING ............................................................................................... 20  

5. REFERENCES .................................................................................................................. 45  

6. DISTRIBUTION LIST ..................................................................................................... 46
NOMENCLATURE followed by units (if any) and CODE NOTATION (if any)

\begin{itemize}
  \item \textbf{a} \quad \text{sound speed (m sec}^{-1}\text{)}
  \item \textbf{B} \quad \text{breakdown parameter [5,6,7]}
  \item \textbf{C} \pm \quad \text{characteristic lines inclined at (} \theta \pm \mu \text{)}
  \item \textbf{D} \quad \text{molecular diameter (hard spheres) (m)}
  \item \textbf{M} \quad \text{Mach number}
  \item \textbf{n} \quad \text{number density (molecules/m}^3\text{)}
  \item \textbf{p} \quad \text{pressure (Pa)}
  \item \textbf{S} \quad \text{coordinate along streamlines (m)}
  \item \textbf{u} \quad \text{flow velocity (m/sec)}
  \item \textbf{x} \quad \text{axial cartesian coordinate}
  \item \textbf{y} \quad \text{radial cartesian coordinate}
  \item \textbf{\gamma} \quad \text{specific-heat ratio (G)}
  \item \textbf{\eta} \quad \text{length coordinate along fan characteristics (} \textbf{C}^+ \text{) (m)}
  \item \textbf{\theta} \quad \text{inclination of flow velocity vector}
  \item \textbf{\lambda}_0 \quad \text{mean free path at stagnation conditions (m)}
  \item \textbf{\mu} \quad \text{Mach angle (} \sin \mu = 1/\textbf{M} \text{) (MU)}
  \item \textbf{v} \quad \text{Prandtl-Meyer function (NU)}
  \item \textbf{\xi} \quad \text{length coordinate along transverse (} \textbf{C}^- \text{) characteristic}
  \item \textbf{\sigma} \quad \text{collision cross-section } \pi \textbf{D}^2 \text{ (m}^2\text{)} (\text{SIGMA})
  \item \textbf{\tau} \quad \text{molecular opacity (expected number of collisions by a fast invading molecule) (XI)}
  \item \textbf{\varphi} \quad \text{collision frequency (sec}^{-1}\text{)}
  \item \textbf{\omega} \quad \text{symmetry index (0 - planar flow, 1 - axisymmetric flow) (DELTA)}
  \item \textbf{\Gamma} \quad \text{the fraction } \left[ (\gamma + 1)/(\gamma - 1) \right]^{1/2}
  \item \textbf{(v + \theta)} \quad \text{Riemann invariant along } \textbf{C}^- \text{ (RM)}
  \item \textbf{(v - \theta)} \quad \text{Riemann invariant along } \textbf{C}^+ \text{ (RP)}
\end{itemize}

INDICES

\begin{itemize}
  \item \textbf{( )}_0 \quad \text{a specific point in the CRW (} \textbf{x}_0,\textbf{y}_0 \text{) (Also: stagnation conditions)}
  \item \textbf{( )}_l \quad \text{nozzle exit conditions}
  \item \textbf{( )}_L \quad \text{limiting CRW characteristic (} \textbf{p} = 0 \text{)}
  \item \textbf{( )}_r \quad \text{final CRW characteristic (boundary of numerical integration)}
  \item \textbf{( )}_c \quad \text{corner of CRW}
\end{itemize}
EMPTY PAGE
1. INTRODUCTION

In a recent report [1] a mixed numerical/analytical approach to the computation of a ring-symmetric spacecraft exhaust plume was presented. The numerical scheme had been implemented in a code named “JET” which is capable of generating whole-plume flow fields, while the analytic approximation is restricted to the ring-symmetric centered rarefaction waves (CRW) that flank the plume. The present report is intended to serve as a supplement to [1] in providing details on the computational scheme and the code JET.

The spacecraft exhaust flow (Fig. 1 of [1]) is idealized as a ring-symmetric steady isentropic expansion of an ideal gas. The nozzle lips are assumed sharp; the supersonic flow from the exit surface of the ring-nozzle is assumed uniform, and the background is considered to be perfect vacuum.

The standard scheme for computing such idealized ring-plumes is the classical (direct) method of characteristics [2]. At a preliminary phase of the present laser exhaust study, a code AXSYM [3] was written for computing ring-plumes using this method. A notorious shortcoming of the direct method of characteristics is that the solution grid is highly irregular, being formed by the (oblique) intersection of the \( C^+ \) and the \( C^- \) families of characteristic lines. We first encountered a difficulty with this grid while seeking a scheme for integrating the molecular opacity along a straight line [1]. This computation would have required rather complex coding for the geometry of intersection between a straight line and an irregular grid. It seemed preferable to opt for a computation scheme that would produce a more regular grid, even at the expense of some loss of accuracy. Such scheme is the inverse marching method of characteristics [4].

Generally the marching in this type of scheme is in the downstream direction, i.e., the \( y \) direction in our case. Grid points are located on a succession of constant-\( y \) rows, thereby introducing a measure of regularity in the solution grid. Just two rows have to be stored in the computer core memory - the "old" row and the "new" row, whereas in the direct method of characteristics whole grid-image matrices are required to reside simultaneously in core memory.

The first version of the JET code was based on the inverse marching scheme given by Zucrow and Hoffman (Section 12-5 in [4]), where the flow variables were the two cartesian velocity components. The computation seemed accurate everywhere, except within the centered rarefaction wave (CRW). In an attempt to replicate a planar CRW (Prandtl-Meyer flow), the numerical solution exhibited an
instability: Mach number increased along the (low pressure) boundary characteristic line, rather than remain constant.

A qualitative explanation for this instability is the following. Flow gradients in a CRW are inversely proportional to distance from the corner, so that the inverse marching scheme gives rise to an amplification of interpolation errors at every marching step, leading to an apparently divergent (unstable) numerical solution. Increasing the order of interpolation from linear to cubic did not eliminate the instability.

Looking for a scheme that would replicate a planar CRW accurately, we tried the modified marching idea as presented by Zucrow and Hoffman for 1-D time-dependent flows (Sections 19-6(a) and 19-6(j) of [4]). In this scheme new grid points are determined by forwardly extending a "primary" family of continuous characteristic lines from old grid points. The primary family in a CRW is the characteristics fanning out from the corner (we assume it is the $C^+$ family). By choosing this modified scheme, the interpolation for trace points obtained from reversely extended $C^+$ lines was eliminated. However, the corresponding interpolation for the transverse $C^-$ characteristics remained, and with it the aforementioned instability.

In order to replicate a planar CRW, we had to replace the flow variables by the Riemann invariants $(v \pm \theta)$. In a $C^+$ planar CRW, the Riemann invariant $(v + \theta)$ is uniformly constant, so that the interpolation in $(v + \theta)$ due to reversely extending $C^-$ characteristics introduces no error at all. This scheme, which we named SIMA (Semi Inverse Marching Algorithm), was indeed verified to replicate a planar CRW exactly, when implemented in the code JET.

The plan of this report is the following. In Ch. 2 we supplement the description of the numerical scheme given in Ch. 2 of [1], by adding more details on the computational procedure. A description of the code JET is given in Ch. 3, and the code listing is reproduced in Ch. 4.

Note on symmetry:
The code JET has two symmetry options. When DELTA = 1 a ring-symmetry is in effect; when DELTA = 0, a planar symmetry is in effect. An axisymmetric jet exiting in the y direction from the same nozzle aperture along the x axis can readily be computed by replacing all terms in the code that correspond to $\sin(\theta)/y$ in the compatibility equations (2.1-1), by $\cos(\theta)/x$. In that case the coding is virtually unchanged, and the only care that should be exercised is for the difference equations for new grid points on or near the y axis. Also, all reference to the analytic approximation of the ring-symmetric CRW [1] should be deleted in this case, as it is designed specifically for ring-symmetry.
2. THE COMPUTATIONAL SCHEME

A basic description of the semi inverse (SIMA) and inverse marching schemes was given in Ch. 2 of [1]. We supplement this description by specifying the slightly modified definition of Riemann invariants in the code, and by giving information about some ancillary computations.

2.1 Riemann Invariants

The compatibility equations whose integration constitutes the numerical solution to the governing equations [1] are expressed in terms of the Riemann invariants as follows:

\[
\begin{align*}
\text{Along } C^+ & \quad \quad (\nu - \theta)_4 = (\nu - \theta)_2 + \omega \sin \mu_{24} \sin \theta_{24} \Delta \eta / y_{24} \\
\text{Along } C^- & \quad \quad (\nu + \theta)_4 = (\nu + \theta)_1 + \omega \sin \mu_{14} \sin \theta_{14} \Delta \xi / y_{14}
\end{align*}
\] (2.1-1)

The Riemann invariants \((\nu \pm \theta)\) are modified for convenience, by adding a constant to both \(\nu\) and \(\theta\). The new definitions of \(v(M)\) and \(\theta\) are:

\[
\begin{align*}
v(M) &= -\Gamma \arctan(\Gamma q) + \arctan(q) \\
q &= (M^2 - 1)^{-1/2} \\
\theta &= \theta - \theta_L
\end{align*}
\] (2.1-2)

Thus, in a Prandtl-Meyer flow with entry Mach number of \(M_1\), the modified values of both \(v(M)\) and \(\theta\) vanish as \(M \to \infty\). As a consequence, in a \(C^+\) Prandtl-Meyer flow the modified invariant \((\nu + \theta)\) vanishes uniformly. In this modified form, the computation of \(M\) from \(v(M)\) is readily done by performing standard Newton-Raphson iterations (in RFUNC), using the derivative:

\[
v'(q) = -(\Gamma^2 - 1) \left[ (1 + \Gamma q^2)(1 + q^3) \right]^{-1}
\] (2.1-3)
2.2 The Integration Scheme for a New Grid Point

The integration scheme has been sketched in Ch. 2 of [1]. It is performed in INVMAR for inverse marching points or in SEMINV for semi-inverse marching (SIMA) points. The computational scheme is specified via the following seven-step procedure:

INVMAR (Inverse Marching)
(a) Grid: At this stage the new grid point has already been defined.
(b) Predictor: Flow variables are the interpolated (linear nearest-neighbor) value on the old row for a point having the new grid x coordinate (x₄).
(c) Centered variables: Denote the Riemann invariants by

\[ \begin{align*}
RM &= (\nu + \theta) \\
RP &= (\nu - \theta)
\end{align*} \]  

(2.2-1)

then centered values for segments (1,4) and (2,4) (using code notation) are:

\[ \begin{align*}
RM_{14} &= (RM_{1} + RM_{4})/2 \\
RP_{14} &= (RP_{1} + RP_{4})/2 \\
RM_{24} &= (RM_{2} + RM_{4})/2 \\
RP_{24} &= (RP_{2} + RP_{4})/2
\end{align*} \]  

(2.2-2)

All other centered flow variables are computed from the centered Riemann invariants by calling RFUNC.
(d) Inverse Extension: old trace points x₁, x₂ are evaluated from the geometrical relations

Along C⁻ \[ \begin{align*}
y_{\text{new}} - y_{\text{old}} &= (x_{5} - x_{4}) \tan(\theta_{14} - \mu_{14})
\end{align*} \]  

(2.2-3)

Along C⁺ \[ \begin{align*}
y_{\text{new}} - y_{\text{old}} &= (x_{4} - x_{2}) \tan(\theta_{24} + \mu_{24})
\end{align*} \]

(e) Interpolation: find Riemann invariants RM, RP at old trace points x₁ and x₂ through nearest-neighbor linear interpolation by calling INTERP.
(f) Integration: Using the compatibility relations in finite-difference form (2.1-1) with segment-centered coefficients, compute iteration-updated values of Riemann invariants at new grid point.
(g) Corrector: if values of Riemann invariants and old trace points \( x_1, x_2 \) are not sufficiently convergent, resume the procedure at step (c) above.

SEMINV (Semi Inverse Marching - SIMA)
(a) Grid: New grid point \( (x_4) \) is determined as part of the SIMA scheme at step (d) below.
(b) Predictor: Flow variables are those of point \( (x_2, y_{old}) \).
(c) Centered variables: Identical to step (c) above.
(d) Semi-Inverse Extension: new grid point \( x_4 \) and old trace point \( x_1 \) are evaluated from the geometrical relations in Eq. (2.2-3) above.
(e) Interpolation: find Riemann invariants RM, RP at old trace point \( x_1 \) through nearest-neighbor linear interpolation by calling INTERP.
(f) Integration: Identical to step (f) above.
(g) Corrector: Identical to step (g) above, except for replacing \( x_2 \) in the convergence test by \( x_4 \).

2.3 Boundary Conditions

On the vacuum side the boundary conditions \( (p = 0) \) can only be approximately implemented in a method of characteristics scheme. We do so by ending the computation on a certain "final" \( C^+ \) fan characteristic line that starts out with a sufficiently high Mach number \( M_f \) at the corner (typically \( M_f = 34 \)). The marching computation of new grid points on the boundary \( C^+ \) characteristic via the SIMA scheme is identical to that of \( C^+ \) characteristics within the ring-symmetric CRW. It is noted that under this boundary scheme some outflow takes place through the boundary characteristic line, so that the total mass flow through a row \( y = y_{new} \) decreases slightly as \( y_{new} \) increases.

At the nozzle exit the boundary conditions are assumed to be uniform outflow in the radial (\( y \)) direction with Mach number \( M_1 \). At the nozzle lip, the SIMA integration starts out from a presumed planar CRW (Prandtl-Meyer flow) at the corner (i.e., the associate CRW in the terminology of Ch. 3 in [1]).

At the plane of symmetry (\( x = 0 \)) the boundary condition is simply \( \theta = \pi/2 \). However, this condition is implemented indirectly, by assuming that the flow at virtual grid points with \( x < 0 \) is a mirror-image of the flow at the corresponding \( x > 0 \) points. The reason is that when a new grid point of \( x_4 = 0 \) or of \( x_4 \) sufficiently close to zero is considered for inverse-marching integration, the inversely extended trace point \( (x_1, y_{old}) \) can be at \( x < 0 \). Considering the subtraction of \( \theta_L \) from \( \theta \) as in Eq.(2.1-2), the reflection rules are:
\[
\begin{align*}
RM & \rightarrow RP \: + \: (\pi - 2\theta_L) \\
RP & \rightarrow RM \: - \: (\pi - 2\theta_L) 
\end{align*}
\] (2.3-1)
where values on the left and right of the \(\rightarrow\) symbol correspond to values left and right of \(x = 0\). This boundary condition is implemented in INTERP.

2.4 Continuum Breakdown Surface

As an informative option, the code JET can compute (in PLUMES) points on a surface of continuum breakdown \([5, 6, 7]\), which is defined as a line of constant \(B\), where \(B\) is given by:

\[
B = -\left(\frac{u}{\varphi}\right) \rho^{-1} \left(\frac{dp}{dS}\right)
\]

(2.4-1)

\[
\varphi = 4(\pi \gamma^{-1/2} \sigma n a
\]

When the standard isentropic relations for \(\rho\) and \(n\) in terms of \(M\) are substituted in (2.4-1), the flow speed is expressed as \(u = Ma\) and the streamwise gradient of \(M\) is expressed in cartesian coordinates, we get:

\[
B = \lambda_0 \left(\frac{\pi \gamma}{8}\right)^{1/2} M^2 \left[1 + \left(\frac{\gamma - 1}{2}\right)M^2\right]^{1/(\gamma - 1) - 1} \left[M_x \cos \theta + M_y \sin \theta\right]
\]

(2.4-2)

\[
\lambda_0 = \left(2^{1/2} \sigma n_0\right)^{-1}
\]

Note that the sign of \(B\) has been chosen as positive for expansion flows. This definition is preferred to taking an absolute value of the flow gradient, since it assures proper interpolation of \(B\) even if its spatial distribution goes through \(B = 0\).

