NAP: A Computer Program for the Computation of Two-Dimensional, Time-Dependent, Inviscid Nozzle Flow

by

Michael C. Cline

Los Alamos Scientific Laboratory
Los Alamos, New Mexico 87545

Issued: January 1977

LASL Logo
[Placeholder]

Prepared for the United States Energy Research and Development Administration
Contract W-7409-ENG. 36

An Affirmative Action/Equal Opportunity Employer
Work supported by the US Energy Research and Development Administration.
Magnetic Fusion Energy Division.

DISCLAIMER

Contents

ABSTRACT

A computer program, NAP, is presented for calculating inviscid, steady, and unsteady flow in two-dimensional and axisymmetric nozzles. Interior mesh points are computed using the MacCormack finite-difference scheme, while a characteristic scheme is used to calculate the boundary mesh points. An explicit artificial viscosity term is included for shock computations. The fluid is assumed to be a perfect gas. This method was used to compute the steady flow in a 45°-15° conical, converging-diverging nozzle, a 15° conical, converging nozzle, and a 10° conical, plug nozzle. The numerical solution agreed well with the experimental data. In contrast to previous time-dependent methods for calculating steady flows, the computational times were < 1 min on a CDC 6600 computer.

Chapter 1

Basic Description of the Method

1.1 Introduction

The equations of motion governing steady, inviscid flow are of a mixed type: hyperbolic in the supersonic region and elliptic in the subsonic region. These mathematical difficulties may be removed by using the "time-dependent" method, in which the flow is assumed to be unsteady or time-dependent. Then the governing equations are hyperbolic in both subsonic and supersonic regions. The steady-state solution may be obtained as the asymptotic solution for large time. This time-dependent technique has been used to compute steady converging-diverging nozzle flows (reported in Refs. 1–6), and it has also been used to compute steady converging nozzle flows (see Refs. 4 and 7). The results of those calculations are mainly good, but the computational times are rather large. In addition, although the computer program of Ref. 6 included a centerbody and those of Refs. 4 and 7 included the exhaust jet, none of the above codes is able to calculate both, that is, plug nozzles.

The object of this research was to develop a production-type computer program capable of solving steady converging, converging-diverging, and plug two-dimensional nozzle flows in computational times of < 1 min on a CDC 6600 computer. Such a program would be able to solve unsteady flows as well.

1.2 Literature Review

The following is a discussion of the methods used in References 1 through 7. The first paragraph deals with the computation of the interior mesh points; the next three paragraphs are concerned with the boundary mesh points.

Prozan (see Ref. 1), Wehofer and Moger, and Laval used variations of the two-step Lax-Wendroff scheme to compute the interior mesh points. Migdal *et al.* and Brown and Ozcan employed the original one-step Lax-Wendroff scheme, but with the equations of motion in nonconservation form. Serra applied the original Lax-Wendroff scheme with the equations of motion in conservation form. To stabilize their schemes, Laval and Serra used artificial viscosity terms in their difference equations.

Wehofer and Moger reset the stagnation conditions along each streamline, reset the mass flow at each axial location, and smoothed the subsonic portion of the flow after each time step.

To compute the nozzle inlet mesh points, Prozan (in Ref. 1) assumed the inlet flow to be uniform. Wehofer and Moger assumed only that the pressure was radially uniform at the inlet. Migdal *et al.* and Brown and Ozcan mapped the inlet to minus infinity after Moretti, thus allowing the static conditions to be set equal to the stagnation conditions. Laval used extrapolation of the interior mesh points to determine the inlet mesh points, while Serra employed a characteristic scheme.

Prozan (in Ref. 1), Wehofer and Moger, Laval, and Brown and Ozcan used an extrapolation technique to compute the wall mesh points. Migdal *et al.* employed a characteristic scheme after Moretti to compute the wall mesh points, while Serra applied a reflection technique. For the converging nozzle problem to be properly posed, an exhaust jet calculation must be included. Wehofer and Moger used an extrapolation procedure to compute the exhaust jet boundary mesh points, while Brown and Ozcan employed a characteristic scheme after Moretti.

All of the above authors used extrapolation to compute the exit mesh points when the flow was supersonic, since any errors incurred would be swept out of the mesh. Serra employed a characteristic scheme when the exit flow was subsonic.

1.3 Choice of a Method

The lengthy computational times associated with time-dependent calculations are usually caused by inefficient numerical schemes or poor treatment of boundaries, resulting in the requirement for excessively fine computational meshes (see Refs. 8 and 9). A technique for a much more efficient calculation of the interior and boundary mesh points will be discussed here.

The computation of steady flows by a time-dependent method differs from ordinary initial-value problems in that the initial data and much of the transient solution have a negligible effect on the final or steady solution. Therefore, accuracy is important only for the asymptotic state, and special attention to intermediate efficiency will result in reasonable computational times. For this reason, interior mesh points can be computed by using a very efficient finite-difference scheme, as opposed to those less efficient finite-difference or characteristic schemes that achieve high accuracy at every step.

In the class of finite-difference schemes, the two-step methods such as the MacCormack and the two-step Lax-Wendroff schemes are more efficient than the original Lax-Wendroff scheme, especially if the governing equations are in conservation form. Moretti showed that using the equations of motion in conservation form decreased efficiency and ease of programming while only slightly increasing the accuracy of shock calculations. The use of an explicit artificial viscosity term for shock-free flows also decreases efficiency and was shown to be physically unjustified. In addition, such increases in the numerical dissipation can often destroy the weak shock structure of transonic flows. Therefore, the MacCormack scheme with the equations of motion in nonconservation form is used to calculate the interior mesh points. An explicit artificial viscosity term was included for shock computations only. Remember that the implicit dissipation always present as an effect of truncation terms assures numerical stability for the shock-free flow results.

The boundary mesh points, while making up only a small part of the total mesh points, must be handled most accurately, because of the flowfield's sensitivity to precise boundary geometry. Moretti and Abbett showed that reflection, extrapolation, and one-sided difference techniques for computing solid wall boundaries give poor results and should be avoided. Therefore, the wall and centerbody mesh points are computed using a characteristic scheme. A characteristic scheme is also used to calculate the exhaust jet boundary mesh points.

In the case of the nozzle inlet mesh points for subsonic flow, the use of extrapolation techniques and the assumption of one-dimensional flow presume the form of the solution and in many cases are physically unjustified. On the other hand, a characteristic scheme could be used to calculate the inlet mesh points. While the stagnation pressure and temperature are assumed to remain constant at the inlet in a characteristic scheme (not necessarily the case for unsteady flow), this assumption would appear to be valid for the time-dependent calculation of steady flows. Moretti recommends mapping the inlet to minus infinity, thus allowing the static conditions to be set equal to the stagnation conditions. In theory, this appears to be the best approach, but it should be kept in mind that the infinite physical plane must be replaced by a finite computational plane. Also, this technique requires additional mesh points upstream of the nozzle inlet. It is not presently resolved as to whether the characteristic scheme approach used by Serra or the mapping-to-minus-infinity approach suggested by Moretti and employed by Migdal *et al.* and Brown and Ozcan is the best technique. To reduce the total number of mesh points to be computed, a characteristic scheme is used to compute the inlet mesh points. For supersonic flow, the inlet mesh points are set equal to specified values of velocity, pressure, and density, because in a supersonic stream the downstream conditions do not propagate upstream. Extrapolation is used to compute the exit mesh points when the flow is supersonic, since any errors incurred will be swept out of the mesh, and a characteristic scheme is employed when the flow is subsonic.

1.4 Equations of Motion

The appropriate non-conservation form of equations for two-dimensional, inviscid, isentropic, rotational flow are:

$$\rho_t + u\rho_x + v\rho_y + \rho u_x + \rho v_y + \epsilon \rho v/y = 0 \tag{1.1}$$

$$u_t + uu_x + vu_y + p_x/\rho = 0 \tag{1.2}$$

$$v_t + uv_x + vv_y + p_y/\rho = 0 \tag{1.3}$$

$$p_t + up_x + vp_y - a^2(\rho_t + u\rho_x + v\rho_y) = 0 \tag{1.4}$$

where ρ is the density, u is the axial velocity, v is the radial velocity, p is the pressure, a is the local speed of sound, t is the time, x and y are the axial and radial coordinates, and the subscripts denote partial differentiation. The symbol ϵ is 0 for planar flow and 1 for axisymmetric flow.

1.4.1 Coordinate Transformation

The physical (x, y) plane is mapped into a rectangular computational plane (ζ, η) by the following coordinate transformation:

$$\zeta = x; \quad \eta = \frac{y - y_c(x)}{y_w(x,t) - y_c(x)}; \quad \tau = t \tag{1.5}$$

where $y_w(x,t)$ denotes the nozzle wall and exhaust jet boundary radius as a function of x and t and $y_c(x)$ denotes the nozzle centerbody radius as a function of x. These mapping functions must be single-valued functions of the x coordinate. In the (ζ, η, τ) coordinate system Equations (1.1)–(1.4) become:

$$\rho_\tau + u\rho_\zeta + \bar{v}\rho_\eta + \rho u_\zeta + \rho\alpha u_\eta + \rho\beta v_\eta + \epsilon\rho v/(y_c + \eta/\beta) = 0 \tag{1.6}$$

$$u_\tau + uu_\zeta + \bar{v}u_\eta + p_\zeta/\rho + \alpha p_\eta/\rho = 0 \tag{1.7}$$

$$v_\tau + uv_\zeta + \bar{v}v_\eta + \beta p_\eta/\rho = 0 \tag{1.8}$$

$$p_\tau + up_\zeta + \bar{v}p_\eta - a^2(\rho_\tau + u\rho_\zeta + \bar{v}\rho_\eta) = 0 \tag{1.9}$$

where

$$\beta = \frac{1}{y_w - y_c} \tag{1.10}$$

$$\alpha = -\beta\frac{\partial y_c}{\partial x} - \left(\frac{\partial y_w}{\partial x} - \frac{\partial y_c}{\partial x}\right) \tag{1.11}$$

$$\delta = -\eta\beta\frac{\partial y_w}{\partial t} \tag{1.12}$$

and

$$\bar{v} = \alpha u + \beta v + \delta \tag{1.13}$$

The fluid is assumed to be thermally and calorically perfect; that is, a constant ratio of specific heats is assumed.

1.4.2 Artificial Viscosity for Shock Computations

For shock computations, an artificial viscosity model of the form suggested by von Neumann-Richtmyer is used. This model, which has a term corresponding to all the viscous and thermal conduction terms in the Navier-Stokes equations, is shown below.

$$[\text{RHS Eq. (2)}] = (\lambda + 2\mu)\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) + \lambda\frac{\partial}{\partial x}\left(\frac{\partial v}{\partial y}\right) + \frac{\epsilon}{y}\left[(\lambda+\mu)\frac{\partial v}{\partial x} + \mu\frac{\partial u}{\partial y}\right] \tag{1.14}$$

$$[\text{RHS Eq. (3)}] = (\lambda+2\mu)\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x}\right) + \lambda\frac{\partial}{\partial y}\left(\frac{\partial v}{\partial y}\right) + \mu\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + \frac{\epsilon(\lambda+2\mu)}{y}\left(\frac{\partial v}{\partial y} - \frac{v}{y}\right) \tag{1.15}$$

where $c_\mu = c_\lambda$ are nondimensional quantities that specify the distribution and amount of smoothing. **[Note: Complete artificial viscosity formulation to be filled from Section II of original report.]**

1.5 Numerical Method

The computational plane is divided into five sets of mesh points: interior, inlet, exit, wall and centerbody, and exhaust jet boundary.

1.5.1 Interior Mesh Points

The interior mesh points are computed using the MacCormack scheme, a second-order, non-centered, two-step, finite-difference scheme. Backward differences are used on the first step; forward differences are used on the second. The governing equations are left in non-conservation form. An explicit artificial viscosity term is used for shock computations. Centerline mesh points are computed by enforcing symmetry of the flow. For example, the finite-difference equations for Equation (1.1) for planar flow ($\epsilon = 0$) and no artificial viscosity are:

$$\bar{\rho}^{N+1}_{L,M} = \rho^N_{L,M} - \left[u^N_{L,M}\left(\frac{\rho^N_{L,M} - \rho^N_{L-1,M}}{\Delta x}\right) + v^N_{L,M}\left(\frac{\rho^N_{L,M} - \rho^N_{L,M-1}}{\Delta y}\right) + \rho^N_{L,M}\left(\frac{u^N_{L,M} - u^N_{L-1,M}}{\Delta x}\right) + \rho^N_{L,M}\left(\frac{v^N_{L,M} - v \cdots}{\Delta y} \cdots \right.\right. \tag{1.16}$$

$$\rho^{N+1}_{L,M} = 0.5\left[\rho^N_{L,M} + \bar{\rho}^{N+1}_{L,M} - \left[\bar{u}^{N+1}_{L,M}\left(\frac{\bar{\rho}^{N+1}_{L+1,M} - \bar{\rho}^{N+1}_{L,M}}{\Delta x}\right) + \bar{v}^{N+1}_{L,M}\left(\frac{\bar{\rho}^{N+1}_{L,M+1} - \bar{\rho}^{N+1}_{L,M}}{\Delta y}\right) + \bar{\rho}^{N+1}_{L,M}\left(\frac{\bar{u}^{N+1}_{L+1,M} - \bar{u}^{N+1}_{L,M}}{\Delta x}\right) + \cdots \right.\right. \tag{1.17}$$

where L and M denote axial and radial mesh points, respectively, N denotes the time step, and the bar denotes values calculated on the first step.