Due to the dependence of \(B\) on a spatial gradient, its numerical evaluation (see BREAK) is attributed to mid-grid points both in \(x\) and in \(y\).
3. THE JET CODE

In this chapter we provide a concise description of the JET code according to its version at the time of the JET018 run. This description is intended as an aid in reading the code listing which is given in Ch. 4.

The plan of this chapter is as follows. Array variables that constitute the mainstay of the computational scheme are described in Section 3.1. Auxiliary array variables that are used primarily for processing the information generated by the numerical scheme, are described in Section 3.2, followed in Section 3.3 by a list of major parameters that control the computation (some of them also serve as run data). Finally, all subroutines are listed and described in Section 3.4.

3.1 Main Variables

The array variables used for the computational scheme are organized in two labeled COMMON groups. The first group /VECS/ is designed to hold two grid rows - the old row designated by suffix F and the new row designated by suffix N. The second group /CHARAC/ are characteristic-indexed arrays that hold information about continuous characteristic lines. This characteristic information is used in two ways: it is incorporated in the SIMA computational scheme for the CRW region, and it is used to store data for optional plotting of characteristic lines (see PLUMES and PRINT).

The basic organization is that the new arrays (suffix N) are those in which values are stored during the course of the marching computational procedure. At the end of each marching step, values are transferred from new arrays to old arrays (suffix F); this is done in MOVE. In the array listing below, we indicate in parenthesis the subroutine (or subroutines) in which that new array is defined.

<table>
<thead>
<tr>
<th>/VECS/</th>
<th>Description</th>
<th>Subroutine(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XN(I)</td>
<td>x coordinate of grid point I.</td>
<td>(GRIDN)</td>
</tr>
<tr>
<td>RMN(I)</td>
<td>modified Riemann invariant ((v + \theta)) at grid point I.</td>
<td>(BEGIN, INVMAR, LOADC).</td>
</tr>
<tr>
<td>RPN(I)</td>
<td>modified Riemann invariant ((v - \theta)) at grid point I.</td>
<td>(BEGIN, INVMAR, LOADC).</td>
</tr>
<tr>
<td>MN(I)</td>
<td>Mach number at grid point I.</td>
<td>(BEGIN, INVMAR, LOADC).</td>
</tr>
<tr>
<td>MUN(I)</td>
<td>Mach angle (\mu) at grid point I.</td>
<td>(BEGIN, INVMAR, LOADC).</td>
</tr>
<tr>
<td>TETAN(I)</td>
<td>true (unmodified) flow angle (\theta) at grid point I.</td>
<td>(BEGIN, INVMAR, LOADC).</td>
</tr>
</tbody>
</table>
BN(I) value of breakdown parameter B at point I-1/2 (and at half a marching step back in y as well). (BREAK).

XTEMP(I) used for auxiliary computation of I-1/2 grid points in PLUMES.

/CHARAC/

XCHARN(KC) x coordinate of point on characteristic line number KC. (BEGIN, SEMINV, PLUMES).

YCHARN(KC) y coordinate of point on characteristic line number KC. (BEGIN, SEMINV, PLUMES).

RM Carn(KC) modified Riemann invariant \((v + \theta)\) of point on characteristic line number KC. (BEGIN, SEMINV).

RPCARN(KC) modified Riemann invariant \((v - \theta)\) of point on characteristic line number KC. (BEGIN, SEMINV).

TCHARN(KC) true (unmodified) flow angle \(\theta\) at point on characteristic line number KC. (BEGIN, SEMINV).

MUCARN(KC) Mach angle \(\mu\) at point on characteristic line number KC. (BEGIN, SEMINV).

CSIGNN(KC) sign of characteristic line number KC. It has value 1 for \(C^+\) and value \(-1\) for \(C^-\). Note that upon reflection of a \(C^+\) line from the symmetry plane \((x=0)\), the sign value is changed from 1 to \(-1\). (BEGIN, SEMINV).

MCHARN(KC) Mach number at point on characteristic line number KC. (BEGIN, SEMINV).

MCHARI(KC) Mach number at Prandtl-Meyer's fan characteristic number KC at the corner. It is defined initially and is not changed during the run. (BEGIN).

3.2 Auxiliary Variables

In addition to the major arrays mentioned above, there are several groups of auxiliary arrays that do not affect the computational scheme, but are intended for informative processing of the results. These groups are /PLUME/, /IPLUME/, /THICKY/, /THICKX/, /GRP/. /PLUME/ is used to preserve points on special lines for later plotting (in a separate code). /THICKY/ and /THICKX/ are for storing values of radial (y) and lateral (x) molecular opacities. The group /GRP/ is used in conjunction with comparative computation of the ring-symmetric CRW flow according to the analytic approximation [1].
/PLUME/ (PLUMES, PRINT)
- \text{XPL}(J,IPL): \text{x coordinate at marching step } J \text{ of special line number IPL.}
- \text{YPL}(J,IPL): \text{y coordinate at marching step } J \text{ of special line number IPL.}

/IPLUME/ (PLUMES, PRINT)
- \text{KPL}: \text{number of special lines computed in PLUMES.}
- \text{ITYPL}(IPL): \text{index indicating the type of special line number IPL.}

/THICKY/ (OPACY, PRINT)
- \text{XTH}(J): \text{x coordinate on boundary characteristic line at marching step } J, \text{ from which radial opacity is integrated.}
- \text{TH}(J): \text{radial opacity computed by y-integration from the boundary point defined by XTH}(J) (up to current YN).}

/THICKX/ (OPACX, PLUMES, PRINT)
- \text{YXI}(JXI): \text{y coordinate of printed row number JXI (the index JXI counts just rows that have been printed). The row to be printed next upon calling PRINT is the row having YF near YXI}(JXI).
- \text{XI}(I,JXI): \text{lateral (x) molecular opacity [1] at point XF(I), for printed row JXI. It is obtained by numerically integrating the solution obtained from the JET computation (see OPACX).}
- \text{XIPM}(I,JXI): \text{same as XI}(I,JXI) except that the Prandtl-Meyer solution is used to estimate the flow at grid points XF(I).
- \text{XIGRP}(I,JXI): \text{same as above, except that the analytic approximation to a ring-symmetric CRW [1] is used to estimate the flow at grid points XF(I).}
- \text{XIAPP}(I,JXI): \text{same as XIGRP}(I,JXI) except that the numerical integration is replaced by an approximate closed-form expression [1].}
- \text{XIF}(I,JXI): \text{stores grid points XF(I) of printed row JXI.}

/GRP/ (PRINT, HMSET, MFUNC, HINTER, MATCH)
- \text{DMINV}: \text{increment of inverse Mach number for array MHINV(I).}
- \text{MHINV}(I): \text{inverse Mach number array (from 0 to I:MEXIT), from which the } H(M) \text{ function can be evaluated (HMSET).}
- \text{HMV}(I): \text{values of the } H(M) \text{ function evaluated by numerical integration. It is used to compute this function by interpolation. (HMSET, HINTER).}

3.3 Major Parameters

Parameters that define and control a particular run (such as the maximum $y$ for the marching computation, the number of grid points on a row and many more) are defined in INIDAT. (The code JET has no input file and no READ statements). The major control parameters are grouped in /PAR/ (floating point) and in /IPAR/ (integers); thermodynamic data are grouped in /STAG/.

We indicate in the listing the subroutines in which the labeled COMMON group or a particular parameter is defined (or sometimes referred to).

/PAR/ (INIDAT)

- **MEXIT** nozzle exit Mach number ($M_1$).
- **MFIN** Mach number of the final (boundary) CRW characteristic at the corner ($M_e$).
- **YMAX** maximum value of $y$ for the marching scheme. When $YF \geq YMAX$ the run is terminated.
- **DY0** initial marching step.
- **DY** current marching step.
- **DYNEXT** next marching step (YSTEP).
- **STAB** stability coefficient for marching step (STAB.LE.1). (See YSTEP).
- **DELTA** symmetry index. $DELTA = 0$ for plane symmetry; $DELTA = 1$ for ring-symmetry.
- **PSII** angle of Prandtl-Meyer fan characteristic at exit conditions (measured from $X$ axis).
- **PSIF** angle of final (boundary) Prandtl-Meyer fan characteristic.
- **SIGMA** collision cross-section ($\sigma$).
- **FRACG** the number of intervals initially allocated to the CRW fan is a FRACG fraction of the total number of intervals ($KFO-1$). (see BEGIN).
- **EPSIL** convergence parameter (small number). (INVMAR, SEMINV).
- **TETLIM** flow angle (from $X$ axis) of the limiting ($p = 0$) velocity vector of the flow at the lip-centered Prandtl-Meyer fan.
- **TETSYM** PAI-2*TETLIM for reflection transformation (see INTERP).

/IPAR/ (INIDAT)

- **JMAX** maximum number of marching steps. If $J \geq JMAX$ run is terminated.
- **KF0** initial (and maximum) number of grid points in a row.
- **KF** current number of grid points in the old row.
- **KN** current number of grid points in the new row.
ITER0 maximum number of iterations for the integration of the compatibility relations (see INVMAR and SEMINV; also used in RFUNC, PLUMES).

IM, IP search indices for interpolation subroutine INTERP. (see INVMAR, SEMINV).

J current row index (also index of a marching step).

KF2 defined as 2*KF; not used in present version.

IDEL, JDEL increments for printing grid point I and row J (see PRINT).

JYXI number of rows to be printed in a run.

JXI index of printing row, to be printed next (see PRINT).

Ilead index I at the first grid point on current new row, where the SIMA integration commences. Initially this point corresponds to the leading characteristic of the CRW. (see GRIDN, BEGIN).

ILEADF value of ILEAD for current old row.

KCLEAD index in the characteristic array for the characteristic line that corresponds to the new grid point I = ILEAD (see GRIDN). Initially KCLEAD = 1.

/STAG/ (INIDAT)

RHOO, NO stagnation density and number density.

P0, T0, A0 stagnation pressure, temperature and sound speed.

MDOT1 mass flow rate from ring-nozzle (only from the $x > 0$ half). (See PRINT).

/ICHARA/ (BEGIN)

KCHARP number of $C^+$ characteristic lines for which data is stored (either for SIMA computation or for subsequent plotting).

KCHARM number of $C^-$ characteristic lines for which data is stored (only for subsequent plotting).

KCHAR0 total number of characteristics for which data is stored, i.e., $KCHAR0 = KCHARP + KCHARM$. 

11
3.4 Description of JET subroutines

MAIN PROGRAM

The main program performs two functions. The first section (up to statement 1) is the initial set up; it is performed just once. The second section is the marching loop with the step index J. This program can be read as a flow chart of the overall computational procedure.

INIDAT is for setting up run data. In BEGIN the initial conditions for the marching computation are set up. A single marching step is performed by calling MARCH, and the loading of new row vectors into old row vectors is done by calling MOVE. The call to YSTEP is for the first computed marching step. All remaining calls are for informative tasks (see HMSET, BREAK, OPACY, PLUMES, PRINT). Run is terminated when either YF. GE. YMAX or when J. GE. JMAX.

NOTE ON EXEC: The only special feature in the EXEC is retaining the output unit 7 file for optional post-plotting. The printed output (unit 6) is the system's standard (default).

INIDAT

Initial data definition and preliminary data computations. The data is defined by statements rather than by reading an input file. The meaning of major parameters was described in Section 3.3 above. User is invited to modify the data definitions, particularly of run-control parameters such as YMAX, JYXI and YXI(JXI) (for printing JYXI selected rows).

BEGIN

Here all initial values (prior to beginning of marching schemes integration) are loaded into all major computational arrays (Section 3.1). Also, values of the key integer parameters KCHARP, KCHARM, KCHAR0, ILEAD, KCLEAD and KF are defined.

In the first loop (loop 1) we define an initial family of $C^+$ characteristic lines for the lip-centered CRW, by storing the Mach number of the Prandtl-Meyer fan characteristics in the array
MCHARI(KC). Note that the fan characteristics are generated at equal RP intervals, since the flow variables are RM and RP. However, a different division might also be acceptable.

The next step is the definition of initial values for all characteristic arrays, first the $C^+$ arrays (loop 2), then the $C^-$ arrays (loop 21). The $C^-$ characteristic lines are needed just for informative output (post-plotting), so the present version contains just one $C^-$ line. The user may modify that.

The remaining grid points (altogether KFO grid points are initially available) are uniformly distributed across the nozzle opening, and the row arrays are loaded with the corresponding nozzle-exit flow variables (loop 3).

PRINT

The main task of this subroutine is the printing of flow variables at grid points of selected rows. The printing of a row is selected when $YF$ is close to a predefined array $YXI(JXI)$. Following the printing, $JXI$ is updated by adding 1.

For comparison, additional flow variables are printed for each row. These are computed from the analytic approximation to a ring-symmetric CRW [1] by calling MATCH. Also, lateral molecular opacities of various kinds of approximation are computed by calling OPACX, and are printed for each grid point within the CRW.

Following the row printing (statement 120), arrays intended for post-processing (plotting of special lines) are printed and subsequently written on output unit 7. This is done once per run, just before run termination.

FIN

This subroutine is called when an error is encountered, in order to terminate the run. Note that the run is terminated by deliberately introducing an error of computing $\text{SQRT}(-1)$, which is done in order to trigger the printing of calling sequences by the operating system.
MARCH

This subroutine performs a single marching step by calling the proper computational subroutines at an appropriate sequence. It can be read as a flow chart of the entire computational scheme. First the segment of the new row suitable for SIMA computation is calculated by calling SEMINV. Then new grid points for that part of the new row for which inverse marching integration is to be performed, are generated by calling GRIDN. The results of the SEMINV computation, which were stored in characteristic arrays, are now loaded into row arrays by calling LOADC. Finally, the computation of the new row is completed by calling INVMAR which computes the flow at the remaining grid points by the inverse marching scheme.

INVMAR

This is one of the two central subroutines for computing the flow at new grid points (the other is SEMINV). Here the inverse marching scheme is used. The computational procedure follows the seven-step description given in Section 2.2 above. Note that the initial value of the search indices IM and IP is not redefined at each call to INTERP, since it is assumed that IM and IP do not change much at consecutive calls to INTERP, so that search efficiency is enhanced by not starting the search from an arbitrary point (such as either end of the row).

SEMINV

This is the subroutine performing the SIMA scheme for computing the flow at new grid points located along continuous characteristic lines of the lip-centered CRW (at prescribed $y$-marching steps). The essence of the computational procedure of this subroutine was given as a seven-step description in Section 2.2. The same remark about IM given in the preceding INVMAR description applies here as well.

The main loop (100) is over all characteristic lines, including some $C^-$ lines in addition to the $C^+$ lines. Thus, the array CSIGNF(KC) is used to get the appropriate expressions for either $C^+$ or $C^-$ characteristics. It is noted that while normally the characteristic segments through points 1 and 2 are $C^-$ and $C^+$ respectively, this is reversed when a $C^-$ rather than a $C^+$ line is computed via the SIMA scheme. In this case, which is characterized by having CSIGNF(KC).LT.0, the Riemann
invariants integrated along segments (1,4) and (2,4) are interchanged. This is done in the few statements just preceding and following statement 21.

An additional capability of this subroutine is to treat a change of a $C^+$ characteristic line into a $C^-$ line upon reflection from the symmetry plane ($x = 0$). This is done by first computing a new grid point having $X4.LT.0$, and then changing its sign after setting $C\text{SIGNN}(KC) = -1$ (statements just preceding statement 30). It is also possible to skip the computation of a particular characteristic by setting $C\text{SIGNN}(KC) = 0$. This feature is not exploited in the present version.