1.5.2 Inlet Mesh Points

The inlet mesh points for subsonic flow are computed using a second-order, reference-plane characteristic scheme. In this scheme, the partial derivatives with respect to η are computed in the initial-value and solution surfaces using non-centered differencing as in the MacCormack scheme. These approximations to the derivatives with respect to η are then treated as forcing terms and the resulting system of equations is solved in the $\eta =$ constant reference planes using a two-independent-variable, characteristic scheme.

The boundary condition is the specification of the stagnation temperature and stagnation pressure. The use of a reference-plane characteristic scheme requires the specification of inlet flow angle as an additional boundary condition. The inlet flow angle can be approximately determined from the nozzle geometry. The equations relating the total and static conditions are:

$$p_T/p = \left[1 + (\gamma - 1)M^2/2\right]^{\gamma/(\gamma-1)} \tag{1.18}$$

$$T_T/T = 1 + (\gamma - 1)M^2/2 \tag{1.19}$$

where γ is the ratio of specific heats, M is the Mach number, T is the temperature, and the subscript T denotes the total conditions.

For supersonic flow, the inlet mesh points are set equal to specified values of velocity, pressure, and density.

1.5.3 Exit Mesh Points

For subsonic flow, a reference-plane characteristic scheme similar to the inlet scheme is used. The exit pressure is specified.

For supersonic flow, the exit mesh points are computed using linear extrapolation.

1.5.4 Wall and Centerbody Mesh Points

The wall and centerbody mesh points are computed using a reference-plane characteristic scheme. The wall and centerbody contours and therefore their slopes are specified. The boundary condition is given by:

$$v = u \tan(\theta) + \frac{\partial y_w}{\partial t} \tag{1.20}$$

where θ is the local wall or centerbody angle.

1.5.5 Exhaust Jet Boundary Mesh Points

The exhaust jet boundary mesh points are computed by the wall routine such that the pressure boundary condition:

$$p = p_{\text{ambient}} \tag{1.21}$$

is satisfied. This is accomplished by first assuming the shape of the jet boundary and then using the wall routine to calculate the pressure. Next, the jet boundary location is slightly changed and a second pressure is computed. By use of an interpolation procedure, a new jet boundary location is determined. This interpolation-extrapolation procedure is then repeated at each point until the jet boundary pressure and the ambient pressure agree within some specified tolerance.

1.5.6 Time Step Control

The step size Δt is controlled by the well-known Courant or CFL condition, which can be expressed as:

$$\Delta t \leq \frac{1}{\left[(V+a)\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}\right)^{1/2}\right]} \tag{1.22}$$

where V is the velocity magnitude. Using the coordinate transformation, Equation (1.22) becomes:

$$\Delta \tau \leq \frac{A}{\left[(V+a)\left(\frac{1}{\Delta \zeta^2}+\frac{\beta^2}{\Delta \eta^2}\right)^{1/2}\right]} \tag{1.23}$$

where the coefficient A was determined from actual calculations and varied between 0.4 and 1.6 depending on the geometry of the flow in question.

1.6 Overall Program Capabilities

The nozzle inlet flow, as well as the flow leaving the nozzle, may be either subsonic or supersonic. The flow may contain variations in stagnation temperature and stagnation pressure from streamline to streamline. The nozzle wall and centerbody geometries may be either one of two analytical contours or a completely general tabular contour. The program is capable of calculating the exhaust jet boundary for subsonic or supersonic flow. The initial data may be read in or calculated internally by the program. The internally computed data are calculated assuming one-dimensional, steady, isentropic flow with area change. The program output includes the coordinates, velocities, pressure, density, Mach number, temperature, mass flow, and axial thrust in both English and metric units.

1.7 Results and Discussion

The results presented here have been adopted from experimental validation work. The CDC 6600 computational times represent the central processor time not including compilation. So that these results can be compared with those of other investigators, the following table of relative machine speeds is provided:

Table 1.1: Relative Machine Speeds Compared to CDC 6600

Computer	**Relative Machine Speed**
IBM 7094	0.1
IBM 360/50	0.1
IBM 360/64	0.3
IBM 360/75	0.5
Univac 1108	0.5
CDC 6600	1.0

The validation cases are presented below.

1.7.1 Case 1: 45°-15° Conical Converging-Diverging Nozzle

Figure 1a: Nozzle Geometry (45°-15° Converging-Diverging)
[Placeholder - Original image required]

Figure 1.1: 45°-15° Converging-Diverging Nozzle Geometry

The present method was used to compute the steady-state solution for flow in the 45°-15° conical, converging-diverging nozzle. A 21×8 computational mesh required 301 time planes and a computational time of 35 seconds. The experimental data are those of Cuffel et al. (Ref. 2). The computed discharge coefficient is 0.983, compared with the experimental value of 0.985.

Figure 2: Mach Number Contours and Wall Pressure
45°-15° Converging-Diverging Nozzle
[Placeholder - Original image required]

Figure 1.2: Mach Number Contours and Wall Pressure Ratio for 45°-15° Conical Converging-Diverging Nozzle

There is good agreement with the experimental data. This case was also solved by other researchers including Prozan, Migdal, Laval, and Serra, with computational times ranging from 45 minutes to 2 hours on various computer systems.

1.7.2 Case 2: 15° Conical Converging Nozzle

Figure 1b: Nozzle Geometry (15° Converging)
[Placeholder - Original image required]

Figure 1.3: 15° Conical Converging Nozzle Geometry

The present method was also used to compute the steady-state flow in a 15° conical, converging nozzle. The nozzle geometry is shown in Figure 1.3. A 23×7 computational mesh required 249 time planes and a computational time of 29 seconds. The experimental data are those of Thornock (Ref. 17). The computed discharge coefficient is 0.957, compared with the experimental value of 0.960.

Figure 3: Mach Number Contours and Wall Pressure
15° Converging Nozzle
[Placeholder - Original image required]

Figure 1.4: Mach Number Contours and Wall Pressure Ratio for 15° Conical Converging Nozzle

There is good agreement with the experimental data. This case was also solved by Wehofer and Moger and Brown and Ozcan, with Wehofer and Moger requiring over 2 hours on an IBM 360/50 (47×11 mesh) and Brown and Ozcan requiring 17 minutes on an IBM 360/65 (20×6 mesh).

1.7.3 Case 3: 10° Conical Plug Nozzle

Figure 1c: Nozzle Geometry (10° Plug)
[Placeholder - Original image required]

Figure 1.5: 10° Conical Plug Nozzle Geometry

Finally, the present method was used to calculate the flow in a 10° conical, plug nozzle. The nozzle geometry is shown in Figure 1.5. A 31×6 computational mesh required 327 time planes and a computational time of 52 seconds. The experimental data are those of Bresnahan and Johns (Ref. 18).

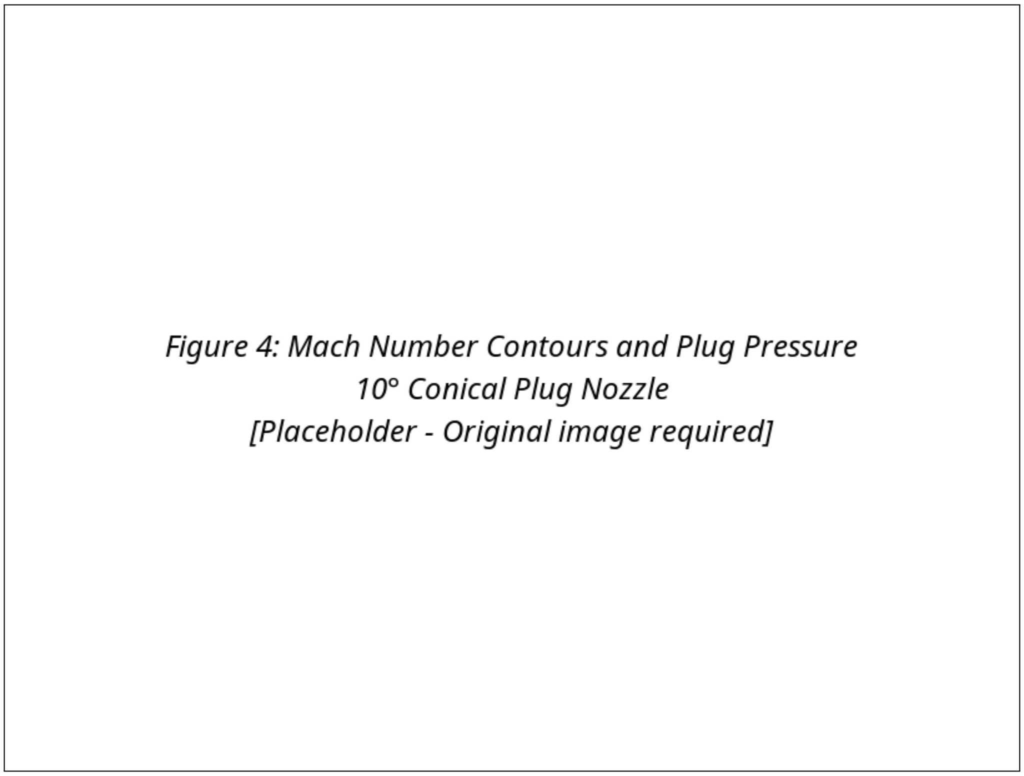

Figure 1.6: Mach Number Contours and Plug Pressure Ratio for 10° Conical Plug Nozzle

Again, there is good agreement with the experimental data. The author is unaware of any other time-dependent analyses of plug nozzles.

1.8 Concluding Remarks

A method of computing nozzle flows has been presented. A production-type computer program capable of solving a wide variety of nozzle flows has been developed. The program's accuracy was demonstrated by computing the steady flow in the three test cases above. The sub-one-minute computational times for these steady flows is considerably faster than for any of the earlier time-dependent techniques.

Chapter 2

Program Description and Usage

2.1 Program Structure

The NAP computer program consists of one main program, one function, and twelve subroutines. The program structure follows this execution flow:

1. **Program MAIN:** Initiates the run by reading input data, printing the program title and abstract, and converting units. Calls geometry subroutines and performs the main time-stepping loop.
2. **GEOM and GEOMCB:** Calculate the nozzle geometry for fixed wall and optional center-body configurations.
3. **ONEDIM:** Calculates the one-dimensional isentropic initial-value surface using a Newton-Raphson scheme to find Mach numbers from area ratios.
4. **MAP:** Calculates mapping functions that transform the physical plane to a rectangular computational plane.
5. **INTER:** Performs the interior mesh calculations using the MacCormack finite-difference scheme.
6. **INLET, WALL:** Implement boundary conditions using second-order characteristic-based schemes.
7. **MASFLO:** Calculates mass flow and thrust during the solution.
8. **PLOT:** Generates film plots of solution contours and velocity vectors.

2.2 Input Data Format

Input data are provided via Fortran NAMELIST format. The primary input namelists are:

NAMELIST /CNTRL/: Control Parameters

Key parameters controlling the simulation:

LMAX, MMAX Grid dimensions (ξ and η directions)

NMAX Maximum number of time steps

NPRINT Print frequency (0 = final solution only, $n > 0$ = every n steps)

TCONV Convergence criterion for steady-state detection

FDT Frequency for time-step recalculation

TSTOP Simulation stop time

GAMMA Specific heat ratio ($\gamma = 1.4$ for air)

RGAS Gas constant (53.35 for air in English units)

NASM Number of inlet stagnation point profiles (1 or > 1 for variable inlet conditions)

IUNIT Unit conversion flag (0 = English, 1 = SI)

NAMELIST /GEMTRY/: Geometry Parameters

Nozzle geometry definition:

NDIM Dimension flag (0 = 2D axisymmetric, 1 = 2D Cartesian)

NGEOM Geometry type (1 = converging, 2 = converging-diverging, 3 = plug)

XI, XE Inlet and exit axial coordinates

RI, RE Inlet and exit radii

RCI, RCT, RCE Centerbody inlet, throat, and exit radii (if centerbody present)

ANGI, ANGE Inlet and exit half-angles (degrees)

NWPTS Number of wall definition points

NAMELIST /BC/: Boundary Conditions

Inlet boundary condition parameters:

PT Stagnation pressure profile (array of NASM values)

TT Stagnation temperature profile (array of NASM values)

THETA Inlet flow angle profile (degrees)

PE Exit static pressure

NSTAG Stagnation profile flag (0 = uniform, > 0 = radial variation)

ISUPER Supersonic inlet flag (0 = subsonic, 1 = supersonic)

2.3 Output Description

The program produces output in three forms:

1. Printed Output

ASCII output containing:

- Program header and version information
- Echo of input parameters (CNTRL, GEMTRY, BC namelists)
- Initial geometry and one-dimensional surface calculations
- Iteration history with time, time-step size, and convergence measures (if requested)
- Final solution statistics including mass flow, momentum, and thrust

2. Film Plots

Vector plots and contour plots on graphics film (if NPLOT ≥ 0):

- Velocity vectors at each solution time
- Contours of Mach number, pressure, density
- Wall streamline positions

3. Punched Card Output (Optional)

Fortran unformatted binary restart deck for continuing previous runs.

2.4 Sample Calculations

Three nozzle geometries have been analyzed and serve as test cases:

Case 1: Converging-Diverging Nozzle

A 45°–15° converging-diverging geometry with uniform inlet conditions at stagnation pressure $P_T =$ 13.78 psia and temperature $T_T = 530$ °R. Results show excellent agreement with one-dimensional theory at the throat and quasi-2D behavior in the diverging section.

Case 2: Converging Nozzle

A simple 15° converging geometry with identical inlet conditions. Used to test subsonic inlet conditions and convergent-only nozzles.

Case 3: Plug Nozzle

A complex plug nozzle configuration with variable centerbody. Comparison with experimental data of Bresnahan and Johns (Reference 18) shows good agreement in gross features including shock structure at off-design conditions.

For detailed sample input and output listings, refer to the original NAP documentation. The program is controlled entirely via namelist input which provides flexibility for analyzing various nozzle configurations and inlet conditions.

Case No. 3 Output (Converted from Original Fig. 12)

Editorial conversion note: In the original report, Figure 12 is a multi-page line-printer output block (input summary, run progression, and solution-surface tables). In this conversion, it is represented as tabulated/listing content for readability and traceability rather than as a graphical figure.

Run Progression (from Fig. 12 continuation page)

Index	N
1	90
2	100
3	110
4	120
5	130
6	140
7	150
8	160
9	170
10	180
11	190
12	200

13	210
14	220
15	230
16	240
17	250
18	260
19	270
20	280
21	290
22	300
23	310
24	320
25	330
26	340

Input Summary Listing (Original Fig. 12, first page)

```
CO
IM                        NAP,    A COMPUTER PROGRAM FOR THE COMPUTATION OF
    TNO"DIMENSIONAL, TIME-DEPENDENT,                     INVISCID NOZZLE FLOW

                                                   BV MICHAEL C, CLINE,        t-I - LOS
                                                       ALAMUS SCIENTIFIC LABORATORY



                  PROGRAM ABSTRACT -

                                          THE EQUATIONS OF MOTION FOR TMO-OIMEN3IONAL,
                                              TIME DEPENDENT, INVISCID FLOM IN A NOZZLE
                                    ARE SOLVED USING THE SECOND-ORDER, MACCORMACK,
                                        FINITE-DIFFERENCE SCHEME, THE FLUID IS ASSUMED
                                    TO BE A PERFECT GAS. ALL BOUNDARY CONDITIONS ARE
                                        COMPUTED USING A SECOND-ORDER, REFERENCE PLANE
                                    CHARACTERISTIC SCHEME, THE STEADY STATE SOLUTION IS
                                        OBTAINED AS THE ASYMPTOTIC SOLUTION FOR
                                    LARGE TIME, THE NUZZLES MAY BE EITHER CONVERGING,
                                        CONVERGING-DIVERGING, OH PLUG GEOMETRIES,


                  JOB TITLE -

                                    CASE NO. 3 - CONVERGING-DIVERGING, PLUG NOZZLE (10
                                        DEG CONE, PT/PE<3,29)

```

```
        CONTROL PARAMETERS -

                    L MAX*31    MMAX* 6     NMAX* 400      NPRINT*    0
                       TCONV*   ,005        FDT * 1,60      NST AG*0
                       NASH*1         IUNIT*0
                    IUI = 1     IUO* 1      lEXsl           NCONVI* 1
                       TST0P*1,00000      N1D* 1          NPLOT* -1
                       IPUNCH*0      ISUPER*0
                    IAV*0       CAV* 5,0    XMU*1, 00      XLA*0,00
                       RKMU*   ,50          CT A* ,50       LSS* 2       SMP*
                       ,95      NST*     0


        FLUID MODEL -

                    THE RATIO OF SPECIFIC HEATS, GAMMA <1,4000 AND THE
                       GAS CONSTANT, R                         53,3500
                       (FT-LBF/LBM-R)


        PLON GEOMETRY -

                    AXISYMMETRIC FLON HAS BEEN SPECIFIED


        NOZZLE GEOMETRY -

                    A CONSTANT AREA DUCT HAS BEEN SPECIFIED BY XI>
                       -4,4400 (IN), Rl<                   4,0000 (IN),
                       AND XE*      2,9600 (IN)

                    AN EXHAUST JET CALCULATION HAS BEEN REQUESTED,  THE
                       NOZZLE ENDS AT X*    ,7400 (IN),  THE
                    MESH POINTS L* 23 TO L* 31 ARE AN INITIAL
                       APPROXIMATION TO THE EXHAUST JET BOUNDARY.

                    A CIRCULAR-ARC, CONICAL CENTERBODY HAS BEEN SPECIFIED
                       BY XlCB* -4,4400 (IN), RICB*  1,3000 (IN),
                    RTCB* 3.3650 (IN), XECB*     2,9600 (IN), RCICB*
                       ,7500 (IN), RCTCB* 4,9500 (IN), ANGICB* 45,00
                       (DEG),
                    AND ANGECB* 10,00 (DEG). THE: COMPUTED VALUES ARE
                       XTCB* -.0143 (IN) AND RECB*  2,9170 (IN),


        BOUNDARY CONDITIONS -
```

```
                    PT* 100.0000 (PSIA)        TT*     10,0000 (F)
                        THETA*       0.0000 (DEG)         PE*   30,4000
                        (PSIA)
   N*   10
   N*   20
   N*   30
   N*   40
   Ns   50
   N*   60
   N*   70
   N*   00

                                                   Fig. 12   Case No.
                                                      3 output
```

Solution Surface Output Listing (Original Fig. 12, pages 38–41)

Note: OCR on pages 40–41 can read the surface number as "395" in the header, but the L column continues monotonically from 1 to 31 and the reported times are effectively unchanged; this block is treated as one continued solution-surface output. A structured OCR-normalized table is provided below using high-confidence parsed rows (136 of 186 expected rows). Low-confidence rows remain in raw text extraction files under 'docs/conversion/fig12/'.

Table 2.2: Case No. 3 solution-surface output (Fig. 12) converted to structured table from OCR (high-confidence rows).

L	M	X (in)	Y (in)	U (fps)	V (fps)	P (psia)	RHO	Q (fps)	Mach	T (F)
1	1	-4.4404	1.3000	109.4812	-0.0000	99.3430	0.506883	109.4812	0.0971	69.0027
1	2	-4.4400	1.8400	106.0169	-0.0000	99.3831	0.507029	106.0169	0.0940	69.0638
1	3	-4.4400	2.3830	212.9815	-0.0000	97.5259	0.500243	212.9815	0.1894	66.2199
1	4	-4.4400	2.9200	236.6861	-0.0000	96.9532	0.498143	236.6861	0.2107	65.3351
1	5	-4.4400	3.4600	276.8857	-0.0000	95.8470	0.494076	276.8857	0.2468	63.6156
1	6	-4.4400	4.0000	273.4428	-0.0000	95.9570	0.494481	273.4428	0.2437	63.7873
2	1	-4.1933	1.3417	107.4307	37.4142	99.2108	0.506393	113.7593	0.1009	68.8098
2	2	-4.1933	1.8734	104.0377	29.6151	99.2062	0.506362	108.1707	0.0960	68.8175
2	5	-4.1933	2.4350	214.2242	22.0820	97.4455	0.499962	215.3593	0.1915	66.0820
2	6	-4.1933	4.0000	274.0878	0.0000	95.9535	0.494468	274.0878	0.2443	63.7824
3	4	-3.9467	2.9940	245.1004	28.8405	96.7471	0.497390	246.7914	0.2197	65.0115
3	5	-3.9467	3.4970	284.0488	15.6501	95.6688	0.493423	284.4796	0.2537	63.3345
3	6	-3.9467	4.0000	279.0134	0.0000	95.8206	0.493979	279.0134	0.2487	63.5740
4	3	-3.7000	2.6376	251.8880	69.9268	96.4944	0.496505	261.4141	0.2328	64.5736
4	4	-3.7000	3.0917	258.5497	37.0252	96.2556	0.495578	261.1873	0.2327	64.2546

L	M	X (in)	Y (in)	U (fps)	V (fps)	P (psia)	RHO	Q (fps)	Mach	T (F)
4	5	-3.7000	3.5459	293.1525	20.9197	95.3831	0.492369	293.8980	0.2622	62.8874
4	6	-3.7000	4.0000	287.4930	0.0000	95.5579	0.493011	287.4930	0.2564	63.1645
5	1	-3.4533	1.9750	165.0872	159.4938	97.2086	0.499081	229.5475	0.2042	65.7285
5	3	-3.4533	2.7850	274.0965	86.3530	95.8902	0.494289	287.3773	0.2562	63.6267
5	4	-3.4533	3.1900	275.5988	45.2297	95.7240	0.493625	279.2855	0.2490	63.4220
5	5	-3.4533	3.5950	305.6486	25.4583	95.0148	0.491013	306.7070	0.2738	62.3076
5	6	-3.4533	4.0000	300.2397	0.0000	95.1903	0.491657	300.2397	0.2679	62.5870
6	1	-3.2067	2.1978	205.2765	173.2552	96.0735	0.494909	268.6183	0.2394	63.9703
6	2	-3.2067	2.5582	229.6718	104.0684	96.2531	0.495595	252.1495	0.2247	64.2229
6	3	-3.2067	2.9187	297.7834	94.9364	95.0763	0.491294	312.5506	0.2790	62.3581
6	4	-3.2067	3.2791	296.9780	50.8083	94.9696	0.490842	301.2929	0.2689	62.2412
6	5	-3.2067	3.6396	321.7514	27.9020	94.4542	0.488941	322.9590	0.2885	61.4256
6	6	-3.2067	4.0000	317.4202	0.0000	94.6142	0.489529	317.4202	0.2835	61.6822
7	1	-2.9600	2.3929	246.1173	182.2747	94.9109	0.490626	306.2642	0.2734	62.1473
7	2	-2.9600	2.7143	267.6866	117.9731	95.0919	0.491347	292.5299	0.2611	62.3756
7	3	-2.9630	3.0357	323.2732	100.9577	94.2213	0.488127	338.6709	0.3027	61.0076
7	4	-2.9600	3.3571	321.5757	56.3837	94.1319	0.487751	326.4813	0.2918	60.9145
7	5	-2.9600	3.6736	341.9079	29.8608	93.7507	0.486340	343.2094	0.3069	60.3106
7	6	-2.9600	4.0000	338.8474	0.0000	93.8859	0.486837	338.8474	0.3030	60.5294
9	1	-2.4667	2.7146	335.9522	191.6407	91.9380	0.479603	386.7687	0.3469	57.4175
9	3	-2.4667	3.2288	387.1967	105.6010	91.7925	0.479105	401.3389	0.3600	57.1358
9	4	-2.4667	3.4859	384.1126	63.0205	91.7586	0.478944	389.2481	0.3492	57.1185
9	5	-2.4667	3.7429	397.2721	31.6727	91.5882	0.478304	398.5327	0.3576	56.8485
9	6	-2.4667	4.0000	396.0271	0.0000	91.6830	0.478654	396.0271	0.3553	57.0061
10	1	-2.2200	2.8463	385.7795	192.0533	90.0749	0.472645	430.9412	0.3876	54.3954
10	2	-2.2200	3.0770	396.9902	137.6498	90.3795	0.473682	420.1767	0.3778	54.7871
10	3	-2.2200	3.3078	426.5248	105.6109	90.1440	0.472945	439.4053	0.3952	54.4635
10	4	-2.2200	3.5385	422.6905	64.3804	90.1463	0.472923	427.5654	0.3845	54.5002
10	5	-2.2200	3.7693	433.1660	31.9255	90.0516	0.472562	434.3409	0.3907	54.3518
10	6	-2.2200	4.0000	432.3408	0.0000	90.1354	0.472872	432.3408	0.3888	54.4940
11	1	-1.9733	2.9607	440.1175	189.7053	87.8573	0.464311	479.2614	0.4326	50.7368
11	2	-1.9733	3.1686	448.0872	136.0339	88.2164	0.465769	466.8662	0.4230	51.2182
11	3	-1.9733	3.3764	471.3597	104.1647	88.1287	0.465373	482.7320	0.4356	51.1456
11	4	-1.9733	3.5843	466.6714	64.4745	88.1745	0.465519	471.1042	0.4250	51.2510
11	5	-1.9733	3.7921	474.8778	31.7414	88.1431	0.465393	475.9374	0.4294	51.2060
11	6	-1.9733	4.0000	474.2174	0.0000	88.2221	0.465686	474.2174	0.4278	51.3423
12	1	-1.7267	3.0593	500.0783	184.4158	85.1723	0.454142	532.9986	0.4833	46.2146
12	2	-1.7267	3.2474	504.7398	135.5893	85.6008	0.455882	522.6344	0.4736	46.8202