Finally, we note that not all characteristic lines computed here are part of the marching flow computation. Only those with indices $KC$ between $KC\text{LEAD}$ and $KC\text{CHARP}$ are. All other characteristic lines are computed just for informative purposes (post-plotting).

**RFUNC**

Here $M$, $MU$, $TETA$ are computed from the two Riemann invariants $RM$, $RP$. The computation of $M$ is performed by a Newton-Raphson iteration using Equations (2.1-2) and (2.1-3) given in Section 2.1 above.

**INTERP**

This subroutine starts by finding through a search procedure the grid interval $(I, I+1)$ that contains a given point $X$. Then the Riemann invariants are computed for this point by linear interpolation, and returned in $RM$, $RP$. Note that $X$ may be negative, which accounts for the relatively elaborate search logic in the determination of $I$, and for the reflection transformation (as in Eq. (2.3-1) above) preceding the last two statements of the subroutine.

**INTERX**

This interpolation routine performs an inverse task to that of INTERP, in that it finds the point $X0$ that corresponds to a given linearly interpolated value of the flow variable $VAR0$. It is used in PLU\text{MES} to compute the location of a breakdown surface point on a new row of $x$-centered and $y$-centered grid points.
This subroutine computes the new breakdown parameter array BN(I). The computation is based on the description given in Section 2.4 above.

Here the radial (Y) molecular opacity array TH(J) is computed. At each marching step J, a new boundary grid point XTH(J) is added, then the radial opacities at all preceding boundary points are updated by adding the contribution of the gas layer between the current old and new rows. Note that since grid points on adjacent rows are not located on equal-x columns, this procedure requires x-interpolation by calling INTERP.

This is a user-defined subroutine, where up to 10 special lines can be computed and subsequently retained on output unit 7 for post-processing (plotting). The type of the line ITYPL(IPL) and a parameter VPL(IPL) that defines the line, are computed through user-inserted statements in the section preceding statement 2000. Then an additional point on the current new row is computed for each line type. The available types are clearly stated in comments. Note that characteristic lines have already been computed in SEMINV using the SIMA scheme, regardless of whether they are part of the solution grid to the flow field, or are just computed for informative purpose. It is the user's choice which of these lines (if any) are to be saved in the /PLUME/ arrays for subsequent post-processing (plotting).

This subroutine computes the grid points in that segment of the new row for which the flow is computed by the inverse marching scheme (in INVMAR). Initially, this segment extends from $x = 0$ to the new row grid point which lies on the leading characteristic of the lip-centered CRW. However, since the leading characteristic is reflected from the symmetry plane ($x = 0$) at some point, this segment steadily shrinks in size as the marching proceeds. The remedy is to declare the next-to-the-
leading characteristic line (KC = 2) as the beginning of the segment for SIMA integration, by setting KCLEAD = 2. This process of increasing KCLEAD is repeated whenever it is deemed necessary. The criterion in the present version for the minimal KCLEAD is that the inverse-marching segment should be at least twice DX1 - the average CRW grid interval (loop 1, the two statements following DX1 = ...). Also, ILEAD is redefined for each row according to XLEAD/DX1 + 2 in order to achieve a row of relatively uniform grid intervals throughout. The result is that the number of grid points in a row is initially KF0, but eventually it decreases due to both increase of KCLEAD and decrease of ILEAD.

YSTEP

In this subroutine the next marching step DYNEXT is computed at the end of the current marching step. It is defined as the smallest step obtained by forward intersection of C− and C+ characteristics from adjacent grid points. Note that the actual value of DYNEXT is reduced by a "stability" factor STAB, and that DY is also limited by the growth-rate factor DDY and by Dymax (see MAIN PROGRAM).

MOVE

Here old row arrays (loop 1) and old characteristic arrays (loop 2) are loaded with values of flow variables from corresponding new arrays, in preparation for the next marching step. As a result of this organizational feature, informative computations (e.g. BREAK, OPACY) that require both new and old rows, have to be performed prior to calling MOVE.

OPACX

Here lateral (X) opacities that correspond to the number of expected collisions of a fast molecule invading the CRW in the −X direction, are computed. All opacities, except XIAPP(1), are computed by numerical integration. In loop 1 we compute the opacity contribution of the segment lying just outside the computational boundary characteristic (MFIN), assuming a Prandtl-Meyer flow. This additional opacity is denoted X10. If the flow is ring-symmetric, X10 is recalculated using the analytic approximation [1] to estimate the flow field at the fringes of the ring-symmetric CRW (see also the closed form expression for τ in [1]).
The computation of opacity arrays starts after statement 14. First, the opacity at each grid point is set to $X_{0}$. Thus, even though the numerical flow computation does not include the fluid outside the boundary characteristic line, the opacity integration includes an estimate of that "missing" part, i.e., of $X_{0}$. In typical case computations of a ring-symmetric CRW [1], we found that the maximum value of $X_{0}$ was about 0.16, which indicated that as far as interaction with invading ambient molecules is concerned, the approximation $M_{FIN} = 34$ was a reasonable substitute for $M_{FIN} = \infty$.

The next step is the computation by numerical integration of three approximations to the lateral opacity: $X_{I}(I,JX_{I})$, $X_{IPM}(I,JX_{I})$, $X_{IGRP}(I,JX_{I})$. (Note that when the flow is ring-symmetric, the approximation $X_{IPM}(I,JX_{I})$ obtained by assuming a Prandtl-Meyer flow is usually grossly exaggerated). The opacity $X_{IGRP}(I,JX_{I})$ is based on the analytic approximation to a ring-symmetric CRW [1], and is reasonably close to $X_{I}(I,JX_{I})$ which is obtained from the numerical solution to the flow field. Finally, a simplified closed-form integration of lateral opacity [1] is computed as $XI_{APP}(I,JX_{I})$ (loop 3). Thus, the quantitative difference between $XI(I,JX_{I})$ and $X_{IGRP}(I,JX_{I})$ is an indication to the degree of accuracy achieved by the analytic approximation to a ring-symmetric CRW [1], while the difference between $X_{IGRP}(I,JX_{I})$ and $XI_{APP}(I,JX_{I})$ indicates the level of error introduced by the closed-form integration of lateral opacity [1].

LOADC

Here flow variables of new grid points computed via the SIMA scheme (SEMINV) are loaded into new row arrays from corresponding characteristic arrays.

NUFUNC

This function computes the modified $v(M)$ value as given by Eq. (2.1-2). Note that presently $NU_{0} = 0$ (see INIDAT).

HMSET

This subroutine is called just once from the MAIN PROGRAM. Its task is to set up the arrays in /GRP/, so that the function $H(M)$ [1] can be evaluated by interpolation (in HINTER). There is also an informative printout of various derivatives (see Ch. 3 of [1]) generated in this subroutine.
MFUNC

This subroutine is called by HMSET in order to compute functions of Mach number that serve in the computation of $H(M)$. The output variable $F$ is the integrand for the integration leading to $H(M)$.

HINTER

This subroutine computes $H(M)$ by linear interpolation in inverse Mach number, using the /GRP/ arrays computed in HMSET.

MATCH

This subroutine is called from PRINT to compute the Mach number according to the analytic approximation of a ring-symmetric CRW [1], for point $(YF, XF(I))$. $M0B$ is the associate Mach number $M(0,\beta)$, which is preserved in the array MCHARI(KC) for all CRW characteristics that are used in the SIMA computation. Hence the Mach number $M(\alpha,\beta)$, denoted by MAB can be computed directly from the analytic approximation [1] to the area function at $(YF, XF(I))$ by calling AREAF. Since typically $M(0,\beta)$ is not known, we also compute the Mach number via the inverse-problem procedure [1], denoting the resulting Mach numbers by suffix I: $M0BI$ for $M(0,\beta)$ and $MABI$ for $M(\alpha,\beta)$. The inverse-problem iterative procedure [1] is performed in loop I, resulting in $M0BI$. From $M0BI$ the value of $MABI$ is computed through the area function approximation as for MAB above.

AREAF

This subroutine computes the Mach number $M$ that corresponds to the area function $F$ (Eq. (3.2-1) of [1]). The computation is done by Newton-Raphson iterations, and it has been found to converge when $M$.GT.1 (and when $M - 1$ is not much smaller than 1).
4. THE JET CODE LISTING

C$OPTIONS
C JET  A SEMI-INVERSE MARCHING CHARACTERISTICS METHOD FOR RING JETS.
C USING RIEMANN INVARIANTS RM=(NU+TETA), RP=(NU-TETA) AS FIELD
C VARIABLES.
IMPLICIT REAL*8(A-H,L-Z,*)
REAL*4 XPL,YPL
COMMON /PLUME/XPL(1002,0),YPL(1002)
COMMON /IPLUME/KPL,ITYPL(10)
COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
1 TETAF(101),BF(101),
2 XN(101),RMN(101),RPN(101),MN(101),MUN(101),
3 TETAN(101),BN(101),XTEMP(101)
COMMON/THICKY/XTH(1002),TH(1002)
REAL*4 YXI,XIXIPMXIGRP,XIAPP,XIF
COMMON THICKX/YXI(20),XI(101,20),XIPM(101,20),XIGRP(101,20)
1 XIAPP(101,20),XIF(101,20)
COMMON /GAMA/G,G,02,G3,G4,G5,6,G7,GS,G9,G100,G1,012,013,G14G15,
1 G16,G17,G18,G19,G20
COMMON /PAR/PAI,K0,DEG,XCYC,MEXITMFIN,YMAX,DYODYDYNEXT,
1 STAB,DELTA,PS1,PSF,ZETA1,SIGMA,FRACG,EPSILON,NU0,
2 TETSYM,TETSYM,TETSYM,DDY,DYMAX
COMMON /STAG/RHOO,NO,PO,TO,AO,MODI
COMMON /IPAR/JMAX,KFO,KF2,KN,IP,J,
1 KF2,DEB,JDEL,IBJ,JX,IXJ,IBF,ICF,ICF
COMMON /ROH/YF,YN,
DXF,DXN
COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
1 RMCARF(92),RPCARF(92),RMCA(92),RPMCA(92)
2 TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92)
3 CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARN(92),MCHARF(92)
4 MCHARI(92)
COMMON /ICHARA/KCHARPKCHARM,KCHARO
COMMON /GRP/DMINV,HMINV(101),HMV(101)
COMMON /IGRP/KHM

C PRINT 101
101 FORMAT('1')
J=1
IF(J.EQ.1) STOP
CALL INIDAT
PRINT 101
CALL HMSET
PRINT 101
CALL BEGIN
CALL MARCH
CALL OPACY
CALL PLUMES
CALL PRINT
J=2
CALL PLUMES
CALL MOVE
CALL OPACY
CALL PRINT
CALL YSTEP
J=J+1
C DY WAS DETERMINED BY THE PREVIOUS CALL TO GRIDN.
DY=DMIN1(DYNEXT,DY*DDY,DYMAX)
C INTEGRATE BY ONE Y-STEP
CALL MARCH
C BREAKDOWN PARAMETER (BF(I)).
CALL BREAK
C SPECIALLY DESIGNATED LINES (FOR PLOTTING).
CALL PLUMES
C STORE NEW LINE (N) IN OLD LINE (F).
CALL MOVE
C COMPUTE RADIAL MOLECULAR OPACITIES
CALL OPACY
C Y-STEP IS VARIABLE, SO JMAX IS USED AS END-OF-RUN CRITERION.
IF(YF.GE.YMAX) JMAX=J
C PRINT FIELD AT MOST RECENT Y.
CALL PRINT
C NEXT Y-STEP.
CALL YSTEP
IF(CY.LT.JMAX) GO TO 1
STOP
END

SUBROUTINE INITDAT
C SUBROUTINE NUMBER 1
IMPLICIT REAL*8(A-H,L-Z,$)
COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
1 TETAF(101),BF(101)
2 XN(101),RMN(101),RPN(101),MN(101),MUN(101)
3 TETAN(101),XTEMP(101)
COMMON/THICKY/XTH(1002),TH(1002)
REAL*8 YXI,XI,XIPM,XIGRP,XIAPP,XIF
COMMON /THICKX/YXI(20),XI(101,20),XIPM(101,20),XIGRP(101,20)
COMMON /GAMA/G1,2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15
1 G16,G17,G18,G19,G20
COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DYO,DY,DYNEXT,
1 STABDELTA,PS11,PSIFZETA1,SIGMA,FRACG,EPSIL,NUO,
1 TETSYM,TETLIM,DDY,DYMAX
COMMON /STAG/RHOO,NO,PO,TO,AO,MDOT
COMMON /IPAR/JMAX,KFO,ITERO,KF,KN,IM,J,
1 KF2,IDEJ,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
COMMON /ROW/YF,YN,DXF,DXN

C
PAI=4.DO*DATAN(1.DO)
PAI2=2.DO*DATAN(1.DO)
S.-DEG=180.DO/PAI
AR=8.3143D3
AV=6.022D26
AW=7.27D0
RHOO=0.0075D0
T0=2300.DO
G=1.54D0
D=2.5D-10
MEXIrZ=4.DO
MF1N=34.DO
XCO0.5D0
YC=2.5DO
C DELTA=0 CORRESPONDS TO PLANE SYMMETRY
C DELTA=1 CORRESPONDS TO CYLINDRICAL SYMMETRY
DELTA=1.DO
FRACG=0.6D0
EPSIL=1.DO
ITERO=20
KF0=101
JMAX=1001
STAB=0.5D0
DDY=1.05D0
DYMAX=0.5D0
YMAX=50.DO
DY0=YC=250.DO
IDEJ=1
JDEL=1
C POINTS FOR PRINTING FLOW FIELD AT YF=YXI(JXI)
JXI=1
JYXI(1)=YXI(1)+5.DO
YXI(2)=YXI(1)+2.DO
I=2
DO 1 I=I0,JXI
1 CONTINUE
IF(KF0.GT.101) CALL FIN(101)
IF(JMAX.GT.1001) CALL FIN(102)
IF(FRACG.GT.1.DO .OR. FRACG.LT.0.) CALL FIN(103)
IF(JYXI.GT.20) CALL FIN(104)
IF(DTMAX(1.DO-DTMAX).NE.0.) CALL FIN(105)
NO=RHOO*AV/AW
AO=DSQRT(GAR*TO/AW)
PO=AR*RHOO*TO/AW
SIGMA = PAI * D**2
LAMDA0 = (1.0D0 / DSQRT(2.0D0)) * SIGMA * N0
G1 = (G - 1.0D0) / 2.0D0
G2 = (G + 1.0D0) / (2.0D0 * (G - 1.0D0))
G3 = G / 2.0D0
G4 = (G + 1.0D0) / (G - 1.0D0)
G5 = DSQRT((G + 1.0D0) / (G - 1.0D0))
G6 = 1.0D0 / (G - 1.0D0)
G7 = 2.0D0 / (G - 1.0D0)
G8 = (1.5D0 * (G + 1.0D0) * 2.0D0 * (G - 1.0D0)) / ((G + 1.0D0) * (G - 1.0D0))

LAMDA0 = SIGMA / N0 * D**2
GluCG1 = D / 2.0D0
G2aC0 = 1.0D0 / (G + 1.0D0)
G5zO = 2.0D0
G4z = (G + 1.0D0) / ((G - 1.0D0) - 1.0D0)
G6z = D / (G - 1.0D0)
G7z = 1.0D0 / (G - 1.0D0)
G8z = 3.0D0 / (2.0D0 * G) / (G + 1.0D0)

ZETA1 = G5z * DATAN(DSQRT(CMEXIT**2 - 1.0D0) / 5.0)
AMIJ1 = DARSINC1(D0 / EXIT)
PSI1 = PAI * AMI1
ZETAF = G5z * DATAN(DSQRT(MFIN**2 - 1.0D0) / 5.0)
PSIF = PSI1 + ZETA1 - ZETAF
NUO = 0.
TETLIM = NUFUNC(EXIT) + PAI**2 * NUO
PSILIM = TETLIM
TETSYM = PAI**2 * TETLIM

RHO1 = RHOO / GOREM**G6
V1 = MEXIT * DSQRT(GOREM)

YYC = 2.0 * PAI * YC

IF (DELTA .EQ. 0.) YC = 1.0D0

MDOT1 = YYC * EXCERH01 * V1

PRINT 21, AR, AV, AW, G, RHOO, NO, P0, T0, AO, D
PRINT 22, XC, YC, MEXIT, RH01, P1, T1, V1, MDOT1, PSI1*DEG, PSIF*DEG, PSILIM*DEG
PRINT 23, DELTA, KFO, JMAX, ITER0, DY0, YMAX, STAB, DDY

RETURN
END

COMMON/VICS/XF(101),RMP(101),RPF(101),MF(101),MUF(101),
TETA(101),BF(101),
XN(101),RNM(101),RPN(101),MN(101),MUN(101),
TETAN(101),BN(101),XTEMP(101)
COMMON/THICKY/XTH(1002),TM(1002)
COMMON/GAMA/G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
G16,G17,G18,G19,G20
COMMON/PAR/PAI,PAII,DEG,XC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NUO,
TETSYM,TETLIM,DDY,DYMAX
COMMON/STAG/RHOO,NO,P0,T0,AD,MDOT1
DEFINE INITIAL CHARACTERISTIC PARAMETERS. USE INTERPOLATION OF RIEMANN INVARIANT ACROSS THE FAN.

KCHARP = 1DINT(FRACT(KFO-1)+1.D-6)+1
KCHARO = KCHARP+1
KCHARM = KCHARO-KCHARP
IF(KCHARP.LT.2) CALL FIN(200)
IF(KCHARO.GT.92) CALL FIN(210)
IF(KCHARM.LT.1) CALL FIN(205)
NUI = NUFUNC(MEXIT)
RM1 = NUO
TET1 = RM1-NUI
RP0 = NUI-TET1
NUMIN = NUFIN-(RM1-NFIN)
DRP = (RPFIN-RP0)/DFLOAT(KCHARP-1)
DO 1 KC=I,KCHARP
RPI = RP0+DRP*DFLOAT(KC-1)
CALL RFUNC(RM1,RP1,M1,MU1,TETA1)
MCHARI(KC) = M1
MUCARI(KC) = MU1
TCHARI(KC) = TETA1
RMCARI(KC) = RM1
RPCARI(KC) = RP1
1 CONTINUE

DEFINE GRID AND INITIAL CONDITIONS AT EXIT PLANE.

KFAN = KCHARP-1
ILEAD = KFO-KFAN
KCLEAD = 1
KF = KF0
KF2 = 2*KF
YF = YC
DO 3 I = 1, KF
KC = KCLEAD+I-ILEAD
IF(KC.GT.KCHARP) CALL FIN(241)
3 CONTINUE
IF(KC.GE.1) GO TO 31
XF(I)=DFLOAT(I-1)*XC/DFLOAT(IHEAD-1)
MF(I)=MEXIT
TETAF(I)=PI2
GO TO 32
CONTINUE
XF(I)=XC
MF(I)=MCHARF(KC)
TETAF(I)=TCHARF(KC)
32 CONTINUE
RMF(I)=NUFUNC(MF(I))+(TETAF(I)-TETLIM)
RPF(I)=NUFUNC(MF(I))-(TETAF(I)-TETLIM)
MU(I)=DARSINC1.DO/MF(I)
BF(I)=0.
3 CONTINUE
DY=DY0
DO 4 KC=1,KCHAR0
CSIGN(KC)=CSIGNF(KC)
4 CONTINUE
DO 5 I=1,Kn
BN(I)=0.
5 CONTINUE
RETURN
END
SUBROUTINE PRINT
IMPLICIT REAL*8(A-H,L-Z,*)
REAL*4 XPL,YPL
COMMON /PLUME/XPL(1002,10),YPL(1002)
COMMON /PLUME/KPL,ITYPL(10)
COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
1 TETAF(101),BF(101),
2 XN(101),RMN(101),RPN(101),MN(101),MUN(101),
3 TETAN(101),BNC(101),XTEMP(101)
COMMON /THICKY/XTH(1002),TH(1002)
REAL*4 XYI,XI,XIPM,XIGRP,XIAPP,XIF
COMMON /THICKY/YYYY(101,20),XI(101,20),XIPM(101,20),XIGRP(101,20),
1 XIAPP(101,20),XIF(101,20)
COMMON /GAMA/G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15
COMMON /PAR/PAI,PAIZ,DEG,XC,YC,MEXIT,MFIN,YMAX,DY,DYNEXT,
1 STAB,DELTA,PSIL,BSYM,TETLIM,DDY,DYMAX
COMMON /STAG/RHO0,NO,AO,MDOT1
COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
1 RMCARF(92),RPMCARN(92),RMPCARN(92),RPMCARN(92),
2 TCHARN(92),TCHARN(92),MUCARF(92),MUCARN(92),
3 CSIGNN(92),CSIGNF(92),MCHARF(92),MCHARN(92),
4 MCHARF(92),MCHARN(92)
COMMON /IPAR/JMAX,KFO,ITER0,KF,KN,IM,TB,J,
1 KF,K,DFD1,DFD2,JYX,JXI,ILEAD,ILEADF,KCLEAD
COMMON /ROW/YF,YN,DXF,DNX
C
SUM=0.
KF1=KF-1
DO 10 I=1,KF1
DX=ZXF(I+1)-ZXF(I)
GOREM=1.DO+G1*MF(I)**2
GOREM=1.DO+G1*MF(I+1)**2
RATEM=RHO0*MF(I+1)*G1*DSINTETA(I)+GOREM*(G6+0.5DO)
RATEP=RHO0*MF(I)*G1*DSINTETA(I+1)+GOREM*(G6+0.5DO)
SUM=SUM+DX*(RATEM+RATEP)/2.DO
10 CONTINUE
YYF=2.DO*CSIGNF(YF)
IF(DELTA.EQ.0.) YYF=1.DO
MDOTFR=YYF*SUM/MDOT1
PRINT 11,J,KCLEAD,KB,ILEAD,YF,DY,XF(KF),MF(KF),MDOTFR
1 FORMAT(1X,J,KCLEAD,KB,ILEAD,YF,DY,XF(KF),MF(KF),MDOTFR=,1)
C
C PRINT FLOW FIELD AT Y=YYF
C
C
```fortran
IF(J.EQ.JMAX) JXI=MINT(JXI,JYXI)
IF(J.EQ.1 .OR. J.EQ.JMAX) GO TO 121
IF(JXI.GT.JYXI) GO TO 120

CONTINUE
YXI(JXI)=YF
CALL OPACX
RETURN
C COMPUTE MACM NUMBER FOR CYLINDRICAL EXPANSION MCYL.
F=(YF/YC)*(G7*(1.D0+G1*MEXIT**2))**2/G2/MEXIT
CALL AREAF(F,MCYL)
PRINT 22,JXI,KCLEAD,ILEAD,KF,MCYL,YF
22 FORMAT(/1X, 'PRINTING NUMBER JXI,KCLEAD,ILEAD,KF=',4I4,
15X,'MCYL,YF:',2D14.5/)
PRINT 1
1 FORMAT(/1X, 'KC XF(I), TETAF(I), MF(I), MAB, MABI, MOBI,
15X, XI(I), XIGRP(I), XIAPP(I), XIPM(I)'/)
IDELz=IDEL1
IF(J.EQ.1 .OR. J.EQ.JMAX) IDEL1=1
DO 20 I=1,KF,IDEL1
KC=KCLEAD+(1-ILEAD)
IF(KC.LT.KCLEAD) KC=0
MAB=1.D10
MABI=1.D10
MPM=MFC(KC)
IF(KC.EQ.0) GO TO 23
MAB=MCHARCKC)
IF(J.EQ.1) GO TO 23
PSIPM=PAI2-DATANC (XF(I)-XC)/(YF-YC)
ZETA=PSIPM+ZETA1-PSIPM
MPN=DSQRT((G5*DTANCZETA/G5)**2+1.DO)
CALL MATCH(I,KF,MAB,MABI)
23 CONTINUE
PRINT 21,I,KC,XF(I),TETAF(I)*DEG,MF(I),MAB,MABI,MABI,
15X, XI(I), XIGRP(I), XIAPP(I), XIPM(I)'/)
21 FORMAT(/2X,'PLUME TYPES IPL,ITYPL(IPL)=',
15X,2(/1X,5(5X,2I.)))
20 CONTINUE
IF(J.EQ.1) GO TO 120
IF(J.EQ.JMAX) GO TO 120
PRINT 101
101 FORMAT('1')
102 FORMAT('RADIAl MOLEClAR THICKNESS J,XTH(J),TH(J)=''/)
103 FORMAT('PLUME TYPES IPL,ITYPL(IPL)=''/)
104 FORMAT('PLUME POINTS J,YPL(J),XPL(J,1),XPL(J,2),...=''/)
105 FORMAT('JDEL1=1
106 DO 203 JJ=1,JMAX, JDEL1
107 PRINT 204, JJ,YPL(JJ),XPL(JJ),IPL=1,KPL
204 FORMAT(/2X,2(/1X,9E13.6/2X,6E13.6/2X,6E13.6/2X,6E13.6))
205 CONTINUE
206 WRITE(7,205) JMAX,KPL
207 WRITE(7,205) (ITYPICIPI), IPL=1,KPL)
208 PRINT 103, (IPL,ITYPL(IPL),IPL=1,KPL)
209 CONTINUE
120 CONTINUE
IF(J.EQ.1) GO TO 120
JXI=JXI+1
121 CONTINUE
IF(J.LT.JMAX) GO TO 200
PRINT 101
101 FORMAT('1')
102 FORMAT('RADIAl MOLEClAR THICKNESS J,XTH(J),TH(J)=''/)
103 FORMAT('PLUME TYPES IPL,ITYPL(IPL)=''/)
104 FORMAT('PLUME POINTS J,YPL(J),XPL(J,1),XPL(J,2),...=''/)
105 FORMAT('JDEL1=1
106 DO 203 JJ=1,JMAX, JDEL1
107 PRINT 204, JJ,YPL(JJ),XPL(JJ),IPL=1,KPL
204 FORMAT(/2X,2(/1X,9E13.6/2X,6E13.6/2X,6E13.6/2X,6E13.6))
205 CONTINUE
206 WRITE(7,205) JMAX,KPL
207 WRITE(7,205) (ITYPICIPI), IPL=1,KPL)
208 PRINT 103, (IPL,ITYPL(IPL),IPL=1,KPL)
209 CONTINUE
120 CONTINUE
IF(J.EQ.1) GO TO 120
JXI=JXI+1
121 CONTINUE
IF(J.LT.JMAX) GO TO 200
PRINT 101
101 FORMAT('1')
102 FORMAT('RADIAl MOLEClAR THICKNESS J,XTH(J),TH(J)=''/)
103 FORMAT('PLUME TYPES IPL,ITYPL(IPL)=''/)
104 FORMAT('PLUME POINTS J,YPL(J),XPL(J,1),XPL(J,2),...=''/)
105 FORMAT('JDEL1=1
106 DO 203 JJ=1,JMAX, JDEL1
107 PRINT 204, JJ,YPL(JJ),XPL(JJ),IPL=1,KPL
204 FORMAT(/2X,2(/1X,9E13.6/2X,6E13.6/2X,6E13.6/2X,6E13.6))
205 CONTINUE
206 WRITE(7,205) JMAX,KPL
207 WRITE(7,205) (ITYPICIPI), IPL=1,KPL)
208 PRINT 103, (IPL,ITYPL(IPL),IPL=1,KPL)
209 CONTINUE
120 CONTINUE
IF(J.EQ.1) GO TO 120
JXI=JXI+1
121 CONTINUE
IF(J.LT.JMAX) GO TO 200
PRINT 101
101 FORMAT('1')
102 FORMAT('RADIAl MOLEClAR THICKNESS J,XTH(J),TH(J)=''/)
103 FORMAT('PLUME TYPES IPL,ITYPL(IPL)=''/)
104 FORMAT('PLUME POINTS J,YPL(J),XPL(J,1),XPL(J,2),...=''/)
105 FORMAT('JDEL1=1
106 DO 203 JJ=1,JMAX, JDEL1
107 PRINT 204, JJ,YPL(JJ),XPL(JJ),IPL=1,KPL
204 FORMAT(/2X,2(/1X,9E13.6/2X,6E13.6/2X,6E13.6/2X,6E13.6))
205 CONTINUE
206 WRITE(7,205) JMAX,KPL
207 WRITE(7,205) (ITYPICIPI), IPL=1,KPL)
208 PRINT 103, (IPL,ITYPL(IPL),IPL=1,KPL)
209 CONTINUE
120 CONTINUE
```
C WRITE LATERAL (X) OPACITIES
  JX1=JXI
  WRITE(7,205) JX10,KFO
  PRINT 226, JX10,KFO
226 FORMAT(//,1X,'LATERAL (X) OPACITIES JX10,KFO=,2I8)
  DO 220 JXI=1,JX10
     WRITE(7,221) JXI,YXI(JXI)
221 FORMAT(10,E13.6)  PRINT 227, JXI,YXI(JXI)
227 FORMAT(//,1X,'JXI,YXI(JXI)=',I,15.6/)
  DO 225 I=1,KFO
     WRITE(7,211) XIAP(I,JXI),XIAPC(I,JXI),XIPM(I,JXI),XIGRP(I,JXI),
                 XIAPP(I,JXI)
225 CONTINUE
  CONTINUE
200 CONTINUE  RETURN
END

SUBROUTINE FIN(FIN)
C SUBROUTINE NUMBER 4
C STOP WHEN ERROR IS DETECTED.
IMPLICIT REAL*8(A-H,L-Z,$)
PRINT 1,IFIN
1 FORMAT(/1X,'FIN CODE IFIN=',I6/)  CALL SEMINV
C INDUCE ERROR IN ORDER TO GENERATE TRACING OF CALLING SUBROUTINES.
X=-.DO
Y=X+DSQRT(X)
IF(IFIN.LE.0) GO TO 100
STOP
100 RETURN
END

SUBROUTINE MARCH
C SUBROUTINE NUMBER 5
IMPLICIT REAL*8(A-H,L-Z,$)
COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
               TETAFC(101),BF(101),
               XN(101),RMN(101),RPN(101),MN(101),MUN(101),
               TETAN(101),BN(101),XTEMP(101)
COMMON /THICKY/XTH(1002),TH(1002)
COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
               G16,G17,G18,G19,G20
COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY,DYNEXT,
               STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NUO,
               TETSYM,TETLIM,DDY,DYMAX
COMMON /ROW/RF,YF,YN,DXF,DXN
COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
                RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
                MCHARN(92),MCHARF(92),CSIGNM(92),CSIGNF(92),
                CSIGNI(92)
COMMON /ICHARA/KCHARP,KCHARM,KCHARO
C ADVANCE FLOW FIELD FROM YF TO YN
IM=KF
IP=KF
YN=RF+DY
KN=KFO
C SEMI-INVERSE INTEGRATION FOR FAN POINTS.
CALL SEMINV
C NEW GRID POINTS (JUST INVERSE MARCHING).
CALL GRIDN
C LOAD FLOW VARIABLES FROM SEMI-INVERSE INTEGRATION INTO VECTORS.
CALL LOADC
C CHARACTERISTIC SCHEME INTEGRATION FOR INNER POINTS (INVERSE MARCH).
CALL INVMAR
RETURN
END
SUBROUTINE INVMAR

IMPLICIT REAL*8(A-H,L-Z,$)

COMMON /VECS/XF(IO1),RMF(IO1),RPF(IO1),MF(IO1),MUF(IO1),TETAF(IO1),BF(IO1),XN(IO1),RMN(IO1),RPN(IO1),MN(IO1),MUN(IO1),TETAN(IO1),BN(IO1),XTEMP(IO1)
COMMON /THICKY/XTH(IO1),TH(IO1)
COMMON /GAMA/G,Gl,GZ,03,G4,G5,G6,07,GS,G9,GlO,011,013G14,GS
COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DYO,DY,DYNEXT,STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSILNUO,TETSYM,TETLIM,DDY,DYMAX
COMMON /STAG/RHOO,NO,PO,TO,AO,MDOT1
COMMON /IPAR/JMAX,KFO,ITERO,KF,KN,IM,J,

*COMMON /ROW/YF,YN,DXF,DXN

C INTEGRATION WITH INVERSE CHARACTERISTICS FOR NEW POINT(X4,Y4).
C OLD POINTS ARE (X1,Y1),(X2,Y2).
C X1 IS OBTAINED BY INVERSE C- FROM X4
C X2 IS OBTAINED BY INVERSE C+ FROM X4
C NOTE THAT X1 MAY BE NEGATIVE (E. G. WHEN X4=0).