L	M	X (in)	Y (in)	U (fps)	V (fps)	P (psia)	RHO	Q (fps)	Mach	T (F)
12	3	-1.7267	3.4356	522.3541	101.1329	85.6519	0.456001	532.0542	0.4820	46.9892
12	4	-1.7267	3.6237	516.6715	63.2783	85.7507	0.456353	520.5320	0.4715	47.1831
12	6	-1.7267	4.0000	522.1891	0.0000	85.8554	0.456738	522.1891	0.4729	47.3748
13	1	-1.4800	3.1429	566.2529	175.5815	61.9333	0.441756	592.8501	0.5405	40.6179
13	2	-1.4800	3.3143	567.4408	130.0717	82.4540	0.443870	582.1578	0.5304	41.4001
13	3	-1.4800	3.4858	579.7939	96.2310	82.6443	0.444521	587.7255	0.5352	41.8215
13	4	-1.4800	3.6572	572.9430	60.6967	82.8076	0.445125	576.1491	0.5245	42.1299
13	5	-1.4800	3.8286	577.3073	29.7520	82.8869	0.445422	578.0735	0.5262	42.2757
13	6	-1.4800	4.0000	576.4726	0.0000	82.9705	0.445737	576.4726	0.5246	42.4267
14	1	-1.2333	3.2125	639.0676	162.4290	78.0504	0.426727	659.3865	0.6054	33.6883
14	2	-1.2333	3.3700	636.5520	121.0465	78.6926	0.429342	647.9589	0.5943	34.7195
14	3	-1.2333	3.5275	643.9203	89.0877	79.0292	0.430567	650.0539	0.5958	35.4213
14	4	-1.2333	3.6850	635.7200	56.5499	79.2706	0.431485	638.2302	0.5847	35.8779
14	5	-1.2333	3.8425	638.3532	27.7308	79.4040	0.431995	638.9553	0.5852	36.1254
14	6	-1.2333	4.0000	637.2739	0.0000	79.4949	0.432340	637.2739	0.5835	36.2970
15	1	-0.9867	3.2685	718.4278	143.9795	73.4700	0.408725	732.7132	0.6786	25.1849
15	2	-0.9867	3.4148	711.9765	107.8905	74.2675	0.411994	720.1048	0.6660	26.5599
15	3	-0.9867	3.5611	714.5835	79.2171	74.7626	0.413865	718.9610	0.6642	27.5894
15	4	-0.9867	3.7074	704.8565	50.5751	75.0970	0.415164	706.6686	0.6524	28.2372
15	5	-0.9867	3.8537	705.8355	24.8312	75.2871	0.415905	706.2721	0.6518	28.6006
15	6	-0.9867	4.0000	704.4402	0.0000	75.3868	0.416293	704.4402	0.6500	28.7919
16	1	-0.7400	3.3115	803.7732	119.1704	68.1821	0.387547	812.5607	0.7607	14.8693
16	2	-0.7400	3.4492	793.2104	89.8965	69.1705	0.391649	798.2882	0.7456	16.7076
16	4	-0.7400	3.7246	779.9013	42.4700	70.2601	0.396011	781.0568	0.7280	19.0199
16	5	-0.7400	3.8623	779.2786	20.8999	70.5298	0.397005	779.5588	0.7262	19.5177
17	2	-0.4933	3.4734	879.1927	66.3566	63.4636	0.368344	881.6933	0.8340	5.0490
17	3	-0.4933	3.6050	873.0032	49.2059	64.3169	0.371780	874.3888	0.8254	6.9467
17	4	-0.4933	3.7367	859.9066	31.9340	64.8781	0.374068	860.4994	0.8113	8.1395
17	5	-0.4933	3.6683	57.7794	15.7672	65.1907	0.375342	657.9243	0.8083	8.7992
17	6	-0.4933	4.0000	855.6361	0.0000	65.3116	0.375837	855.6361	0.8059	9.0498
18	2	-0.2467	3.4876	968.3735	36.6601	57.2806	0.342422	969.0721	0.9303	-8.4827
18	3	-0.2407	3.6157	958.2832	28.1325	58.3275	0.346782	958.6961	0.9179	-6.0113
18	6	-0.2467	4.0000	937.6150	0.0000	59.5321	0.351811	937.6150	0.8950	-3.2594
19	2	0.0000	3.0920	1056.8199	0.3966	50.8097	0.310533	1058.8195	1.0300	-23.6353
19	3	0.0000	3.6190	1005.3991	2.5186	52.0880	0.319889	1005.3522	1.0172	-20.0885
19	4	0.0000	3.7960	1029.0988	2.6307	52.9016	0.323016	1029.0521	0.9990	-18.0950
19	5	0.0000	3.8730	1020.3133	1.0601	53.3012	0.325310	1020.3103	0.9932	-17.0236
21	1	0.9933	3.3389	1269.0032	130.7062	35.7235	0.200355	1275.7208	1.3101	-65.3967

L	M	X (in)	Y (in)	U (fps)	V (fps)	P (psia)	RHO	Q (fps)	Mach	T (F)
21	2	0.9933	3.9711	1236.3910	-93.2116	37.9196	0.255030	1201.8900	1.2606	-58.6706
21	3	0.9933	3.6030	1219.8855	-62.9528	39.5789	0.202656	1221.5088	1.2360	-53.5867
22	2	0.7900	3.9058	1305.5170	155.2005	32.8860	0.230385	1310.7103	1.3663	-70.7132
22	3	0.7900	3.5893	1208.8579	111.5273	30.0362	0.238202	1293.6703	1.3360	-69.7912
22	4	0.7900	3.7229	1272.2609	-71.9720	35.0028	0.203108	1270.2950	1.3113	-66.9980
22	5	0.7900	3.6619	1267.2158	-32.7531	35.0110	0.205260	1267.6390	1.3026	-65.8919
23	1	0.9667	3.2609	1383.2720	203.9082	27.2665	0.201293	1000.6112	1.0980	-90.3500
23	2	0.9667	3.0123	1370.5310	187.5136	28.2071	0.206581	1387.2628	1.0731	-90.9272
23	3	0.9867	3.5596	1370.0887	128.3918	28.8860	0.209959	1380.0700	1.0609	-86.6062
23	4	0.9667	3.7070	1372.0803	-67.7226	29.0913	0.210905	1373.7506	1.0520	-87.6901
23	5	0.9867	3.8503	1375.5103	-12.7883	29.2328	0.211388	1375.5738	1.0520	-86.7353
25	1	1.9600	3.1779	1001.9167	261.3019	21.3229	0.168627	1500.7777	1.6616	-116.6920
25	3	1.9600	3.0939	1070.3210	165.8729	22.8291	0.177072	1083.6231	1.6220	-112.0096
25	4	1.9600	3.6519	1033.9630	109.1605	25.1620	0.189852	1001.7000	1.5509	-102.2670
25	5	1.9800	3.8098	1392.5720	137.2705	27.8310	0.200008	1399.3220	1.0877	-91.8057
25	6	1.0800	3.9678	1309.1683	120.2698	30.9025	0.217206	1350.8790	1.0221	-82.2663
26	1	1.7267	3.1309	1509.8281	266.2235	20.0212	0.161080	1533.1197	1.7096	-125.3505
26	2	1.7267	3.2992	1099.0577	251.5096	20.6565	0.165113	1520.0007	1.6878	-122.3211
26	4	1.7267	3.6137	1017.7007	227.3063	25.5073	0.191703	1035.8071	1.5057	-100.9358
26	5	1.7267	3.7735	1380.0098	207.8969	27.9910	0.200890	1395.9772	1.0830	-91.2585
26	6	1.7267	3.9333	1302.5008	187.8090	30.0071	0.217269	1355.5830	1.0228	-82.2083
27	1	1.9733	3.0909	1059.0269	257.2658	22.5791	0.175550	1081.5308	1.6221	-112.8006
27	3	1.9733	3.0101	1021.8776	293.0628	20.3002	0.185300	1051.7650	1.5730	-105.5272
27	5	1.9733	3.7292	1365.8397	260.7981	28.2135	0.206032	1390.5157	1.0750	-90.3801
28	1	2.2200	3.0975	1910.7995	-298.7612	25.9098	0.192169	1932.5583	1.5969	-103.1005
28	2	2.2200	3.2097	1906.2072	-302.8893	25.0096	0.189801	1938.9577	1.5561	-109.9101
28	3	2.2200	3.3620	1391.8996	-333.8577	25.5297	0.192227	1931.3259	1.5921	-101.5290
28	4	2.2200	3.5193	1370.9298	-339.0730	26.7231	0.198370	1910.5562	1.5090	-96.3879
28	6	2.2200	3.8339	1317.7933	-293.3750	30.3985	0.217221	1350.0599	1.9170	-82.2726
29	1	1.2333	3.2219	1030.0370	252.8596	20.1025	0.180583	1056.1601	1.5809	106.9657
29	2	1.2333	3.3752	1036.0195	196.1252	20.5272	0.186710	1009.7069	1.5706	-105.0328
29	3	1.2333	3.5291	1002.6305	135.8278	20.8330	0.188222	1009.0107	1.5660	-103.8817
29	4	1.2333	3.6829	1029.7078	-69.7732	25.6828	0.192670	1032.5235	1.5006	-100.2039
29	5	1.2333	3.8367	1397.0981	-69.0626	27.8136	0.203959	1399.2230	1.0876	-91.9200
29	6	1.2333	3.9905	1350.6985	-61.0675	30.3938	0.217202	1356.0702	1.0230	-82.2982
30	1	2.7131	2.9605	1199.3369	-210.5938	90.0065	0.269795	1212.7615	1.2251	-52.1976
30	2	2.7133	3.1101	1210.9119	-238.5102	38.6329	0.258299	1239.1772	1.2530	-56.2885
30	3	2.7133	3.2598	1290.8253	-278.9910	36.0291	0.295565	1271.6627	1.3037	-69.0377

L	M	X (in)	Y (in)	U (fps)	V (fps)	P (psia)	RHO	Q (fps)	Mach	T (F)
30	5	2.7133	3.5591	1293.9769	-339.6986	31.3972	0.222919	1336.5995	1.3968	-78.9811
30	6	2.7133	1.7088	1306.3885	-325.0857	30.3957	0.217212	1396.2286	1.9100	0.0000
31	2	2.9600	3.0631	1118.7287	-193.6976	95.5797	0.291692	1135.3698	1.1278	-38.2301
31	3	2.9600	3.2092	1167.8982	-221.2927	91.6999	0.273125	1188.6785	1.1953	-98.9999
31	4	2.9600	3.3552	1221.3161	-267.3618	36.7919	0.299537	1250.2381	1.2785	-62.0356
31	5	2.9600	3.5013	1278.0603	-305.8819	32.6871	0.228973	1319.1593	1.3657	-79.6800
31	6	2.9600	3.6979	1297.5788	-322.8935	30.3963	0.217213	1337.1503	1.9035	-82.2872

Note: This chapter was reconstructed from OCR-extracted text and program code analysis. For production use, consult the original LASL technical documentation and verify input/output specifications with the actual Fortran source code listings in Appendix C.

References

1. L. M. Saunders, "Numerical Solution of the Flow Field in the Throat Region of a Nozzle," Brown Engineering Co. report BSVD-P-66-TN-001 (NASA CR 82601), August 1966.

2. R. F. Cuffel, L. H. Back, and P. F. Massier, "Transonic Flow-Field in a Supersonic Nozzle with Small Throat Radius of Curvature," *AIAA J.* 7, 1364–1366, July 1969.

3. A. A. Migdal, E. J. Wuchina, and W. D. Boyd, "Method for Predicting the Nozzle Operating Line," NASA Lewis Research Center Report NASA CR-72547, April 1968.

4. S. Prozan, "Numerical Solution of Transonic Flow in a Nozzle," Grumman Aircraft Engineering Corporation Report, 1966.

5. C. W. Laval, "Calculation of Two-Dimensional Compressible Flow in Convergent-Divergent Channels," AEDC Report TR-69-120, August 1969.

6. P. Serra, "Numerical Simulation of the Transonic Flow in a Convergent-Divergent Nozzle," CNES Report, 1970.

7. C. A. Brown and H. Ozcan, "Numerical Solution of Steady Flow in a Nozzle," Douglas Aircraft Report, 1969.

8. R. W. MacCormack, "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, 1969.

9. G. Moretti and M. Abbett, "A Time-Dependent Method for the Analysis of Transonic Flows," *AIAA J.* 4, 2136–2141, December 1966.

10. G. Moretti, "Transonic and Supersonic Flow Computations," *Advances in Applied Mathematics* 2, 1981.

11. J. von Neumann and R. D. Richtmyer, "A Method for the Numerical Calculation of Hydrodynamic Shocks," *J. Applied Phys.* 21, 232–237, March 1950.

12. *Ibid*, Reference 11.

13. M. C. Cline, "NAP: A Computer Program for the Computation of Two-Dimensional, Time-Dependent, Inviscid Nozzle Flow," Los Alamos Scientific Laboratory Report, 1977.

14. R. E. Loh and R. L. Loh, “A Comparison of Computer Execution Times,” TRW Technical Report, 1975.

15. IBM Corporation, “System/360 Performance Characteristics,” IBM Technical Manual, 1971.

16. R. H. Prozan and D. E. Kooker, “Numerical Analysis of Steady Nozzle Flow Using Relaxation Methods,” AIAA Paper, 1971.