KNI=ILEAD-1
IF(KNI.LE.0) CALL FIN(601)
DO 1000 I=1,KN1
        I4=I
        X4=XN(I)
        Y4=YN
        IF4=(IM+IP)/2
        CALL INTERP(O,IF4,KF,X4,XF,RM4,RMF,RP4,RPF)
        CALL RFUNC(RM4,RP4,M4,MU4,TETA4)
        M14=M4
        MU14=MU4
        TETA14=TETA4
        M24=M4
        MU24=MU4
        TETA24=TETA4
        Y1=YF
        Y2=YF
        Y14=(Y1+Y4)/2.DO
        Y24=(Y2+Y4)/2.DO
        X1=1.DO
        X2=1.DO
        RM4=1.DO
        RP4=1.DO
        ITER=0
        GOTO 2

C CORRECTOR

1 ITER=ITER+1
C AVERAGED PROPERTIES ON C-(14),C+(24) CHARACTERISTICS.
RM14=(RM1+RM4)/2.DO
RP14=(RP1+RP4)/2.DO
RM24=(RM2+RM4)/2.DO
RP24=(RP2+RP4)/2.DO
M14,MU14,TETA14, M24,MU24,TETA24 AVERAGED ON C-,C+ CHARACTERISTICS.
CALL RFUNC(RM14,RP14,M14,MU14,TETA14)
CALL RFUNC(RM24,RP24,M24,MU24,TETA24)
2 CONTINUE

C NEW X1,X2
X10=X1
X10=X1
X20=X2
X1=X4-DY/DTAN(TETA14-MU14)
X2=X4-DY/DTAN(TETA24+MU24)
IF(X2.LT.0.) CALL FIN(670)
D14=DSQRT((X1-X4)**2+DY**2)
D24=DSQRT((X2-X4)**2+DY**2)
C INTERPOLATE OLD DISTRIBUTION FOR RM1,RP1 RM2,RP2 AT X1,X2.
CALL INTERP(O,IM,KF,X1,XF,RM1,RP1,RM2,RP2)
CALL INTERP(O,IP,KF,X2,XF,RM2,RP2,RM1,RP1)
C NO NEED FOR RE-AVERAGING SINCE IT INTRODUCES ONLY HIGHER ORDER
C CHANGES INTO THE ITERATION SCHEME.
C INTEGRATE THE CHARACTERISTIC EQUATIONS FOR RM4,RP4 AT X4,Y4.
RM0=RM4
RP0=RP4
RM1=RM1+DELTA*DSIN(TETA14)*D14/(M14*Y14)
RP1=RP2+DELTA*DSIN(TETA24)*D24/(M24*Y24)
C CONVERGENCE TEST
EPS=(DABS(X1-X1O)+DABS(X-X20))/DY+DABS(RM4-RM4O)+DABS(RP4-RP40)
IFCITER.GT.ITER0 GO TO 10
IF(EPS.GT.EPSIL) GO TO 1
RMN(I)=RM4
RPN(I)=RP4
CALL RFUNC(RM4,RP4,MN(I),MUN(I),TETAN(I))
1000 CONTINUE
RETURN
10 CONTINUE
PRINT 11,I4,KN,IF4,IM,IP,KF,ITER,ITERO,EPS,EPSIL,X1,X2,X4,M14,M24
11 FORMAT(IX,'SUBR. INVMAR. ',1X,IF4,IM,IP,KF,ITER,ITERO=',815/
     EPS,EPSIL,XZ,X2,X4,M14,M24=',7DI4.6/)
CALL FIN(611)
RETURN
END
SUBROUTINE SEMINV
IMPLICIT REAL*8(A-H,L-Z,$)
COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
1 TETAF(101),BF(101),
2 XN(101),RMN(101),RPN(101),MN(101),MUN(101),
3 TETAN(101),BN(101),XTEMP(101)
COMMON/THICKY/XTH(1002),TH(1002)
COMMON /GAMA/G,G1,G2,G3,04,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
1 G16,G17,G18,G19,G20
COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DYO,DY,DYNEXT,
1 STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NUO,
2 TETSY,TETLIM,DDY,DYMAX
COMMON /ROH/YF,YN,DXF,DXN
COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
1 RMCAF(92),RPCAF(92),RMCAF(92),RPCAF(92),
2 TCHAF(92),TCHAF(92),MUCAF(92),MUCAF(92),
3 CSIGNN(92),CSIGNF(92),MUCARN(92),MCHARF(92),
4 MCHARF(92)
COMMON /ICHARA/KCHARP,KCHARM,KCHARO
C COMPUTE NEW POINT (X4,Y4), BY PASSING A C+ CHARACTERISTIC
C THROUGH OLD POINT (X2,Y2). BOTH POINTS ARE ON CHARACTERISTIC LINE
C NUMBER KC.
IM=1
DO 100 KC=1,KCHARO
IF(CSIGNN(KC).EQ.O.) GO TO 100
Y1=YS
Y2=YS
Y4=YN
Y1=(Y1+Y4)/2.DO
Y2=(Y2+Y4)/2.DO
X2=XCHARF(KC)
RM2=RMCAF(KC)
RP2=RPCAF(KC)
M2=MCHARF(KC)
MU2=MUCAF(KC)
TETA2=TCHARF(KC)
M14=M2
MU14=MU2
TETA14=TETA2
M24=M2
MU24=MU2
100 CONTINUE
RETURN
END
FILE: JETPR FORTRAN A1

TETA24=TETA2
X4=1. D10
X1=1. D10
RM4=. D10
RP4=1. D10
ITER=0
GO TO 2

C CORRECTOR
1 ITER=ITER+1
C AVERAGED PROPERTIES ON C-(14),C+(24) CHARACTERISTICS.
RM14=(RM1+RM4)/2.DO
RP14=(RP1+RP4)/2.DO
RM24=(RM2+RM4)/2.DO
RP24=(RP2+RP4)/2.DO
C M14,MU14,TETA14, M24,MU24,TETA24 AVERAGED ON C-,C+
CALL RFUNC(RM14,RP14,M14,MU14,TETA14)
CALL RFUNC(RM24,RP24,M24,MU24,TETA24)
2 CONTINUE
C NEW X4,X1
X40=X4
X10=X1
X4=X2+DY/DTAN(TETA24+CSIGNF(KC)*MU24)
X1=X4-DY/DTAN(TETA14-CSIGNF(KC)*MU14)
D14=DSQRT((X1-X4)**2+DYW**2)
D24=DSQRT((X2-X4)*2+DY*2)
C INTERPOLATE OLD DISTRIBUTION FOR RM1,RP1, AT X1.
CALL INTERP(O,IM,KF,X1,XFRM1,RMFRP1,RPF)
IF(J.GT.1) GO TO 22
IF(CSIGNF(KC).LT.O.) GO TO 22
RP1=RP2
22 CONTINUE
C NO NEED FOR RE-AVERAGING SINCE IT INTRODUCES ONLY HIGHER ORDER
C CHANGES INTO THE ITERATION SCHEME.
C INTEGRATE THE CHARACTERISTIC EQUATIONS FOR RM4,RP4 AT X4,Y4.
RM40=RM4
RP40=RP4
IF(CSIGNF(KC).LT.O.) GO TO 21
RM4=RM1+DELTA*DSIN(TETA14)*D14/(M14*Y14)
RP4=RP2+DELTA*DSIN(TETA24)*D24/(M24*Y24)
GO TO 20
21 CONTINUE
RM4=RM2+DELTA*DSIN(TETA24)*D24/(M24*Y24)
RP4=RP1+DELTA*DSIN(TETA14)*D14/(M14*Y14)
20 CONTINUE
C CONVERGENCE TEST
EPS=(DABS(X4-X4O)+DABS(X1-X1O))/DY+DABS(RM4-RM4O)+DABS(RP4-RP4O)
IF(ITER.GT.ITERO) GO TO 10
IF(EPS.GT.EPSIL) GO TO 1
RMSAVE=RM4
RM4=RP4+TETSYM
RP4=RM4-TETSYM
CSIGNF(KC)=-1.DO
30 CONTINUE
RMCHAR(KC)=RM4
RPMCHAR(KC)=RP4
CALL RFUNC(RM4,RP4,M4,MU4,TETA4)
TCHAR(KC)=TETA4
XMCHAR(KC)=DABS(X4)
YMCHAR(KC)=Y4
MUMCHAR(KC)=MU4
MCHAR(KC)=M4
100 CONTINUE
RETURN
10 CONTINUE
PRINT 11,KC,KCHARO,IM,KF,ITER,ITERO,EPS,EPSIL,X1,X2,X4,MI4,M24
1 FORMAT(1X,'SUBR. SEMINV. KC,KCHARO,IM,KF,ITER,ITERO=',7D14.6/)
CALL FIN(711)
SUBROUTINE RFUNC(RM, RP, M, MU, TETA)
C
IMPLICIT REAL*8(A-H,L-Z,$)
COMMON /VECS/XF(101),RMM(101),RPF(101),MF(101),MUF(101),
1 TETA(101),BF(101),
2 XN(101),RMM(101),RPN(101),MN(101),MUN(101),
3 TETAN(101),BN(101),XTEMP(101)
COMMON/THICKY/XTH(1002),TH(1002)
COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
1 G16,G17,G18,G19,G20
COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY,DYNEXT,
1 STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NUO,
2 TETSYM,TETLIM,DDY,DYMAX
COMMON/RHOO,NO,PO,TO,A0,MDOT1
COMMON/TH/TH(1002),TH(1002)
COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY,DYNEXT,
1 STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NUO,
2 TETSYM,TETLIM,DDY,DYMAX
COMMON/STAG/RHOO,NO,PO,TO,A0,MDOT1
COMMON/JMAXDKFO,ITERO,KFKN,IM,IP,J,
1 KF2,IDEL,JDEL,JYXI,JXI,I LEAD,ILEADF,KCLEAD
COMMON/YF,YM,YF,YM
C
C COMPUTE M,MU,TETA AT A POINT, AS FUNCTION OF RIEMANN INVAR. RM,RP.
TETA=(RM-RP)/2.DO+TETLIM
NU=(RM+RP)/2.DO
C NU=NU0-(G5*ARCTAN(G5*Q)-ARCTAN(Q)), WHERE Q=(M**2-1)**(-1/2)
C FIND Q(NU), AND HENCE M(NU), THROUGH NEWTON-RAPHSON ITERATIONS.
Q=-(NU-NU0)/(G4-1.DO)
IF(Q.LE.0.) CALL FIN(801)
ITER=0
1 ITER=ITER+1
QF=Q
DNUDT=-(G4-1.DO)/((1.DO+G4**2)*(1.DO+Q**2))
DNU=NU-(NU0-(G5*DATAN(G531Q)-DATAN(Q)))
Q=Q+DNU/DNUDT
IF(Q.LE.0.) CALL FIN(811)
EPS=DABS(Q-QF)/Q
IF(ITER.GT.ITERO) GO TO 10
IF(EPS.GT.EPSIL*1.D-3) GO TO 1
M=DSQRT(1.DO+Q/Ex2)
MU=DARSIN(1.DO/M)
RETURN
10 CONTINUE
CALL FIN(810)
RETURN
END
SUBROUTINE INTERP(JNEWP I,KGRID,X,XVEC,RM,RMVEC,RP,RPVEC)
C
IMPLICIT REAL*8(A-H,L-Z,$)
COMMON /VECS/XF(101),RMM(101),RPF(101),MF(101),MUF(101),
1 TETA(101),BF(101),
2 XN(101),RMM(101),RPN(101),MN(101),MUN(101),
3 TETAN(101),BN(101),XTEMP(101)
COMMON/THICKY/XTH(1002),TH(1002)
COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
1 G16,G17,G18,G19,G20
COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY,DYNEXT,
1 STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NUO,
2 TETSYM,TETLIM,DDY,DYMAX
COMMON/RHOO,NO,PO,TO,A0,MDOT1
COMMON/JMAX/DKFO,ITERO,KFKN,IM,IP,J,
1 KF2,IDEL,JDEL,JYXI,JXI,I LEAD,ILEADF,KCLEAD
COMMON/YF,YM,YF,YM
C
C FIND I SUCH THAT XVEC(I).LE.X.AND.XVEC(I+1).GE.X
C FIND RM,RP BY LINEAR INTERPOLATION.
C NOTE THAT X MAY BE NEGATIVE.
IF(DABS(X).LE.XVEC(KGRID)) GO TO 901
PRINT 900,X,KGRID,XVEC(KGRID)
901 CONTINUE
CALL FIN(900)
KGRID=KGRID+1
RETURN
END
FILE: JETPR FORTRAN A1

```fortran
ID=MIXO(I,KGRID-2)
ICOUNT=0
I=10
SIGNI=1.DO
IF(I.GE.1) GO TO 10
SIGNI=-1.DO
I=I+2
10 CONTINUE
IF(I.GT.KGRID) CALL FIN(901)
XX1=SIGN1*XVEC(I)
I=1
IF(XX1.LE.X) GO TO 11
I=I-1
ICOUNT=ICOUNT+1
IF(ICOUNT.GT.KGRID) CALL FIN(911)
GO TO 11
11 CONTINUE
I=I+1
SIGN2=1.DO
IF(I.GE.1) GO TO 12
SIGN2=-1.DO
I=I+2
12 CONTINUE
IF(I.GT.KGRID) CALL FIN(912)
XX2=SIGN2*XVEC(I)
I=I+1
IF(XX2.GE.X) GO TO 13
I=I+1
ICOUNT=ICOUNT+1
IF(ICOUNT.GT.KGRID) CALL FIN(913)
GO TO 13
13 CONTINUE
F1=(XX2-X)/(XX2-XX1)
F2=1.DO-F1
IF(F1.LT.0.) CALL FIN(991)
IF(F2.LT.0.) CALL FIN(992)
RM1=RMF(I1)
RP1=RPF(I1)
RM2=RMF(I2)
RP2=RPF(I2)
IF(SIGN1.LT.0.) RM1=RPF(I1)+TETSYM
IF(SIGN1.LT.0.) RP1=RMF(I1)-TETSYM
IF(SIGN2.LT.0.) RM2=RPF(I2)+TETSYM
IF(SIGN2.LT.0.) RP2=RMF(I2)-TETSYM
RM=F3*RM1+F2*ERM2
RP=Fl*RP1+F2*RP2
RETURN
END
```

SUBROUTINE INTERX(N1,N2,VARO,VAR,KGRID,XO,XVEC)

C SUBROUTINE NUMBER 10

IMPLICIT REAL*8(A-H,L-Z,$)

COMMON /VECS/XF(10),RMF(10),RF(10),MF(10),MUF(10),
1 TETA(101),BFC(101),
2 XNC(101),RMNC(101),RPN(101),MN(101),MUND(101),
3 TETAN(101),BNC(101),XTEMP(101)

COMMON /THICKY/XTHC(1002),XTHC(1002)

COMMON /GAMA/G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15
1 G16,G17,G18,G19,G20

COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DYO,DY,DYNEXT,
1 STAB,DELTA,PSII,PSIF,ZETA1,SIGMA,FRACG,EPSSIL,NUO,
2 TETSYM,TETLIM,DY,DYMAX

COMMON /STAG/RHOO,NPODP0,T0,AD,MOT1

COMMON /IPAR/JMAX,JFIF,JFEND,KF,IF0,KN,IM,IP,J,
1 KF2,IDEJ1,IDEJ2,JI,JXI,JEAD,JEADF,JEALD,JEAD,JEAD,
2 COMMON /ROW/YF,YD,DX,DXF,DYN

DIMENSION VARO(1),XVEC(1)

C FIND XO AND I1 SUCH THAT XVEC(I1)<XO<XVEC(I1+1), AND XO CORRESPONDS
C TO THE LOCATION AT WHICH VARO IS A LINEAR INTERPOLATION OF VAR(I).

C TO THE LOCATION AT WHICH VARO IS A LINEAR INTERPOLATION OF VAR(I).

XO=1.D3
IFIRST=I1
IF(I1.GT.0) GO TO 10
IFIRST=KGRID-1ABS(I1)+2
```
10 CONTINUE
DO 1 I = IFIRST, KGRID
I = I + 1
IF(I.GT.0) GO TO 11
I = KGRID - I + 2
11 CONTINUE
IF(I.LE.0) CALL FIN(1001)
IF(I.GT.KGRID) CALL FIN(1002)
IF(I.EQ.1) GO TO 1
IF((VAR(I) - VAR(0)) * (VAR(I - 1) - VAR(0)).GT.0.) GO TO 1
IF(VAR(I).EQ.VAR(I - 1)) GO TO 1
IF((VAR(I) - VAR(0)) / (VAR(I) - VAR(I - 1))) F1 = 1.DO - F1
F2 = 1.DO - F2
IF(F1.LT.0.) CALL FIN(1011)
IF(F2.LT.0.) CALL FIN(1012)
X0 = F1 * XVEC(I - 1) + F2 * XVEC(I)
1 = I - 1
GO TO 2
1 CONTINUE
2 CONTINUE
RETURN
END

C SUBROUTINE BREAK
C SUBROUTINE NUMBER 11
IMPLICIT REAL*8(A-H,L-Z,$)
REAL MB,MX,MY
COMMON /VECS/XF(101),RMF(101),MUF(101),
1 TETAF(101),BF(101),
2 XN(101),RNM(101),RPN(101),MN(101),MUN(101),
3 TETAN(101),BN(101),XTEMP(101)
COMMON /THICKY/XTH(1002),TH(1002)
COMMON /GAMA/G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15
1 G16,G17,G18,G19,G20
COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DYO,DY,DYNEXT,
1 STAB,DELTAPSIFZETASIGMA,FRACG,EPSIL,NUO,
2 TETSYM,TETLIM,DDY,DYMAX
COMMON /IPAR/JMAX,KFO,ITERO,KF,KN,IM,IP,J,
1 KFD,KFD2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
COMMON /ROW/YF,YN, DXF, DXN
C COMPUTE THE BREAKDOWN PARAMETER AT (I-1/2,K-1/2). STORE IN BN(I).
YB = 0.5D0* (YF + YN)
DYY = DY
IM = 2
DO 1 I = 2, KN
X1 = XNCI - 1
X2 = XNCI
DX = X2 - X1
IF(X2.GT.XF(KF)) GO TO 2
CALL INTERP(0,IM,KF,X1,XF,RM1,RM2,RP1,RP2,RP)
CALL INTERP(0,IM,KF,X2,XF,RM2,RM2,RP2,RP)
CALL RFUNC(RM1,RP1,MI,MI,TETAl)
CALL RFUNC(RM2,RP2,MI,MI,TETA2)
MY = 0.5D0* ((MN(I) - MN(I - 1)) + (M2 - M1)) / DYY
MB = 0.25D0* (MN(I) + MN(I - 1))
MB = 0.25D0* (MN(I) + MN(I - 1) + M2)
TETAB = 0.25D0* TETAN(I) + TETAN(I) + TETA1 + TETA2
GOREM = MBX2* (1.DO + G1*MBX2)* (G6 - 1.DO)
GRAD = MXDCOS(TETAB) + MYDSINC(TETAB)
B = 0.2*GOREMGRA
GO TO 3
2 B = 1.DO2
3 BN(I) = B
1 CONTINUE
BN(I) = BN(2)
RETURN
END

C SUBROUTINE OPA CY
C SUBROUTINE NUMBER 12
IMPLICIT REAL*8(A-H,L-Z,$)
COMMON /VECS/XF(101),RMF(101),MUF(101),
1 TETAF(101),BF(101),
2 XN(101),RNM(101),RPN(101),MN(101),MUN(101),
3 TETAN(101),BN(101),XTEMP(101)
COMMON /THICKY/XTH(1002),TH(1002)
COMMON /GAMA/G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15
1 G16,G17,G18,G19,G20
COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DYO,DY,DYNEXT,
1 STAB,DELTAPSIFZETASIGMA,FRACG,EPSIL,NUO,
2 TETSYM,TETLIM,DDY,DYMAX
COMMON /IPAR/JMAX,KFO,ITERO,KF,KN,IM,IP,J,
1 KFD,KFD2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
COMMON /ROW/YF,YN, DXF, DXN
FILE: JETPR FORTRAN A1

1 TETAF(101), BF(101), JET0937
2 XN(101), RMN(101), RPN(101), MN(101), MUN(101), JET0938
3 TETAN(101), BN(101), XTEMP(101), JET0939

COMMON /THICKY/XTH(1002), TH(1002), JET0940
COMMON /GAMA/G, G1, G2, G3, G4, G5, G6, G7, G8, G9, G10, G12, G13, G14, G15, JET0941
1 G16, G17, G18, G19, G20, JET0942
COMMON /PAR/PAI, PAI2, DEG, XC, YC, MEXIT, MFIN, YMAX, DY0, DY, DYNEXT, JET0943
1 STAB, DELTA, PSI1, PSI2, ZETA1, SIGMA, FRACG, EPSIL, NU0, JET0944
2 TETSYM, TETLIM, DDY, Dymax, JET0945
COMMON /STAGE/ROHO, NO, PO, TO, AO, MDOT1, JET0946
COMMON /IPAR/ JMAX, KFO, ITERO, KF, KN, IM, IP, J, JET0947
1 KF2, IDEL, JDEL, JX1, JXI, ILEAD, IDEADF, KCLEAD, JET0948
COMMON /ROW/YF, YN, DXF, DXN, JET0949

C COMPUTE THE MOLECULAR THICKNESS AT END POINTS OF EACH ROW.
IM=2
XTH(J)=XF(KF)
TH(J)=0.
DTH0=NO*SIGMA*DY
IF(J.EQ.1) GO TO 11
J1=J-1
DO 1 JJ=1,J1
XX1=XTH(JJ)
CALL INTERP(OIMKF, XX1, XF, RM1, RMF, RP1, RPF)
CALL RFUNC(RM1, RP1, M1, MU1, TETA1)
GOREM=1.DO+Gl*M1*W3E2
DTH=DTH0/GOREM**G6
TH(JJ)=TH(JJ)+DTH
1 CONTINUE
11 CONTINUE
RETURN
PLUMES
JET0950
SUBROUTINE NUMBER 13 JET0951
IMPLICIT REAL*8(A-H,L-Z,$) JET0952
REAL*4 XPL,YPL
COMMON /PLUME/XPL(1002,10), YPL(1002)
COMMON /ITPLUME/KPL, ITYPL(10)
DIMENSION VPL(92)
COMMON /VECS/XF(101), RMF(101), RPF(101), MF(101), MUF(101), JET0953
1 TETAF(101), BF(101), XN(101), RMN(101), RPN(101), MN(101), MUN(101), JET0954
3 TETAN(101), BN(101), XTEMP(101), JET0955
COMMON /THICKY/XTH(1002), TH(1002), JET0956
REAL*4 YXI,XI,XIPM,XIGRP,XIAPP,XIF
COMMON /THICKX/YXI(20), XI(101,20), XIPM(101,20), XIGRP(101,20), JET0957
1 XIAPP(101,20), XIF(101,20)
COMMON /GAMA/G, G1, G2, G3, G4, G5, G6, G7, G8, G9, G10, G11, G12, G13, G14, G15, JET0958
1 G16, G17, G18, G19, G20, JET0959
COMMON /PAR/PAI, PAI2, DEG, XC, YC, MEXIT, MFIN, YMAX, DY0, DY, DYNEXT, JET0960
1 STAB, DELTA, PSI1, PSI2, ZETA1, SIGMA, FRACG, EPSIL, NU0, JET0961
2 TETSYM, TETLIM, DDY, Dymax, JET0962
COMMON /STAGE/ROHO, NO, PO, TO, AO, MDOT1, JET0963
COMMON /IPAR/ JMAX, KFO, ITERO, KF, KN, IM, IP, J, JET0964
1 KF2, IDEL, JDEL, JX1, JXI, ILEAD, IDEADF, KCLEAD, JET0965
RETURN
END

C COMPUTE SPECIAL POINTS AT Y=YN, AND STORE THEM AS
C (XPL(J,IPL), YPL(J)=YN).
C J IS THE MARCHING INDEX OF YN.
C IPL=1, 2, ..., KPL IS THE "PLUME" INDEX. PRESENTLY KPL.LE.5
C VPL(IPL) IS A VALUE DEFINING THE "PLUME" CURVE.
C ITYPL(IPL) IS THE TYPE OF CURVE. IT DEFINES CURVES AS FOLLOWS:
C ITYPL(IPL)=0 DO NOTHING
C ITYPL(IPL)=1 REAL PLUME. IT IS THE BREAKDOWN SURFACE, DEFINED
C BY A CONSTANT VALUE OF THE BREAKDOWN PARAMETER B.
C SET VPL(IPL)=B.
C DEFINE ITYPL(IPL) AND VPL(IPL)
    KPL=10
    IF(KPL.GT.10) CALL FIN(1301)
    DO 2000 IPL=I,KPL
2001 ITYPL(IPL)=4
    VPL(IPL)=1
    GO TO 2000
2002 ITYPL(IPL)=4
    VPL(IPL)=KCHARP
    GO TO 2000
2003 ITYPL(IPL)=4
    VPL(IPL)=19
    GO TO 2000
2004 ITYPL(IPL)=4
    VPL(IPL)=31
    GO TO 2000
2005 ITYPL(IPL)=4
    VPL(IPL)=47
    GO TO 2000
2006 ITYPL(IPL)=4
    VPL(IPL)=55
    GO TO 2000
2007 ITYPL(IPL)=1
    VPL(IPL)=0.02DO
    GO TO 2000
2008 ITYPL(IPL)=1
    VPL(IPL)=0.03DO
    GO TO 2000
2009 ITYPL(IPL)=1
    VPL(IPL)=0.05DO
    GO TO 2000
2010 ITYPL(IPL)=1
    VPL(IPL)=0.08DO
    GO TO 2000
2000 CONTINUE
C COMPUTE "PLUME" POINTS AT Y=YN
    DO 1000 IPL=1,KPL
    ITYP=ITYPL(IPL)
    IF(ITYP.EQ.0) GO TO 1000
    GO TO (1,2,3,4,5), ITYP
1 CONTINUE
C BREAKDOWN SURFACE PLUME.
C NOTE THAT DUE TO DIFFERENCE-CENTERING OF GRADIENTS, THE ACCURATE
C Y-COORDINATE IS 0.5*(YF+YN), RATHER THAN YN. IT CAN BE ADJUSTED
C IN THE PLOTTING CODE.
BO=VPL(IPL)
    XTEMP(1)=XN(1)
    DO 11 I=2,KN
        XTEMP(I)=0.5DO(XN(I)+XN(I-1))
11 CONTINUE
2 CONTINUE
C FIND BY INTERPOLATION THE X-COORDINATE WHERE M=MPL.
    IF(J.GT.1) GO TO 200
    XPL(J,IPL)=XC
    GO TO 2001
200 CONTINUE
    MPL=VPL(IPL)
    I=-KN
    CALL INTERX(1,I,BO,BN,KN,XBO,XTEMP)
CALL INTERX(1,1,MP,L,MN,KN,KNM,KXM,KXN)
XPL(J,JPL)=XMO
GO TO 1001
3 CONTINUE
C STREAMLINE INTERPOLATION.
IF(J.GT.1) GO TO 500
XPL(J,JPL)=VPL(JPL)
GO TO 1001
300 CONTINUE
XSF=XPL(J-1,JPL)
ISF=2
ISN=2
CALL INTERP0(ISF,KF,XSF,XF,RMSF,RPSF,RPF)
CALL RFUNC(RMSF,RPSF,MF,MUF,TETASF)
XSN=XSF+DY*DTAN(PAI2-TETASF)
ITER=1
301 ITER=ITER+1
CALL INTERP(ISN,KN,XSN,XN,RMSN,RM,RPN,RP)
CALL RFUNC(RMSN,RPN,MN,MUN,TETASN)
TETAASV=0.5D0*(TETASF+TETASN)
XSN=XSF+DY*DTAN(PAI2-TETASF)
IF(ITER.LT.ITER0+2) GO TO 301
XPL(J,JPL)=XSN
GO TO 1001
4 CONTINUE
C CHARACTERISTIC LINE.
KC=IDINT(VPL(IPL)+1.D-5)
IF(J.GT.1) GO TO 41
XPL(J,JPL)=XCHARF(KC)
GO TO 1001
41 CONTINUE
XPL(J,JPL)=XCHARF(KC)
IF(CSIGNN(KC).EQ.0.) XPL(J,JPL)=1.0E33
GO TO 1001
5 CONTINUE
C CONSTANT LATERAL (X) OPACITY
CALL OPAC
XIC=VPL(IPL)
DO 51 II=2,KF
II=KF-II+1
I2=II+1
XI=VIX(J,JXI)
XIC=XIC+I2*(I2-I)*(XI-I)
IF((XIC-XI)3(XIC-XI2).GT.0.) GO TO 51
F2=(XI2-XIC)/(XI2-XI1)
F=I.DO-F2
IF(F1.LT.O.) CALL FIN(1351) JET1125
IF(FZ.LT.O.) CALL FIN(1352) JET1125
XIFC=F2*XF(I1)+FIXF(I2)
GO TO 52
51 CONTINUE
XIFC=1.0D30
52 CONTINUE
XPL(J,JPL)=XIFC
GO TO 1001
1001 CONTINUE
IF(J.GT.1) GO TO 1002
YPL(J)=YC
GO TO 1000
1002 CONTINUE
YPL(J)=YN
1000 CONTINUE
RETURN
C SUBROUTINE GRIDN
SUBROUTINE GRIDN
IMPLICIT REAL*(A-H,L-Z,*)
REAL*4 XPL,YPL
COMMON /PLUME/XPL(1002,10),YPL(1002)
COMMON /IPLUME/XPL,ITYPL(10)
COMMON /VECS/XF(101),RMF(101),RF(101),MF(101),MUF(101),
1 TETAF(101),BF(101),
SUBROUTINE YSTEP

C DIVIDE LINE Y=YN INTO KN-1 INTERVALS.

C THE X-GRID IS NON-UNIFORMLY DEFINED AS FOLLOWS:

C (1) (XCHARN(I),YCHARN(I)), (XCHARN(I),YCHARN(I)), I=1,2,...,KCHARP,

C DENOTE NEW AND OLD (FORMER) CHARACTERISTIC (C+) POINTS. LET I=1

C AND I=KCHARP CORRESPOND TO THE LEADING AND BOUNDARY

C CHARACTERISTICS (C+).

C (2) THE GRID CONSISTS OF TWO SEGMENTS. THE SO-CALLED FLAT SEGMENT

C IS BETWEEN X=0 AND X=XLEAD=XCHARN(KCLEAD). THE SECOND IS THE

C FAN SEGMENT. IT IS FROM XLEAD TO XBOUND=XCHARN(KCHARP).

C (3) THE FAN SEGMENT IS INITIALLY DIVIDED INTO FRACG*(KFO-1) INTERVALS

C DEFINED BY THE FAMILY OF C+ CHARACTERISTIC LINES MCHAR(I) TO

C MCHAR(KCHARP). THE FLAT SEGMENT IS INITIALLY 1.

C (4) THE FLAT SEGMENT IS INITIALLY DIVIDED INTO (1-FRACG)*(KFO-1) EQUAL

C INTERVALS, AS LONG AS THEY ARE NOT SMALLER THAN THE AVERAGE

C FAN INTERVAL. WHEN THEY ARE, THEIR NUMBER IS REDUCED, BUT NOT

C BELOW THREE.