17. T. P. Thornock, “Experimental Study of Converging Nozzle Flow,” NASA Technical Memorandum, June 1968.

18. D. L. Bresnahan and A. L. Johns, “Experimental Flow in a Plug Nozzle,” NASA Contractor Report, 1972.

Appendix A

Characteristic Relations

A.1 $\eta =$ constant Reference Plane

[To be completed: Derivation of characteristic relations for inlet and exit boundaries (Appendix A from original)]

A.2 $\zeta =$ constant Reference Plane

[To be completed: Derivation of characteristic relations for wall and centerbody boundaries (Appendix B from original)]

Appendix B

Fortran Code Listing (LASL Identification: LP-0537)

B.1 Main Program (fortran_main.f)

This is the main program that orchestrates the NAP solver. It handles input/output, initialization, and time-stepping control. *OCR note: this listing received a conservative normalization pass for obvious token errors; unresolved ambiguities are tracked in conversion notes.*

```
      PROGRAM MAIN(INPUT,OUTPUT,FILM,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
     1TAPE7=FILM)
C
C     ****************************************************************
C     NAP, A COMPUTER PROGRAM FOR THE COMPUTATION OF TWO-DIMENSIONAL,
C                    TIME-DEPENDENT, INVISIC NOZZLE FLOW
C
C                          BY MICHAEL C. CLINE, T-3
C                     LOS ALAMOS SCIENTIFIC LABORATORY
C
C     ****************************************************************
C
C                               PROGRAM ABSTRACT
C
C     THE EOUATIONS OF MOTION FOR TWO-DIMENSIONAL, TIME DEPENDENT,
C     INVISCID FLOW IN A NOZZLE ARE SOLVED USING THE SECOND-ORDER,
C     MACCORMACK, FINITE-DIFFERENCE SCHEME, THE FLUID IS ASSUMED TO BE
C     A PERFECT GAS, ALL BOUNDARY CONDITIONS ARE COMPUTED USING A
C     SECOND-ORDER, REFERENCE PLANE CHARACTERISTIC SCHEME, THE STEADY
C     STATE SOLUTION IS OBTAINED AS THE ASYMPTOTIC SOLUTION FOR LARGE
C     TIME, THE NOZZLES MAY BE EITHER CONVERGING, CONVERGING-DIVERGING,
C     OR PLUG GEOMETRIES.
C
      DIMENSION TITLE(8), UI(21), VI(21), PI(21), ROI(21)
      COMMON /AV/ IAV,CAV,NST,SMP,LSS,CTA,XMU,XLA,RKMU,QUT(81,21),QVT(81
     1,21),QPT(81,21)
      COMMON /ONESID/ UD(4),VD(4),PD(4),ROD(4)
      COMMON /SOLUTN/ U(8l,21,2),V(8l,21,2),P(8l,21,2),RO(81,2l,2)
      COMMON /CNTRLC/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,GAMMA,RGAS,GAM1,GA
     1M2,L1,L2,L3,Ml,M2,DX,DY,DT,N,N1,N3,NASM,IVEL,ICHAR,N1D,LJET,JFLAG,
     2IERR,IUI,IUO,DXR,DYR,LD,MD,LMD1,LMD3,IB,RSTAR,RSTARS,NPLOT,G,PC,TC
     3,LC,PLOW,ROLOW
      COMMON /GEMTRYC/ NGEOM,XI,RI,XT,RT,XE,RE,RCI,RCT,ANGI,ANGE,XW(8l),
     1YW(8l),XWI(8l),YWI(8l),NXNY(8l),NWPTS,IINT,IDIF,LT,NDIM
      COMMON /GCB/ NGCB,XICB,RICB,XTCB,RTCB,XECB,RECB,RCICB,RCTCB,ANGICB
     1,ANGECB,XCB(8l),YCB(8l),XCBI(81),YCBI(8l),NXNYCB(8l),NCBPTS,IINTCB
     2,IDIFCB,LECB
      COMMON /BCC/ PT(21),TT(21),THETA(21),PE,MASSE,MASSI,MASST,THRUST,N
     3STAG
```

```
      REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE
      NAMELIST /CNTRL/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,TSTOP,GAMMA,RGAS,
     1NASM,NAME,NCONVI,NST,IUI,IUO,SMP,IPUNCH,IAV,CAV,NPLOT,IEX,LSS,CTA,
     2XMU,XLA,RKMU,IUNIT,PLOW,ROLOW
      NAMELIST /IVS/ U,V,P,RO,N1D,NSTART,TSTART,RSTAR,RSTARS
      NAMELIST /GEMTRY/ NDIM,XI,RI,RT,XE,RCI,RCT,ANGI,ANGE,NGEOM,XWI,YWI
     1,NWPTS,IINT,IDIF,LJET,JFLAG,NXNY,YW
      NAMELIST /GCBL/ NGCB,RICB,RTCB,RCICB,RCTCB,ANGICB,ANGECB,YCB,NXNYC
     1B,XCBI,YCBI,NCBPTS,IINTCB,IDIFCB
      NAMELIST /BC/ PT,TT,THETA,PE,NSTAG,ISUPER,UI,VI,PI,ROI
C
C          READ IN DATA
C
 10   TCONV=0.0 $ FDT=1.0 $ TSTOP=1.0 $ NASM=1 $ NSTAG=0 $ NAME=0
      IPUNCH=0 $ NGCB=0 $ IINTCB=1 $ IDIFCB=1 $ NSTART=0 $ TSTART=0.0
      IINT=1 $ IDIF=1 $ NMAX=0 $ NPRINT=0 $ GAMMA=1.4 $ RGAS=53.35
      N1D=1 $ NDIM=1 $ THETA(1)=0.0 $ PE=14.7 $ NST=0 $ N=0 $ IEX=1
      NCONVI=1 $ IERR=0 $ JFLAG=0 $ IUI=1 $ IUO=1 $ SMP=0.95 $ ISUPER=0
      IAV=0 $ CAV=4.0 $ NPLOT=-1 $ G=32.174 $ PC=144.0 $ TC=460.0
      LC=12.0 $ IUNIT=0 $ LSS=2 $ CTA=0.5 $ XMU=0.2 $ XLA=1.0
      RKMU=0.7 $ PLOW=0.01 $ ROLOW=0.0001 $ RSTAR=0.0 $ RSTARS=0.0
      READ 650, TITLE
      IF (EOF,5) 20,30
 20   STOP
 30   READ (5,CNTRL)
      READ (5,IVS)
      READ (5,GEMTRY)
      READ (5,GCBL)
      READ (5,BC)
      IF (NAME.EQ.0) GO TO 40
      WRITE (6,CNTRL)
      WRITE (6,IVS)
      WRITE (6,GEMTRY)
      WRITE (6,GCBL)
      WRITE (6, BC)
C
C     PRINT INPUT DATA
C
 40   PRINT 660
      PRINT 690
      PRINT 680
      PRINT 700
      PRINT 670
      PRINT 710, TITLE
      PRINT 670
      PRINT 720
      NPRIND=ABS(FLOAT(NPRINT))
      PRINT 730,
          LMAX,MMAX,NMAX,NPRIND,TCONV,FDT,NSTAG,NASM,IUNIT,IUI,IUO,IEX,NCONVI,TSTOP,N1D,NPLOT,IPUNCH,ISUPER,IAV,CAV,XMU,XLA,RKMU,CTA,LSS,SMP,I
      PRINT 670
      IF (IUI.EQ.1) PRINT 740, GAMMA,RGAS
      IF (IUI.EQ.2) PRINT 750, GAMMA,RGAS
      PRINT 670
      PRINT 780
      IF (NDIM.EQ.0) PRINT 790
      IF (NDIM.EQ.1) PRINT 800
C
C     CALCULATE THE NOZZLE RADIUS AND NORMAL
C
            PRINT 670
            CALL GEOM
            IF (IERR.NE.0) GO TO 10
            DY=1.0/FLOAT(MMAX-1)
            IF (NGCB.NE.0) GO TO 60
            RICB=0.0
            RTCB=0.0
            DO 50 L=1,LMAX
            YCB(L)=0.0
            NXNYCB(L)=0.0
 50   CONTINUE
      GO TO 90
 60   XICB=XI
      XECB=XE
      CALL GEOMCB
      LT=1 $ XI=XICB $ XE=XECB
      Y0=0.0
```

```
      DO 80 L=1,LMAX
      IF (NDIM.EQ.0) Y=YW(L)-YCB(L)
      IF (NDIM.EQ.1) Y=YW(L)**2-YCB(L)**2
      IF (Y.GT.0.0) GO TO 70
      PRINT 920
      GO TO 10
70    IF (Y.LT.Y0) LT=L
      Y0=Y
80    CONTINUE
90    IF (NSTAG.NE.0) GO TO 110
      DO 100 M=2,MMAX
      PT(M)=PT(1)
      TT(M)=TT(1)
      THETA(M)=THETA(1)
100   CONTINUE
      PRINT 670
      IF (IUI.EQ.1) PRINT 760, PT(1),TT(1),THETA(1),PE
      IF (IUI.EQ.2) PRINT 770, PT(1),TT(1),THETA(1),PE
      GO TO 130
110   PRINT 660
      IF (IUI.EQ.1) PRINT 890, PE
      IF (IUI.EQ.2) PRINT 770, PE
      DO 120 M=1,MMAX
      PRINT 910, M,PT(M),TT(M),THETA(M)
120   CONTINUE
C
C     CONVERT METRIC UNITS TO ENGLISH UNITS
C
130   IF (IUI.EQ.1) GO TO 180
      RSTAR=RSTAR/2.54
      RSTARS=RSTARS/6.4516
      RGAS=RGAS/5.38032
      DO 140 M=1,MMAX
      PT(M)=PT(M)/6.8948
      TT(M)=(TT(M)+40.0)*9.0/5.0-40.0
140   CONTINUE
      PE=PE/6.8948
      IF (ISUPER.EQ.0) GO TO 160
      DO 150 M=1,MMAX
      UI(M)=UI(M)/0.3048
      VI(M)=VI(M)/0.3048
      PI(M)=PI(M)/6.8948
      ROI(M)=ROI(M)/16.02
150   CONTINUE
160   IF (N1D.NE.0) GO TO 180
      IF (NSTART.NE.0) GO TO 180
      DO 170 L=1,LMAX
      DO 170 M=1,MMAX
      U(L,M,1)=U(L,M,1)/0.3048
      V(L,M,1)=V(L,M,1)/0.3048
      P(L,M,1)=P(L,M,1)/6.8948
      RO(L,M,1)=RO(L,M,1)/16.02
170   CONTINUE
C
C     CONVERT INPUT DATA UNITS TO INTERNAL UNITS
C
180   IF (IUNIT.EQ.0) GO TO 190
      PC=1.0 $ LC=1.0 $ G=1.0
      TC=0.0
190   TCONV=TCONV/100.0
      T=TSTART*LC
      TSTOP=TSTOP*LC
      DO 200 L=1,LMAX
      XWI(L)=0.0
200   CONTINUE
      DO 210 M=1,MMAX
      PT(M)=PT(M)*PC
      TT(M)=TT(M)+TC
      THETA(M)=THETA(M)*0.0174533
210   CONTINUE
      PE=PE*PC
      IF (N1D.NE.0) GO TO 230
      DO 220 L=1,LMAX
      DO 220 M=1,MMAX
      P(L,M,1)=P(L,M,1)*PC
      RO(L,M,1)=RO(L,M,1)/G
```

```
220 CONTINUE
230 GAM1=GAMMA/(GAMMA-1.0)
    GAM2=(GAMMA-1.0)/2.0
    IF (ISUPER.EQ.0) GO TO 250
    DO 240 M=1,MMAX
    U(1,M,1)=UI(M)
    V(1,M,1)=VI(M)
    P(1,M,1)=PI(M)*PC
    RO(1,M,1)=ROI(M)/G
    U(1,M,2)=U(1,M,1)
    V(1,M,2)=V(1,M,1)
    P(1,M,2)=P(1,M,1)
    RO(1,M,2)=RO(1,M,1)
240 CONTINUE
250 L1=LMAX-1
    L2=LMAX-2
    L3=LMAX-3
    M1=MMAX-1
    M2=MMAX-1
    IF (N1D.EQ.0) GO TO 260
C
C   COMPUTE THE 1-D INITIAL-DATA SURFACE
C
    CALL ONEDIM
    IF (IERR.NE.0) GO TO 10
C
C   COMPUTE THE INITIAL-DATA SURFACE MASS FLOW AND THRUST
C
260 IF (NPRINT.GT.0) GO TO 270
    NPRINT=-NPRINT
    GO TO 340
270 CALL MASFLO (0)
C
C   CALCULATE AND PRINT THE INITIAL-VALUE SURFACE
C
    DO 330 IU=1,2
    IF (IUO.EQ.1.AND.IU.EQ.2) GO TO 330
    IF (IUO.EQ.2.AND.IU.EQ.1) GO TO 330
    NLINE=0
    PRINT 660
    PRINT 810, TSTART,NSTART
    PRINT 820
    IF (IU.EQ.1) PRINT 830
    IF (IU.EQ.2) PRINT 840
    PRINT 670
    X=XI-DX
    DO 300 L=1,LMAX
    X=X+DX
    CALL MAP (0,L,1,AL,BE,DE,LD1,AL1,BE1,DE1)
    DYIO=DY/BE
    Y=YCB(L)-DYIO
    DO 300 M=1,MMAX
    Y=Y+DYIO
    VELMAG=SQRT(U(L,M,1)**2+V(L,M,1)**2)
    XMACH=VELMAG/SQRT(GAMMA*P(L, M,1)/RO(L,M,1))
    PRES=P(L,M,1)/PC
    RHO=RO(L,M,1)/G
    TEMP=P(L,M,1)/RHO/RGAS-TC
    XP=X
    YP=Y
    UP=U(L,M,1)
    VP=V(L,M,1)
    IF (IU.EQ.1) GO TO 280
    XP=XP*2.54
    YP=YP*2.54
    UP=UP*0.3048
    VP=VP*0.3048
    PRES=PRES*6.8948
    RHO=RHO*16.02
    VELMAG=VELMAG*0.3048
    TEMP=(TEMP+40.0)*5.0/9.0-40.0
280 NLINE=NLINE+1
    IF (NLINE.LT.55) GO TO 290
    PRINT 660
    PRINT 810, TSTART,NSTART
    PRINT 820
```

```
     IF (IU,EQ,1) PRINT 830
     IF (IU,EQ,2) PRINT 840
     PRINT 670
     NLINE=1
290  PRINT 850, L,M,XP,YP,UP,VP,PRES,RHO,VELMAG,XMACH,TEMP
300  CONTINUE
     IF (IU,EQ,2) GO TO 310
     PRINT 870, MASST,THRUST,MASSI,MASSE
     GO TO 320
310  MASST=MASST*0.4536
     MASSI=MASSI*0.4536
     MASSE=MASSE*0.4536
     THRUST=THRUST*4.4477
     PRINT 880, MASST,THRUST,MASSI,MASSE
320  IF (IUO,NE,3) GO TO 340
330  CONTINUE
340  IF (NPLOT,LE,0) GO TO 350
     CALL PLOT (TITLE,TSTART,NSTART)
     PRINT 1030, NSTART
350  IF (NMAX,EQ,0) GO TO 10
C
C    INITIALIZE THE TIME STFP INTEGRATION LOOP PARAMETERS
C
     N1=1 $ N3=2 $ DQM=0.0 $ NS=0 $ NCONV=0 $ NC=0 $ LDUM=1 $ NPC=0
     DXR=1.0/DX $ DYR=1.0/DY $ DXRS=DXR*DXR $ DYRS=DYR*DYR
     LD=81 $ MD=21 $ LMD=LD*MD
     IF (NASM,NE,0,AND,LT,NE,1) LDUM=LT-1
     NPD=0