C (5) KCLEAD IS INITIALLY 1. IT IS UPDATED SO THAT THE FLAT SEGMENT

C IS AT LEAST TWICE THE AVERAGE FAN INTERVAL.


1 CONTINUE

11 CONTINUE

IF(KCLEAD.EQ.0) CALL FIN(1401)

IF(KCLEAD.EQ.KCHARP) CALL FIN(1402)

ILEAD=ILED1

KN=IHEAD+KCHARP

IF(KN.GT.KFO) CALL FIN(1411)

DX=XLEAD/DFLOAT(ILED1)

XN(I)=0.

DO 2 I=1,ILEAD

XN(I)=XN(I)+DX*DFLOAT(I-1)

2 CONTINUE

DO 3 I=ILEAD,KN

XN(I)=XCHARN(KCLEAD+I-ILEAD)

3 CONTINUE

RETURN

END

SUBROUTINE YSTEP

C SUBROUTINE NUMBER 15

IMPLICIT REAL*8(A-H,L-Z,$)

REAL*4 XPLYPL

COMMON /PLUME/XPL(1002,10),YPL(1002)

COMMON /IPLUME/KPL,ITYPL(10)

COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101).
C COMPUTE NEXT Y-STEP.
C DYNEXT IS DEFINED AS THE MINIMAL "TRIANGULATION" Y-STEP DYT, OBTAINED BY FORWARD INTERSECTION OF C-PC+
C POINTS X1,X2.

DYMIN=1.D40
DO 1 I=3,KF
X1=XF(I-1)
X2=XF(I)
DX=X2-X1
TP1=DTAN(TETA(I-1)-MUF(I-1))
TP2=DTAN(TETA(I)+MUF(I))
F1=-TP2/(TP1-TP2)
IF(R1.LT.O.) CALL FIN(1501)
DYT=F1*DX*TP1
IF(DYT.LE.0.) CALL FIN(1502)
DYMIN=DMIN1(DYMIN,STABDYT)
1 CONTINUE
DYNEXT=DYMIN
RETURN
AA~FJET1244
END

SUBROUTINE MOVE
C SUBROUTINE NUMBER 16
IMPLICIT REAL*8(A-H,L-Z,$)
REAL*4 XPL,YPL
COMMON /PLUME/XPL(1002,10),YPL(1002)
COMMON /IPLUME/KPL,ITYPL(10)
COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
1 TETA(101),BF(101),
2 XN(101),RMN(101),RPN(101),MN(101),MUN(101),
3 TETAN(101),BN(101),XTEMP(101)
COMMON /THICKY/XTH(1002),TH(1002)
COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
1 G16,G17,G18,G19,G20
COMMON /PAR/PAI,PAI2,DEG,HC,MC,EXT,FMFIN,FMFIN,MFIN,MYMAX,MYDX,MYDY,DY,DYNEXT,
1 STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPISL,NUO,
2 TETSYM,TETLIM,DYDYMAX
COMMON /STAG/RH00,NO,PO,TO,A0,MOD01
COMMON /IPAR/MAX,KF0,ITER0,KF,KN,IM,IP,J,
1 KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
COMMON /ROW/YF,YN,DXF,DYN
COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
1 RMCHARF(92),RPMCHARF(92),RMCMARN(92),RPCARN(92),
2 TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
3 CSIGNN(92),CSIGNF(92),MCHARF(92),MCHARN(92),MCHARF(92),
4 MCHARI(92)
COMMON /ICHARA/KCHARP,KCHARM1,KCHARO
C STORE NEW LINE (N) IN OLD LINE (F).
KF=KN
KF2=2*KF
YF=YN
DO 1 I=1,KN
SUBROUTINE OPACX

IMPLICIT REAL*8(A-H,L-Z,$)
REAL*4 XPL,YPL
COMMON /PLUME/XPL(1002,10),YPL(1002)
COMMON /IPLUME/KPL,ITYPL(10)
COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
1 TETAF(101),BF(101),
2 XN(101),RMN(101),RPN(101),MN(101),MUN(101),
3 TETAN(101),BFX(101),
4 XTEMP(101)
COMMON/THICKY/XTHC(1002),THC(1002)
REAL*4 XI,XI,XPIM,XIGRP,XIAPP,XIF
COMMON /THICKX/XXI(20),XI(101,20),XPIM(101,20),XIGRP(101,20),
1 XIAPP(101,20),XIF(101,20)
COMMON /GAMA/G,G1,GZ,G3,G4,G5,G6,G7,G8,G9,G10,G12,G14,G15,
1 G16,G17,G18,G19,G20
COMMON /PAR/PAI,PAIZ,DEG,XC,YC,MEXIT,MFIN,YMAX,DY,DYMAX,
1 STAB,DELTA,PSI,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NUO,
2 TETSYM,TETLIM,DDY,DYNEXT,
3 COMMON /ROWYF,YN,DXF,DXN
C COMPUTE X-OPACITY.
C BEGIN FROM LIMITING CHARACTERISTIC OF AN ASSUMED P.M. FAN.
C XIO -- THE THICKNESS BETWEEN THE LIMITING CHARACTERISTIC AND THE
C BOUNDARY CHARACTERISTIC OF THE NUMERICAL COMPUTATION.
C DO 12 I=1,KFO
XIF(I,JXI)=XF(I)
XI(I,JXI)=0.
XPIM(I,JXI)=0.
XIGRP(I,JXI)=0.
XIAPP(I,JXI)=0.
12 CONTINUE
IF(J.EQ.1) GO TO 1000
PSILIM=TETLIM
XIM=XF+(YF-YC)/DTAN(PSILIM)
XBOUND=XF(KF)
KPM=10
DX=(XIM-XBOUND)/DFLOAT(KPM)
SUM=0.
DO 1 I=1,KPM
XI=XBOUND+DFLOAT(I-1)*DX
X2=XI+DX
PSI=PAI2-DATAN((XI-XC)/(YF-YC))
PSZ=PAI2-DATAN((X2-XC)/(YF-YC))
Q1=(PS1-PSILIM)/G5
Q2=(PS2-PSILIM)/G5
IF(I.EQ.KPM) Q2=1.0-10
IF(Q2.LT.0.) CALL FIN(1701)
F1=G11*(DSIN(Q1))**2*(2.0/(G-1.0))
F2=G11*(DSIN(Q2))**2*(2.0/(G-1.0))
SUM=SUM+DX*(F1+F2)/2.0
CONTINUE
FORTRAN A1

C RE-EVALUATE XI0 FOR A RING-JET.
IF(Delta.EQ.O.) GO TO 14
M=MFIN
CALL MFUNC(M,F,ETA,TETA)
PSI=TETA+DARSINQ1. DO/N)
GOREM=1. DO+GlM*2
GOR=M*N2-1.DO
CALC MINTER(M,MM)
DELTOB=0.5D0*DSQRT(GOR)*(1.DO/METAXMETA)+DSIN(TETA)/M)/DSIN(PSI)
EVER=SIGMA*NONYC/(M*DSIN(TETA)*DSIN(PSI)*GOREM**G6)
GGG=2.DO-DELTOB3E(G+1.DO)/Z.DO
IF(DABSCGGG).GT.l.D-1O) GO TO 15
PRINT 16, DELTOB,G,GGG
CALL FIN(1715)
15 CALL TINE
EVER=EVER/GGG
XIO=EVER*(YFYC)**GGG-1.DO)/(YF/YC)
CONTINUE
XI(KF,JXI)=XIO
XIPMCKF, JXI )XI0
XIGRP(KF,JXI)=XIO
KF1=KF-1
DO 2 II=1,KF1
I=KF-II+1
X1=XF(I)
X2=XF(I-1)
DX=X1-X2
F1=1.DO/(1.DO+G1*MF(I)**2)EWG6
F2=1.DO/(1.DO+Gl*MF(I-W3E2)**0E6
DTNUM(NO*(SIGMA)*DX*E(F1+F2)/2.DO
XI(I-1,JXI)=XI(I,JXI)+DTNUM
XIPM(I-1,JXI)=1. D24
XIGRP(I-1,JXI)=1. D24
PSI=PA12-DATAN( (Xl-XC)/(YF-YC))
PSZ=PAI2-DATAN( CX2-XC)/(YF-YC))
IF(CPS2.GT.PSI1) GO TO 2
Q1=(PS1-PSILIM)/G5
Q2=(PS2-PSILIM)/G5
IF(Q1.LT.0.) CALL FIN(1711)
F1=G11*(DSIN(Q1))**2*(2.0/(G-1.0))
F2=G11*(DSIN(Q2))**2*(2.0/(G-1.0))
DTPM=(NO*SIGMA)*DX*(F1+F2)/2.0
XI(P-1,JXI)=XI(P,JXI)+DTPM
XIGRP(P-1,JXI)=1. D24
PS1=PAI2-DATAN((XI-XC)/(YF-YC))
PS2=PAI2-DATAN((X2-XC)/(YF-YC))
IF(P2.GT.PS1) GO TO 2
Q1=(PS1-PSILIM)/G5
Q2=(PS2-PSILIM)/G5
IF(Q1.LT.0.) CALL FIN(1711)
F1=G11*(DSIN(Q1))**2*(2.0/(G-1.0))
F2=G11*(DSIN(Q2))**2*(2.0/(G-1.0))
DTPM=(NO*SIGMA)*DX*(F1+F2)/2.0
XI(P-1,JXI)=XI(P,JXI)+DTPM
XIGRP(P-1,JXI)=1. D24
KCI=KCLEAD+I-ILEAD
KC2=KCI-1
IF(KC2.IE.KCLEAD) GO TO 21
39
JET PR FORTRAN

XIAPP(I,JX)=1.D24
KC=KCLEAD+(I-ILEAD)
IF(DELTA.EQ.0.) GO TO 3
IF(KC.LT.KCLEAD) GO TO 3
IF(KFXX(KF).LT.XCHARF(I)) GO TO 3
M=MCHARK(KC)
CALL MFUNC(M,F,ETA,TETA)
PSI=TETA+DARSIN(1.DO/M)
1
IF(KC.LT.KCLEAD) GO TO 5
IF(XFCI).LT.XCHARF(I) GO TO 3
M=MSCHARI(KC)
CALL MFUNC(M,F,ETA,TETA)
PSI=TETA+DARSIN(1.DO/M)
GGEII=I DO+Gl*M*3E2
GGOR=M2+1.DO
CALL HINTER(MHM)
DELTOB=0.5DO*DSQRT(GOR)*(1.DO/M*ETA+DSIN(TETA)/M)/DSIN(PSI)
3
EVER:=SIGMA/NOYC/(MXDSINT(EPSIN(TETA))*DSIN(PSI)*GOREM3(G6)
GGG=2.DO+DELTOB*(G+1.DO)/2.DO
EVER:=SIGMA/SIGMA3(NO*YC/(M*DSIN(TETA)*DSIN(PSI)*GOREM3(G6)
IF(DABS(GGG).GT.1.D-1O) GO TO 25
PRINT 26,I,KC,M,DELTOB,G,GGG
26
CONTINUE
1000 CONTINUE
RETURN
END

LOADC

SUBROUTINE LOADC
IMPLICIT REAL*$(A-H,L-Z,$)
REAL*4 XPL,YPL
COMMON /PLUME/XPL(1002,10),YPL(1002)
COMMON /IPLUME/KPL,ITYPL(10)
COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
 XIant, TETAF(101),BF(101),
 XN(101),RMN(101),RPN(101),MN(101),MUN(101),
 TETAN(101),BNI(101),XTEMP(101)
COMMON /THICKY/XTH(1002),TH(1002)
REAL*4 YXI,XI,XPIM,XIGRP,XIAPP,XIF
COMMON /THICKX/X XI(20),XI(101,20),XPM(101,20),XIGRP(101,20)
1
XIAPP(XI,JXI)=EVER*((XF/YC)**GGG-1.DO)/(XF/YC)
3
CONTINUE
1000 CONTINUE
RETURN
END

SUBROUTINE NUMBER 18
IMPLICIT REAL*$(A-H,L-Z,$)
REAL*4 XPL,YPL
COMMON /PLUME/XPL(1002,10),YPL(1002)
 COMMON /IPLUME/KPL,ITYPL(10)
 COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
 TETAF(101),BF(101),
 XN(101),RMN(101),RPN(101),MN(101),MUN(101),
 TETAN(101),BNI(101),XTEMP(101)
 COMMON /THICKY/XTH(1002),TH(1002)
 REAL*4 YXI,XI,XPIM,XIGRP,XIAPP,XIF
 COMMON /THICKX/X XI(20),XI(101,20),XPM(101,20),XIGRP(101,20)
1
XIAPP(XI,JXI)=EVER*((XF/YC)**GGG-1.DO)/(XF/YC)
3
CONTINUE
1000 CONTINUE
RETURN
END

DOUBLE PRECISION FUNCTIONNUFUNC(M)

40
C SUBROUTINE NUMBER 19
IMPLICIT REAL*8(A-H,L-Z,$)
REAL*8 XPL,YPL
COMMON /PLUME/XPL(1002,10),YPL(1002)
COMMON /IPLUME/KPL,ITYPL(10)
COMMON /VECS/XF(101),RMF(101),RF(101),MF(101),MFU(101),
1 TETAF(101),B(101),X(101),RMN(101),RFN(101),MN(101),MUN(101),
2 TETAN(101),BN(101),XTEMP(101)
COMMON/THICKY/XTH(1002),TH(1002)
REAL*4 YXI,XI,XIAPP,XIF
COMMON/THICKX/YXI(20),XI(101,20),XIPM(101,20),XIGRP(101,20)
1 XIAPP(101,20),XIF(101,20)
COMMON/GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
2 G16,Gl7,G18,G19,G20
COMMON/PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DYO,DY,DYNEXT,
1 STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NUO,
2 TETSYM,TETLIM,DDY,DYMAX
COMMON/IPAR/JMAX,KFO,ITERO,KF,KN,IM,IP,J,
1 KF2,INDEL,INDEL,JXI,ILEAD,ILEADF,KCLEAD
COMMON/ROW/YF,YN,DXF,DXN
COMMON/CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YChARN(92),
1 RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
2 CSIGNF(92),CSIGNF(92),MCARF(92),MCARN(92),
3 MCHARF(92),MCHARF(92),MCHARN(92),MCHARN(92),
4 MCHARF(92)
COMMON/ICHARA/KCHARP, KCHARM, KCHARO
C COMPUTE NU AS FUNCTION OF MACH NUMBER M.
DEFINITION OF NU HAS BEEN MODIFIED BY ADDING A CONSTANT.
THE USUAL CHOICE OF THE CONSTANT IS SUCH THAT NU=0 FOR INFINITE M.

Q=1.DO/DSQRT(MN*2-DO)
NUFUNC=NUO-(G5*DATAN(G5*Q)-DATAN(Q))
RETURN
END

C SUBROUTINE NUMBER 20
IMPLICIT REAL*8(A-H,L-Z,$)
REAL*8 KAPAOB
COMMON/GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
1 G16,Gl7,G18,G19,G20
COMMON/PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DYO,DY,DYNEXT,
1 STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NUO,
2 TETSYM,TETLIM,DDY,DYMAX
COMMON/IPAR/JMAX,KFO,ITERO,KF,KN,IM,IP,J,
1 KF2,INDEL,INDEL,JXI,ILEAD,ILEADF,KCLEAD
COMMON/ROW/YF,YN,DXF,DXN
COMMON/CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YChARN(92),
1 RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
2 CSIGNF(92),CSIGNF(92),MCARF(92),MCARN(92),
3 MCHARF(92),MCHARF(92),MCHARN(92),MCHARN(92),
4 MCHARF(92)