     IF (JFLAG,EQ,0) GO TO 360
     UD(1)=U(LJET-1,MMAX,N1)
     VD(1)=V(LJET-1,MMAX,N1)
     PD(1)=P(LJET-1,MMAX,N1)
     ROD(1)=RO(LJET-1,MMAX,N1)
     UD(2)=UD(1)
     VD(2)=VD(1)
     PD(2)=PD(1)
     ROD(2)=ROD(1)
C
C    ENTER THE TIME STEP INTERGRATION LOOP
C
360  DO 580 N=1,NMAX
     NDP=NPD+1
     IF (NPD,NE,10) GO TO 370
     NP=N+NSTART
     PRINT 1040, NP
     NPD=0
370  CONTINUE
     LMD1=LMD*(N1-1)
     LMD3=LMD*(N3-1)
C
C    CALCULATE DELTA T
C
     DO 380 L=1,LMAX
          CALL MAP (0,L,MD,AL,BE,DE,LD1,AL1,BE1,DE1)
          DXDY=DXRS+BE*BE*DYRS
          DO 380 M=1,MMAX
          LMN1=L+LD*(M-1)+LMD1
          QS=U(LMN1)*U(LMN1)+V(LMN1)*V(LMN1)
          AS=GAMMA*P(LMN1)/RO(LMN1)
          UPA=SQRT(QS*DXDY)+SQRT(AS*DXDY)
          IF (L,EQ,1,AND,M,EQ,1) UPAM=UPA
          IF (UPA,GT,UPAM) UPAM=UPA
380       CONTINUE
          DT=FDT/UPAM
          T=T+DT
          IF (T,LE,TSTOP) GO TO 390
          T=T-DT
          DT=TSTOP-T
          T=TSTOP
C
C         DETERMINE IF THE EXIT FLOW IS SUBSONIC OR SUPERSONIC
C
390       IVEL=0
          IF (QS,GE,AS) IVEL=1
C
c         CALCULATE THE NOZZLE WALL AND INTERIOR MESH POINTS
```

```
C
      IF (IAV,NE,0) CALL SHOCK (1)
      ICHAR=1
      IB=1
      CALL INTER
      CALL WALL
      IF (IERR,NE,0) GO TO 10
      IF (NGCB,EQ,0) GO TO 400
      IB=2
      CALL WALL
      IF (IERR,NE,0) GO TO 10
400   ICHAR=2
      IB=1
      CALL INTER
      CALL WALL
      IF (IERR,NE,0) GO TO 10
      IF (NGCB,EQ,0) GO TO 410
      IB=2
      CALL WALL
      IF (IERR,NE,0) GO TO 10
C
C     EXTRAPOLATE THE EXIT MESH POINTS FOR SUPERSONIC FLOM
C
410   DO 420 M=1,MMAX
      U(LMAX,M,N3)=U(L1,M,N3)+IEX*(U(L1,M,N3)-U(L2,M,N3))
      V(LMAX,M,N3)=V(L1,M,N3)+IEX*(V(L1,M,N3)-V(L2,M,N3))
      P(LMAX,M,N3)=P(L1,M,N3)+IEX*(P(L1,M,N3)-P(L2,M,N3))
      RO(LMAX,M,N3)=RO(L1,M,N3)+IEX*(RO(L1,M,N3)-RO(L2,M,N3))
      IF (P(LMAX,M,N3),GT,0.0,AND,RO(LMAX,M,N3),GT,0.0) GO TO 420
      P(LMAX,M,N3)=P(L1,M,N3)
      RO(LMAX,M,N3)=RO(L1,M,N3)
420   CONTINUE
      V(LMAX,MMAX,N3)=-U(LMAX,MMAX,N3)*NXNY(LMAX)
      V(LMAX,1,N3)=-U(LMAX,1,N3)*NXNYCB(LMAX)
C
C     CALCULATE THE NOZZLE INLET MESH POINTS
C
      IF (ISUPER,EQ,0) CALL INLET
C
C     CALCULATE THE NOZZLE EXIT MESH POINTS FOR SUBSONIC FLOW
C
      IF (IVEL,EQ,0) CALL EXITT
      IF (N,LE,NST) CALL SHOCK (2)
C
C     DETERMINE THE MAXIMUM (DELTA U)/U
C
      IF (TCONV,LE,0.0) GO TO 440
      DDQM=0.0
      DO 430 L=LDUM,LMAX
      DO 430 M=1,MMAX
      IF (U(L,M,N1),EQ,0.0) GO TO 430
      DQ=ABS((U(L,M,N3)-U(L,M,N1))/U(L,M,N1))
      IF (DQ,GT,DQM) DQM=DQ
430   CONTINUE
440   NC=NC+1
      NPC=NPC+1
      IF (DQM,GE,TCONV) GO TO 450
      NCONV=NCONV+1
      IF (NCONV,EQ,1) NCHECK=N-1
      IF (NCONV,GE,NCONVI) NC=NPRINT
450   IF (N,EQ,NMAX) NC=NPRINT
      IF (N,GE,NCHECK+NCONVI) NCONV=0
      IF (T,EQ,TSTOP) NC=NPRINT
      IF (NC,EQ,NPRINT) GO TO 460
      IF (NPC,EQ,NPLOT) GO TO 550
      GO TO 570
C
C     COMPUTE THE SOLUTION SURFACE MASS FLOW AND THRUST
C
460   ICN=0
      IF (JFLAG,EQ,0) GO TO 470
      IF (LT,NE,LJET-1) GO TO 470
      UDUM=U(LT,MMAX,N3)
      RODUM=RO(LT,MMAX,N3)
      U(LT,MMAX,N3)=UD(3)
      RO(LT,MMAX,N3)=ROD(3)
```

```
      ICN=1
470   CALL MASFLO (1)
      IF (ICN,EQ,0) GO TO 480
      U(LT,MMAX,N3)=UDUM
      RO(LT,MMAX,N3)=RODUM
C
C     CALCULATE AND PRINT THE SOLUTION SURFACE
C
480   DO 540 IU=1,2
      IF (IUO,EQ,1,AND,IU,EQ,2) GO TO 540
      IF (IUO,EQ,2,AND,IU,EQ,1) GO TO 540
      NLINE=0
      PRINT 660
      TIME=T/LC
      DTIME=DT/LC
      NP=N+NSTART
      PRINT 860, NP,TIME,DTIME
      PRINT 820
      IF (IU,EQ,1) PRINT 830
      IF (IU,EQ,2) PRINT 840
      PRINT 670
      X=XI-DX
      DO 510 L=1,LMAX
      X=X+DX
      CALL MAP (0,L,1,AL,BE,DE,LD1,AL1,BE1,DE1)
      DYIO=DY/BE
      Y=YCB(L)-DYIO
      DO 510 M=1,MMAX
      Y=Y+DYIO
      VELMAG=SQRT(U(L,M,N3)**2+V(L,M,N3)**2)
      XMACH=VELMAG/SQRT(GAMMA*P(L,M,N3)/RO(L,M,N3))
      PRES=P(L,M,N3)/PC
      RHO=RO(L,M,N3)/G
      TEMP=P(L,M,N3)/RHO/RGAS-TC
      XP=X
      YP=Y
      UP=U(L,M,N3)
      VP=V(L,M,N3)
      IF (IU,EQ,1) GO TO 490
      XP=XP*2.54
      YP=YP*2.54
      UP=UP*0.3048
      VP=VP*0.3048
      PRES=PRES*6.8948
      RHO=RHO*16.02
      VELMAG=VELMAG*0.3048
      TEMP=(TEMP+40.0)*5.0/9.0-40.0
490   NLINE=NLINE+1
      IF (NLINE,LT,55) GO TO 500
      PRINT 660
      PRINT 860, NP,TIME,DTIME
      PRINT 820
      IF (IU,EQ,1) PRINT 830
      IF (IU,EQ,2) PRINT 840
      PRINT 670
      NLINE=1
500   PRINT 850, L,M,XP,YP,UP,VP,PRES,RHO,VELMAG,XMACH,TEMP
510   CONTINUE
      IF (IU,EQ,2) GO TO 520
      PRINT 870, MASST,THRUST,MASSI,MASSE
      GO TO 530
520   MASST=MASST*0.4536
      MASSI=MASSI*0.4536
      MASSE=MASSE*0.4536
      THRUST=THRUST*4.4477
      PRINT 880, MASST,THRUST,MASSI,MASSE
530   IF (IUO,NE,3) GO TO 550
540   CONTINUE
550   IF (NPLOT,LT,0) GO TO 560
      TIME=T/LC $ NP=N+NSTART
      CALL PLOT (TITLE,TIME,NP)
      PRINT 1030, NP
C
C     CHECK FOR CONVERGENCE OF THE STEADY STATE SOLUTION
C
560   IF (DQM,LT,TCONV) GO TO 590
```

```
      IF (T,EQ,TSTOP) GO TO 590
      IF (N,EQ,NMAX) GO TO 590
      IF (NC,EQ,NPRINT) NC=0
      IF (NPC,EQ,NPLOT) NPC=0
570   CONTINUE
      NNN=N1
      N1=N3
      N3=NNN
580   CONTINUE
C
C     PUNCH A SIVS NAMELIST FOR RESTART
C
590   IF (NPLOT,GE,0) CALL ADV (10)
      IF (IPUNCH,EQ,0) GO TO 10
      DO 600 L=1,LMAX
      DO 600 M=1,MMAX
      P(L,M,N3)=P(L,M,N3)/PC
      RO(L,M,N3)=RO(L,M,N3)*G
600   CONTINUE
      PUNCH 930, NP,TIME
      DO 610 M=1,MMAX
      PUNCH 940, M
      PUNCH 950, (U(L,M,N3),L=1,LMAX)
610   CONTINUE
      DO 620 M=1,MMAX
      PUNCH 960, M
      PUNCH 950, (V(L,M,N3),L=1,LMAX)
620   CONTINUE
      DO 630 M=1,MMAX
      PUNCH 970, M
      PUNCH 980, (P(L,M,N3),L=1,LMAX)
630   CONTINUE
      DO 640 M=1,MMAX
      PUNCH 990, M
      PUNCH 1000, (RO(L,M,N3),L=1,LMAX)
640   CONTINUE
      PUNCH 1010
      NCARDS=((LMAX/7+2)*MMAX*4+22)
      PRINT 1020, NCARDS
      GO TO 10
C
C     FORMAT STATEMENTS
C
650   FORMAT (8A10)
660   FORMAT (1H1)
670   FORMAT (1H )
680   FORMAT (1H0)
690   FORMAT (1H0,15X,NAP, A COMPUTER PROGRAM FOR THE COMPUTATION OF TWO-DIMENSIONAL, TIME-DEPENDENT, INVISCID NOZZLE
    FLOW,//,37X,59HBY MICHAEL C. CLINE, T-3 - LOS ALAMOS SCIENTIFIC LABORATORY)
700   FORMAT (1H0,10X,18HPROGRAM ABSTRACT 26X,86HTHE EQUATIONS OF MOTION FOR TWO-DIMENSIONAL, TIME DEPENDENT,
    INVISCID FLOW IN A NOZZ2LE,/,21X,93HARE SOLVED USING THE SECOND-ORDER, MACCORMACK, FINITE-DIFFERENCE SCHEME, THE
    FLUID IS ASSUMED,/,21X,95HT0 BE A PERFECT GAS, ALL BOUNDARY CONDITIONS ARE COMPUTED USING A SECOND-ORDER,
    REFERENCE PLANE,/,21X,91HCHARACTERISTIC SCHEME, THE STEADY STATE SOLUTION IS OBTAINED AS THE ASYMPTOTIC SOLUTION
    FOR,/,21X,91HLARGE TIME, THE NOZZLES MAY BE EITHER CONVERGING, CONVERGING-DIVERGING, OR PLUG GEOMETRIES,)
710   FORMAT (1H0,10X,11HJOB TITLE -//21X,8A10)
720   FORMAT (1H0,10X,20HCONTROL PARAMETERS -)
730   FORMAT
    (1H0,20X,5HLMAX=,I2,2X,5HMMAX=,I2,3X,5HNMAX=,I4,2X,7HNPRINT=,I4,2X,6HTCONV=,F6.3,3X,4HFDT=,F4.2,2X,6HNSTAG=,I1,5X,5HNASM=,I1,4X,6HIUNIT
740   FORMAT (1H0,10X,13HFLUID MODEL -,//,21X,36HTHE RATIO OF SPECIFIC HEATS, GAMMA =,F6.4,26H AND THE GAS CONSTANT,
    R =,F9.4,15H (FT-LBF/LBM-R))
750   FORMAT (1H0,10X,13HFLUID MODEL -,//,21X,36HTHE RATIO OF SPECIFIC HEATS, GAMMA =,F6.4,26H AND THE GAS CONSTANT,
    R =,F9.4,9H (J/KG-K))
760   FORMAT (1H0,10X,21HBOUNDARY CONDITIONS -,//,21X,3HPT=,F9.4,7H (PSIA),5X,3HTT=,F9.4,4H (F),5X,6HTHETA=,F9.4,6H
    (DEG),5X,3HPE=,F9.4,7H (PSIA))
770   FORMAT (1H0,10X,21HBOUN0ARY CONDITIONS -,//,21X,3HPT=,F9.4,6H (KPA),5X,3HTT=,F9.4,4H (C),5X,6HTHETA=,F9.4,6H
    (DEG),5X,3HPE=,F9.4,6H (KPA))
780   FORMAT (1H0,10X,15HFLOW GEOMETRY -)
790   FORMAT (1H0,20X,47HTWO-DIMENSIONAL, PLANAR FLOW HAS BEEN SPECIFIED)
800   FORMAT (1H0,20X,36HAXISYMMETRIC FLOW HAS BEEN SPECIFIED)
810   FORMAT (1H ,30HINITIAL-DATA SURFACE - TIME = ,F10.8,8H SECONDS,4H (N=,I4,1H))
820   FORMAT (1H0,11X,1HL,4X,1HM,9X,1HX,10X,1HY,10X,1HU,1IX,1HV,12X,1HP,11X,3HRHO,9X,1HQ,11X,4HMACH,8X,1HT)
830   FORMAT (1H ,25X,4H(IN),7X,4H(IN),6X,5H(FPS),7X,5H(FPS),7X,6H(PSIA),6X,9H(LBM/FT3),4X,5H(FPS),10X,2HNO,8X,3H(F))
840   FORMAT (1H ,25X,4H(CM),7X,4H(CM),6X,5H(MPS),7X,5H(MPS),7X,6H (KPA),7X,7H(KG/M3),5X,5H(MPS),10X,2HNO,8X,3H(C))
850   FORMAT (1H ,7X,2I5,4F12.4,F13.4,F12.6,3F12.4)
860   FORMAT (1H ,20HSOLUTION SURFACE N0.,I5,3H - ,7HTIME = ,F10.8,20H SECONDS (DELTA T = ,F10.8,1H))
870   FORMAT (1H0,10X,5HMASS=,F9.4,10H (LBM/SEC),5X,7HTHRUST=,F11.4,6H (LBF),5X,6HMASSI=,F9.4,5X,6HMASSE=,F9.4)
```

```
880     FORMAT (1H0,10X,5HMASS=,F9.4,9H (KS/SEC),5X,7HTHRUST=,F11.4,10H (NEWTONS),5X,6HMASSI=,F9.4,5X,6HMASSE=,F9.4)
890     FORMAT (1H0,10X,21HBOUNDARY CONDITIONS
       -,//,22X,1HM,11X,8HPT(PSIA),10X,5HTT(F),10X,10HTHETA(DEG),10X,3HPE=,F7.3,7H (PSIA),/)
900     FORMAT (1H0,10X,21HBOUNDARY CONDITIONS
       -,//,22X,1HM,11X,7HPT(KPA),12X,5HTT(C),10X,10HTHETA(DEC),10X,3HPE=,F7.3,6H (KPA),/)
910     FORMAT (1H ,20X,I2,10X,F7.2,10X,F7.2,10X,F7.2)
920     FORMAT (1H0,78H***** THE RADIUS OF THE CENTERBODY IS LARGER THAN THE NOZZLE WALL RADIUS *****)
930     FORMAT (1X,18HSIVS N1D=0,NSTART=,I4,8H,TSTART=,F14.10,1H,)
940     FORMAT (1X,4HU(1,,I2,5H,1) =)
950     FORMAT (1X,7(F10.3,1H,))
960     FORMAT (1X,4HV(1,,I2,5H,1) =)
970     FORMAT (1X,4HP(1,,I2,5H,1) =)
980     FORMAT (1X,7(F10.4,1H,))
990     FORMAT (1X,4HRHO(1,,I2,5H,1) =)
1000    FORMAT (1X,7(F10.6,1H,))
1010    FORMAT (1X,1HS)
1020    FORMAT (1H0,27H***** EXPECT APPROXIMATELY ,I4,20H PUNCHED CARDS *****)
1030    FORMAT (1H0,31H***** EXPECT FILM OUTPUT FOR N=,I4,6H *****)
1040    FORMAT (1H ,2HN=,I4)
        END
```