COMMON/ICHARA/KCHARP, KCHARM, KCHARO
C A ROUTINE FOR THE C+ DERIVATIVE DUE TO RING SYMMETRY (GRP).

KHM=51
IF(KHM.GT.101) CALL FIN(2001)
MINV0=1.DO/MEXIT
DMINV=MINV0/DFLOAT(KHM-1)
M=MEXIT
SUM=0.
KHM1=KHM-1
DO 1 I=1,KHM1
1 MF=M
MINV(I)=MINV0-DFLOAT(I-1)*DMINV
M=1.DO/MINV(I)
M=M-MF
M=M+M/2.DO
M=M
CALL MFUNC(M1,F1,ETAL,TETAL)
CALL MFUNC(M2,F2,ETAL2,TETAL2)
CALL MFUNC(M3,F3,ETAL,TETAL3)
SUM=SUM+M*DF(F1+4.DO*F2+F3)/6.DO
ETAL=ETAL3
TETA=TETA3
PSI=PSI+DARSIN(1.DO/M)
NORM=(3.DO-G)/4.DO*10**(-2.1.DO)**0.75DO/
1 0.5DO/DSQRT(GR)-DSIN(TETA)/GORMINPSI)KCI.DO4G1*M**2)**Gl4)
JET1585
HM:SUM*NORM
JET1586
HNVCI)zHM
JET1587
GOREM=l DO+G1MM2
JET1588
GOR=M**2-l DO
JET1589
DELTOB.5DO*DSQRTGOR)(.DO/CMEXITETA)+DSINTETA)/)/DSINCPSI)
JET1590
1+C(G+l.DO)/(2.DON(3.DO-G)))*HM JET1591
EPSIOB=DELTOI/DSQRTCGOR)-DSIN(TETA)/(M*DSINCPSI))
JET1592
KAPAOBl
JET1593
IF(DABS(PAI2-TETA).GT.l.D-6)
JET1594
lKAPAOB:DTAN(TETA)IEEPSIOB
JET1595
LAMDOB=EPSIOB-DELTOBEGOREM/(OOR*DSQRT(GOR))
JET1596
PRINT 11, I,M,HM,TETA*DEG,PSI*DEG
JET1597
11 FORtIAT(/lX,' I,M,HM,TETA,PSI=',I5,5D12.4)
JET1598
PRINT l2,DELTOB,EPSIOB*DEG,KAPAOB*DEG,LAMDOB*DEG
JET1599
12 FORMAT( lX,'DELTOB,EPSIOB,KAPAOB,LAMDOB=I,5X,5D12.4)
JET1600
1 CONTINUE JET 1601
MHINV(KHM)0O.
JET1602
IMMV(KHMz1.DO
J ET16 03
RETURN
JET1604
END
MNCJET1605
SUBROUTINE MFUNC(M,F,ETA,TETA)
JET1606
C SUBROUTINE NUMBER 21
IMPLICIT REAL*8(A-H,L-Z,*)
COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,JET1607
1 G16,G17,G18,G19,G20 JET1608
COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DYO, DY, DYNEXT,
JET1609
1 STAB,DELTA,PSII,PSIF,ZETAI,SIGMA,FRACG,EPSIL,NUO,
JET1610
2 NUP1,TETSYM
JET1611
COMMON /IPAR/JMAX,KFO,ITERO,KF,KN,IM,IP,J,
JET1612
1 IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
JET1613
COMMON /GRP/DMINV,MHINV(101),HMV(101)
JET1614
C COMPUTE H(M) BY INTERPOLATION
MINV=1.DO/M
JET1615
I=KHM-IDINT(MINV/DMINV-1.D-9)-1
JET1616
IF(G1.GE.1.AND.I.LT.KHM)
JET1617
GO TO 11
JET1618
IF(I.GE.1.AND.I.LT.KHM)
JET1619
GO TO 11
JET1620
IF(I.GE.1.AND.I.LT.KHM)
JET1621
GO TO 11
JET1622
IF(I.GE.1.AND.I.LT.KHM)
JET1623
GO TO 11
JET1624
IF(I.GE.1.AND.I.LT.KHM)
JET1625
GO TO 11
JET1626
RETURN
JET1627
END
JET1628
SUBROUTINE MACH(M0,A,0IAI
JET1629
C SUBROUTINE NUMBER 22
IMPLICIT REAL*8(A-H,L-Z,*)
COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,JET1630
1 G16,G17,G18,G19,G20 JET1631
COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DYO, DY, DYNEXT,
JET1632
1 STAB,DELTA,PSII,PSIF,ZETAI,SIGMA,FRACG,EPSIL,NUO,
JET1633
2 NUP1,TETSYM
JET1634
COMMON /IPAR/JMAX,KFO,ITERO,KF,KN,IM,IP,J,
JET1635
1 IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
JET1636
COMMON /GRP/DMINV,MHINV(101),HMV(101)
JET1637
COMMON /IGRP/KHM
JET1638
C FORTRAN A1
IMPLICIT REAL*8(A-H,L-Z,$)
COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
1 TETA(101),BF(101),
2 XN(101),RMN(101),RPN(101),MN(101),MUN(101),
3 TETAN(101),BN(101),XTEMP(101)
COMMON /ROW4YF, YN,
4 DXF, DXN
5 COMMON /GAMA/G,G1,02,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
6 G16,G17G18,619,G20
COMMON /IPAR/JMAX,KFO,ITERO,KF,KN,IM,IP,J,
7 KF2,IDEL,JDEL,JI,ILEAD,ILEADF,KCLEAD
COMMON /GRP/DMINV,MHINV(101),HMV(101)
COMMON /IGRP/KHM
C COMPUTE K(M) AND THE ALFA-DERIVATIVES
M=MOB
CALL MFUNC(M,F,ETA,TETA)
PSI=TETA+DARSIN(1.DO/M)
CALL HINTER(M,HM)
GOREM=1.DO+G1*KM**2
GOR=KM**2-1.DO
DELTOB=0.5D0*DSQRT(GOR)*X1.DO/(MEXIT*ETA)+DSIN(TETA)/M)/DSIN(PSI)
1 +G15*HM/2.DO
FAB=F0B(YF/YC)*DELTOB
CALL AREAF(FAB,MAB)

C COMPUTE MABI FROM THE INVERSE PROBLEM SOLUTION
COTAV=(XF(I)-XC)/(YF-YC)
PSIO=PA12-DATAN(COTAV)
EVY=YF/LOG2(YF/YC)/(YF-YC)-1.DO
PSIN=PSIO
DO 1 ITER=1,50
PSI=PSIN
M=DSQRT(1.DO+G4/TANCCPSI-TETLIM)/G5)**2)
M=DMAXI(M,MEXIT)
CALL HINTER(M,HM)
CALL MFUNCCM,F,ETA,TETA)
GOREM=I.DO+G1*M**2
GOR=M5EM2-
DELTGB=0.5D03(DSQRT(GOR)3*(1.DO/(MEXIT*ETA)+DSIN(TETA)/M)/DSIN(PSI)
1 +G15*HM/2.DO
EPSIOB=DELTOB/DSQRT(GOR)-DSIN(TETA)/M/DSIN(PSI)
LAMDOB=EPSIOB-DELTOB*GOREM/GOR*DSQRT(GOR)
COTN=COTAV+LAMDOB/DSQRT(GOR)
PSIN=PSIO
EPSIN=EPSIN-PSI
DPSIN=PSIN-PSIN
IF(DABS(DPSIN).LT.1.D-9)
GO TO 11
1 CONTINUE
PRINT 12, I,ITER,PSI,PSIN,DPSI,M,XF(I),YF,XC,YC
12 FORMAT(/1X,'IITER,PSI,PSIN,DPSI,M,XF(I),YF,XC,YC=',//
1 1X,214,8D13,/)  
CALL FIN(2301)
11 CONTINUE
C USING MOBI=M AS COMPUTED FROM THE INVERSE PROBLEM, FIND MABI.
MOBI=M
M=MOBI
CALL MFUNC(M,F,ETA,TETA)
PSI=TETA+DARSIN(1.DO/M)
CALL HINTER(M,HM)
GOREM=1.DO+G1*KM**2
GOR=KM**2-1.DO
DELTOB=0.5D0*DSQRT(GOR)*X1.DO/(MEXIT*ETA)+DSIN(TETA)/M)/DSIN(PSI)
1 +G15*HM/2.DO
F0B=G07*GOREM)*KG2/M
FAB=F0B(YF/YC)*DELTOB
CALL AREAF(FAB,MAB)
RETURN
END

SUBROUTINE AREAF(F,M)
IMPLICIT REAL*8(A-H,L-Z,$)
C SUBROUTINE NUMBER 24
SUBROUTINE AREA~F
COMMON /GAMA/G, G1, G2, G3, G4, G5, G6, G7, G8, G9, G10, G11, G12, G13, G14, G15, JET1729
1 G16, G17, G18, G19, G20
COMMON /PAR/ PAI, PAI2, DEG, XC, YM, X, YMAX, DY0, DY, DY, DYNEXT,
1 STAB, DELTA, PSI1, PSI, ZETA1, SIGMA, FRACG, EPSIL, NU0,
2 TETSYM, TETLIM, DDDY, DYMAX
COMMON /IPAR/ JMAX, KF, ITER0, KF, KN, IM, IP, J,
1 KF2, IDEL, JDEL, JYXI, JXI, ILEAD, ILEADF, KCLEAD
COMMON /GRO/ DMINV, MINV(101), MINV(101)
COMMON /KHM/ C
*.COMPUTE MACH NUMBER M FROM AREA RATIO FUNCTION F
C F=((2/(G+1)))((1+(G-1)*M**2))**((G+1)/(2*(G-1)))/M
C INITIAL GUESS IS MIN
E1= CF*MEXIT)**(1.0/G7)
E2=(E1-1.0)/G1
E3=DMAX1(E2, MEXIT**2)
MIN=DSQRT(E3)
EMN=MIN
DO 1 I=1, 100
EMO=EMN
GOREM=1.0+G1*EMO**2
GOR=EMO**2-1.0
FO=(G7*GOREM)**G2/EMO
DF=FO-F
*.PRINT 123, I, EMO, EMN, FO, F, DF, GOR, GOREM
C123 FORMAT(1X,'I, EMO, EMN, FO, F, DF, GOR, GOREM=',I5,7D12.4)
DMDN=DF/GOR**2
EMN=EMO-DMDN
EMN=EMO-DM0
EPSEM=DABS(EMN/EMO)
IF(EPSEM.LT.1.D-10) GO TO 11
1 CONTINUE
11 CONTINUE
CALL FIN(2401)
M=EMN
RETURN
END
5. REFERENCES


6. DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>No.</th>
<th>Name and Address</th>
</tr>
</thead>
</table>
| 1.  | Defense Technical Information Center  
Cameron Station  
Alexandria, VA 22314 | 2 |
| 2.  | Library, Code 0142  
Naval Postgraduate School  
Monterey, CA 93943-5100 | 2 |
| 3.  | Department Chairman, Code 67  
Department of Aeronautics  
Naval Postgraduate School  
Monterey, CA 93943-5100 | 1 |
| 4.  | Distinguished Professor Allen E. Fuhs  
Space Systems Academic Group, Code 72  
Naval Postgraduate School  
Monterey, CA 93943-5100 | 5 |
| 5.  | Dr. Neil Griff  
SDIO DEO  
Washington, DC 20301-7100 | 3 |
| 6.  | Mr. Bruce Pierce  
SDIO DEO  
Washington, DC 20301-7100 | 1 |
| 7.  | Dr. Joseph Falcovitz  
Code 72  
Naval Postgraduate School  
Monterey, CA 93943-5100 | 5 |
<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Address</th>
<th>City, State Zip Code</th>
<th>Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Professor Max F. Platzer</td>
<td>Department of Aeronautics, Code 67</td>
<td>Monterey, CA 93943-5100</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>Professor Oscar Biblarz</td>
<td>Department of Aeronautics, Code 67</td>
<td>Monterey, CA 93943-5100</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>Professor David W. Netzer</td>
<td>Department of Aeronautics, Code 67</td>
<td>Monterey, CA 93943-5100</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>Research Administration Office</td>
<td>Code 012</td>
<td>Monterey, CA 93943-5100</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>Dr. P. Avizonis</td>
<td>Air Force Weapons Laboratory</td>
<td>Kirtland Air Force Base, NM 87117</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>Dr. John Lawless</td>
<td>Space Power Inc.</td>
<td>San Jose, CA 95131</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>Dr. Mark Thornton</td>
<td>Boeing Aerospace Company</td>
<td>Seattle, WA 98124-2499</td>
<td>1</td>
</tr>
<tr>
<td>No.</td>
<td>Name</td>
<td>Address</td>
<td>Copies</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----------------------------</td>
<td>----------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>LT. Mark Price</td>
<td>AFRPL</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Edwards AFB, CA 93523</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Mr. Arthur W. Rogers</td>
<td>Space Systems Division</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hughes Aircraft Co.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P. O. Box 92919, Los Angeles, CA 90009</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>LCOL Rick Babcock, USAF</td>
<td>Air Force Geophysical Laboratory</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hanscomb Field</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bedford, MA 01730</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Dr. James Stark Draper</td>
<td>Aerodyne Research, Inc.</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>45 Manning Road</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Billerica, MA 01821</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
END
9-87
DTIC