B.2 Geometry Subroutine (geom.f)

```
      SUBROUTINE GEOM                                                   GEO   10
C                                                                       GEO   20
C     ***************************************************************** GEO   30
C                                                                       GEO   40
C     THIS SUBROUTINE CALCULATES THE NOZZLE RADIUS AND OUTER NORMAL     GEO   50
C                                                                       GEO   60
C     ***************************************************************** GEO   70
C                                                                       GEO   80
      COMMON /AV/ IAV,CAV,NST,SMP,LSS,CTA,XMU,XLA,RKMU,QUT(81,21),QVT(81GEO   90
     1,21),QPT(81,21)                                                   GEO  100
      COMMON /ONESID/ UD(4),VD(4),PD(4),ROD(4)                          GEO  110
      COMMON /SOLUTN/ U(81,21,2),V(81,21,2),P(81,21,2),RO(81,21,2)      GEO  120
      COMMON /CNTRLC/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,GAMMA,RGAS,GAM1,GAGEO  130
     1M2,L1,L2,L3,M1,M2,DX,DY,DT,N,N1,N3,NASM,IVEL,ICHAR,N1D,LJET,JFLAG,GEO  140
     2IERR,IUI,IUO,DXR,DYR,LD,MD,LMD1,LMD3,IB,RSTAR,RSTARS,NPLOT,G,PC,TCGEO  150
     3,LC,PLOW,ROLOW                                                    GEO  160
      COMMON /GEMTRYC/ NGEOM,XI,RI,XT,RT,XE,RE,RCI,RCT,ANGI,ANGE,XW(81),GEO  170
     1YW(81),XWI(81),YWI(81),NXNY(81),NWPTS,IINT,IDIF,LT,NDIM           GEO  180
      COMMON /GCB/ NGCB,XICB,RICB,XTCB,RTCB,XECB,RECB,RCICB,RCTCB,ANGICBGEO  190
     2,ANGECB,XCB(81),YCB(81),XCBI(81),YCBI(81),NXNYCB(81),NCBPTS,IINTCBGEO  200
     3,IDIFCB,LECB                                                      GEO  210
      COMMON /BCC/ PT(21),TT(21),THETA(21),PE,MASSE,MASSI,MASST,THRUST,NGEO  220
     1STAG                                                              GEO  230
      REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE                            GEO  240
C                                                                       GEO  250
      GO TO (10,30,120,170), NGEOM                                      GEO  260
C                                                                       GEO  270
C     CONSTANT AREA DUCT CASE                                           GEO  280
C                                                                       GEO  290
10    PRINT 230                                                         GEO  300
      IF (IUI.EQ.1) PRINT 250, XI,RI,XE                                 GEO  310
      IF (IUI.EQ.2) PRINT 260, XI,RI,XE                                 GEO  320
      LT=LMAX                                                           GEO  330
      DX=(XE-XI)/(LMAX-1)                                               GEO  340
      XT=XE                                                             GEO  350
      RT=RI                                                             GEO  360
      RE=RI                                                             GEO  370
      DO 20 L=1,LMAX                                                    GEO  380
      YW(L)=RI                                                          GEO  390
      NXNY(L)=0.0                                                       GEO  400
20    CONTINUE                                                          GEO  410
      IF (JFLAG.EQ.0) GO TO 210                                         GEO  420
C                                                                       GEO  430
      XWL=XI+(LJET-2)*DX                                                GEO  440
      IF (IUI.EQ.1) PRINT 370, XWL,LJET,LMAX                            GEO  450
      IF (IUI.EQ.2) PRINT 380, XWL,LJET,LMAX                            GEO  460
      GO TO 210                                                         GEO  470
C                                                                       GEO  480
C     CIRCULAR-ARC, CONICAL NOZZLE CASE                                 GEO  490
```

```
C                                                                    GEO  500
 30     PRINT 230                                                    GEO  510
        IF (RCI,EQ,0.0,OR,RCT,EQ,0.0) GO TO 200                      GEO  520
        ANI=ANGI*3.141593/180.0                                      GEO  530
        ANE=ANGE*3.141593/180.0                                      GEO  540
        XTAN=XI+RCI*SIN(ANI)                                         GEO  550
        RTAN=RI+RCI*(COS(ANI)-1.0)                                   GEO  560
        RT1=RT-RCT*(COS(ANI)-1.0)                                    GEO  570
        XT1=XTAN+(RTAN-RT1)/TAN(ANI)                                 GEO  580
        IF (XT1,GE,XTAN) GO TO 40                                    GEO  590
        XT1=XTAN                                                     GEO  600
        RT1=RTAN                                                     GEO  610
 40     XT=XT1+RCT*SIN(ANI)                                          GEO  620
        XT2=XT+RCT*SIN(ANE)                                          GEO  630
        RT2=RT+RCT*(1.0-COS(ANE))                                    GEO  640
        RE=RT2+(XE-XT2)*TAN(ANE)                                     GEO  650
        LT=1                                                         GEO  660
        DX=(XE-XI)/(LMAX-1)                                          GEO  670
        IF (IUI,EQ,1) PRINT 270, XI,RI,RT,XE,RCI,RCT,ANGI,ANGE,XT,RE GEO  680
        IF (IUI,EQ,2) PRINT 280, XI,RI,RT,XE,RCI,RCT,ANGI,ANGE,XT,RE GEO  690
        DO 110 L=1,LMAX                                              GEO  700
        X=XI+(L-1)*DX                                                GEO  710
        IF (X,GE,XI,AND,X,LE,XTAN) GO TO 50                          GEO  720
        IF (X,GT,XTAN,AND,X,LE,XT1) GO TO 60                         GEO  730
        IF (X,GT,XT1,AND,X,LE,XT) GO TO 70                           GEO  740
        IF (X,GT,XT,AND,X,LE,XT2) GO TO 80                           GEO  750
        IF (X,GT,XT2,AND,X,LE,XE) GO TO 90                           GEO  760
C                                                                    GEO  770
 50     YW(L)=RI+RC*(COS(ASIN((X-XI)/RCI))-1.0)                      GEO  780
        NXNY(L)=(XI-XI)/(YW(L)-RI+RCI)                               GEO  790
        GO TO 100                                                    GEO  800
C                                                                    GEO  810
 60     YW(L)=RT1+(XT1-X)*TAN(ANI)                                   GEO  820
        NXNY(L)=TAN(ANI)                                             GEO  830
        GO TO 100                                                    GEO  840
C                                                                    GEO  850
```

B.3 Inlet Boundary Conditions (inlet.f)

OCR note: this listing received a conservative normalization pass for obvious token errors; unresolved ambiguities are tracked in conversion notes.

```
      SUBROUTINE INLET
C     ************************************************************
C
c     THIS SUBROUTINE CALCULATES THE BOUNDARY MESH POINTS AT THE NOZZLE
C     INLET FOR SUBSONIC FLOW
C
C     ************************************************************
C
      COMMON /AV/ IAV,CAV,NST,SMP,LSS,CTA,XMU,XLA,RKMU,QUT(81,21),QVT(81
     1,21),QPT(81,21)
      COMMON /ONESID/ UD(4),VD(4),PD(4),ROD(4)
      COMMON /SOLUTN/ U(81,21,2),V(81,21,2),P(81,21,2),RO(81,21,2)
      COMMON /CNTRLC/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,GAMMA,RGAS,GAM1,GA
     1M2,L1,L2,L3,M1,M2,DX,DY,DT,N,N1,N3,NASM,IVEL,ICHAR,N1D,LJET,JFLAG,
     2IERR,IUI,IUO,DXR,DYR,LD,MD,LMD1,LMD3,IB,RSTAR,RSTARS,NPLOT,G,PC,TC
     3,LC,PLOW,ROLOW
      COMMON /GEMTRYC/ NGEOM,XI,RI,XT,RT,XE,RE,RCI,RCT,ANGI,ANGE,XW(81),
     1YW(81),XWI(81),YWI(81),NXNY(81),NWPTS,IINT,IDIF,LT,NDIM
      COMMON /GCB/ NGCB,XICB,RICB,XTCB,RTCB,XECB,RECB,RCICB,RCTCB,ANGICB
     1,ANGECB,XCB(81),YCB(81),XCBI(81),YCBI(81),NXNYCB(81),NCBPTS,IINTCB
     2,IDIFCB,LECB
      COMMON /BCC/ PT(21),TT(21),THETA(21),PE,MASSE,MASSI,MASST,THRUST,N
     1STAG
      REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE
C
      GRGB=GAMMA*RGAS*G
      X3=XI
      ATERM2=0.0
      ATERM3=0.0
```

```
      DO 180 ICHAR=1,2
      DO 180 M=1,MMAX
      LMN1=1+LD*(M-1)+LMD1
      LMN3=1+LD*(M-1)+LMD3
      L1MN1=2+LD*(M-1)+LMD1
      L1M1N1=2+LD*(M-2)+LMD1
      LM1N1=1+LD*(M-2)+LMD1
      LM1N3=1+LD*M+LMD3
      CALL MAP (2,1,M,AL,BE,DE,2,AL1,BE1,DE1)
      U2=U(LMN1)
      A2=SQRT(GAMMA*P(LMN1)/RO(LMN1))
      IF (ICHAR,EQ,2) GO TO 10
      U(LMN3)=U2
      V(LMN3)=V(LMN1)
      A3=A2
C
C     CALCULATE THE PROPERTY INTERPOLATING POLYNOMIAL COEFFICIENTS
C
 10   BU=(U(L1MN1)-U(LMN1))*DXR
      BV=(V(L1MN1)-V(LMN1))*DXR
      BP=(P(L1MN1)-P(LMN1))*DXR
      BRO=(RO(L1MN1)-RO(LMN1))*DXR
      BYCB=(YCB(2)-YCB(1))*DXR
      BAL=(AL1-AL)*DXR
      BBE=(BE1-BE)*DXR
      CU=U(1,M,N1)-BU*X3
      CV=V(1,M,N1)-BV*X3
      CP=P(1,M,N1)-BP*X3
      CRO=RO(1,M,N1)-BRO*X3
      CYCB=YCB(1)-BYCB*X3
      CAL=AL-BAL*X3
      CBE=BE-BBE*X3
C
C     CALCULATE THE CROSS DERIVATIVE INTERPOLATING POLYNOMIAL
C     COEFFICIENTS
C
      IF (M,EQ,1) GO TO 20
      DU=(U(L1MN1)-U(L1M1N1))*DYR
      DV=(V(L1MN1)-V(L1M1N1))*DYR
      DP=(P(L1MN1)-P(L1M1N1))*DYR
      DRO=(RO(L1MN1)-RO(L1M1N1))*DYR
      DU1=(U(LMN1)-U(LM1N1))*DYR
      DV1=(V(LMN1)-V(LM1N1))*DYR
      DP1=(P(LMN1)-P(LM1N1))*DYR
      DRO1=(RO(LMN1)-RO(LM1N1))*DYR
      GO TO 40
 20   IF (NGCB,NE,0) GO TO 30
      DU=0.0
      DV=V(2,2,N1)*DYR
      DP=0.0
      DRO=0.0
      DU1=0.0
      DV1=V(1,2,N1)*DYR
      DP1=0.0
      DRO1=0.0
      GO TO 40
 30   DU=(U(2,2,N1)-U(2,1,N1))*DYR
      DV=(V(2,2,N1)-V(2,1,N1))*DYR
      DP=(P(2,2,N1)-P(2,1,N1))*DYR
      DRO=(RO(2,2,N1)-RO(2,1,N1))*DYR
      DU1=(U(1,2,N1)-U(1,1,N1))*DYR
      DV1=(V(1,2,N1)-V(1,1,N1))*DYR
      DP1=(P(1,2,N1)-P(1,1,N1))*DYR
      DRO1=(RO(1,2,N1)-RO(1,1,N1))*DYR
 40   BDU=(DU-DU1)*DXR
      BDV=(DV-DV1)*DXR
      BDP=(DP-DP1)*DXR
      BDRO=(DRO-DRO1)*DXR
      CDU=DU1-BDU*X3
      CDV=DV1-BDV*X3
      CDP=DP1-BDP*X3
      CDRO=DRO1-BDRO*X3
C
C     CALCULATE X2
C
      IF (ICHAR,EQ,2) A3=SQRT(GAMMA*P(LMN3)/RO(LMN3))
```

```
      DO 50 IL=1,2
      X2=X3-(U(1,M,N3)-A3+U2-A2)*0.5*DT
C
C     INTERPOLATE FOR THE PROPERTIES
C
      U2=BU*X2+CU
      P2=BP*X2+CP
      RO2=BRO*X2+CRO
      A2=SQRT(GAMMA*P2/RO2)
50    CONTINUE
      V2=BV*X2+CV
      YCB2=BYCB*X2+CYCB
      AL2=BAL*X2+CAL
      BE2=BBE*X2+CBE
      UV2=U2*AL2+V2*BE2
C
C     INTERPOLATE FOR THE CROSS DERIVATIVES
C
      DU2=BDU*X2+CDU
      DV2=BDV*X2+CDV
      DP2=BDP*X2+CDP
      DRO2=BDRO*X2+CDRO
C
C     CALCULATE THE PSI TERMS
C
      IF (NDIM,EQ,0) GO TO 70
      IF (M,EQ,1,AND,NGCB,EQ,0) GO TO 60
      ATERM2=RO2*V2/((DY*(M-1))/BE2+YCB2)
      GO TO 70
60    ATERM2=RO2*BE2*DV2
70    PSI12=-UV2*DRO2-RO2*AL2*DU2-RO2*BE2*DV2-ATERM2
      PSI22=-UV2*DU2-AL2*DP2/RO2
      PSI42=-UV2*DP2+A2*A2*UV2*DRO2
      IF (ICHAR,EQ,1) GO TO 130
C
C     CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
C
      IF (M,EQ,1,AND,NGCB,EQ,0) GO TO 80
      IF (M,EQ,MMAX) GO TO 90
      DU3=(U(LM1N3)-U(LMN3))*DYR
      DV3=(V(LM1N3)-V(LMN3))*DYR
      DP3=(P(LM1N3)-P(LMN3))*DYR
      DRO3=(RO(LM1N3)-RO(LMN3))*DYR
      GO TO 100
80    DU3=0.0
      DV3=V(1,2,N3)*DYR
      DP3=0.0
      DRO3=0.0
      GO TO 100
90    DU3=(U(1,MMAX,N3)-U(1,M1,N3))*DYR
      DV3=(V(1,MMAX,N3)-V(1,M1,N3))*DYR
      DP3=(P(1,MMAX,N3)-P(1,M1,N3))*DYR
      DRO3=(RO(1,MMAX,N3)-RO(1,M1,N3))*DYR
C
C     CALCULATE THE PSI TERMS AT THE SOLUTION POINT
C
100   IF (NDIM,EQ,0) GO TO 120
      IF (M,EQ,1,AND,NGCB,EQ,0) GO TO 110
      ATERM3=RO(LMN3)*V(LMN3)/((DY*(M-1))/BE+YCB(1))
      GO TO 120
110   ATERM3=RO(LMN3)*BE*DV3
120   UV3=U(LMN3)*AL+V(LMN3)*BE
      PSI13=-UV3*DRO3-RO(LMN3)*AL*DU3-RO(LMN3)*BE*DV3-ATERM3
      PSI23=-UV3*DU3-AL*DP3/RO(LMN3)
      PSI43=-UV3*DP3+A3*A3*UV3*DRO3
      GO TO 140
130   PSI23=PSI22
      PSI43=PSI42
      PSI13=PSI12
C
C     SOLVE THE COMPATIBILITY EQUATIONS FOR U,V,P, AND RO
C
140   MN3=SQRT(U(LMN3)*U(LMN3)+V(LMN3)*V(LMN3))/A3
      T2=P2/(RO2*RGAS*G)
      PSI1B=(PSI12+PSI13)*0.5
      PSI2B=(PSI22+PSI23)*0.5
```

```
      PSI4B=(PSI42+PSI43)*0.5
      GPSI1B=GAMMA*PSI1B
      TTHETA=TAN(THETA(M))
      UCORR=0.5+0.5/SQRT(1.0+TTHETA*TTHETA)
C
      DO 160 ITER=1,20
      DEM=(1.0+GAM2*MN3*MN3)
      P(LMN3)=PT(M)/(DEM**GAM1)
      T3=TT(M)/DEM
      PB=(P2+P(LMN3))*0.5
      RTB=RGAS*(T2+T3)*0.5*G
      U(LMN3)=U2+DT*PSI2B+(P(LMN3)-P2-(PSI4B+RTB*GPSI1B)*DT)*SQRT(RTB/GA
     1MMA)/PB
      U(LMN3)=U(LMN3)*UCORR
      V(LMN3)=-U(LMN3)*TTHETA
      OMN3=MN3
      MN3=SQRT((U(LMN3)*U(LMN3)+V(LMN3)*V(LMN3))/(T3*GRGB))
      IF (OMN3.NE.0.0) GO TO 150
      IF (ABS(MN3-OMN3).LE.0.0001) GO TO 170
      GO TO 160
150   IF (ABS((MN3-OMN3)/OMN3).LE.0.001) GO TO 170
160   CONTINUE
C
      PRINT 190, M,N
170   RO(LMN3)=P(LMN3)/(RGAS*T3*G)
180   CONTINUE
      RETURN
C
190   FORMAT (1H0,58H***** THE SOLUTION FOR NOZZLE ENTRANCE BOUNDARY POI
     1NT ( 1,,I2,1H,,I4,43H) FAILED TO CONVERGE IN 20 ITERATIONS *****)
      END
```

B.4 Wall Boundary Conditions (wall.f)

OCR note: this listing received a conservative normalization pass for obvious token errors; unresolved ambiguities are tracked in conversion notes.

```
      SUBROUTINE WALL
C
C     ****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE BOUNDARY MESH POINTS AT THE NOZZLE
C     WALL, EXHAUST JET BOUNDARY, AND CENTERBODY
C
C     ****************************************************************
C
      COMMON /AV/ IAV,CAV,NST,SMP,LSS,CTA,XMU,XLA,RKMU,QUT(81,21),QVT(81
     1,21),QPT(81,21)
      COMMON /ONESID/ UD(4),VD(4),PD(4),ROD(4)
      COMMON /SOLUTN/ U(81,21,2),V(81,21,2),P(81,21,2),RO(81,21,2)
      COMMON /CNTRLC/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,GAMMA,RGAS,GAM1,GA
     1M2,L1,L2,L3,M1,M2,DX,DY,DT,N,N1,N3,NASM,IVEL,ICHAR,N1D,LJET,JFLAG,
     2IERR,IUI,IUO,DXR,DYR,LD,MD,LMD1,LMD3,IB,RSTAR,RSTARS,NPLOT,G,PC,TC
     3,LC,PLOW,ROLOW
      COMMON /GEMTRYC/ NGEOM,XI,RI,XT,RT,XE,RE,RCI,RCT,ANGI,ANGE,XW(81),
     1YW(81),XWI(81),YWI(81),NXNY(81),NWPTS,IINT,IDIF,LT,NDIM
      COMMON /GCB/ NGCB,XICB,RICB,XTCB,RTCB,XECB,RECB,RCICB,RCTCB,ANGICB
     2,ANGECB,XCB(81),YCB(81),XCBI(81),YCBI(81),NXNYCB(81),NCBPTS,IINTCB
     3,IDIFCB,LECB
      COMMON /BCC/ PT(21),TT(21),THETA(21),PE,MASSE,MASSI,MASST,THRUST,N
     1STAG
      REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE
C
      IF (N.EQ.1) DELY=0.005
      XWID=0.0
      IF (IB.EQ.1) GO TO 10
      Y1=0.0 $ Y3=0.0 $ MDUM=1 $ MDUM1=2 $ SIGN=-1.0
      GO TO 20
10    Y1=1.0 $ Y3=1.0 $ MDUM=MMAX $ MDUM1=M1 $ SIGN=1.0
20    ATERM2=0.0
      ATERM3=0.0
```

```
      LDUM=LMAX
      IF(ICHAR.EQ.2) LDUM=L1
      LMDM=LD*(MDUM-1)
      LMDM1=LD*(MDUM1-1)
      DYS=SIGN*DYR
      DO 350 L=2,LDUM
      LMN1=L+LMDM+LMD1
      LMN3=L+LMDM+LMD3
      LM1N1=L+LMDM1+LMD1
      L1MN1=L-1+LMDM+LMD1
      L1MN3=L+1+LMDM+LMD3
      L1M1N1=L-1+LMDM1+LMD1
      IF (JFLAG.EQ.0) GO TO 50
      IF (IB.EQ.2) GO TO 50
C
      XWID=XWI(L)
      IF (ICHAR.EQ.1) GO TO 30
C
C     USE THE DUMMY ARRAYS TO MANIPULATE THE ONE-SIDED SOLUTIONS
C
      IF (L.NE.LJET-2) GO TO 30
      U(L1MN3)=UD(3)
      V(L1MN3)=VD(3)
      P(L1MN3)=PD(3)
      RO(L1MN3)=ROD(3)
      GO TO 50
 30   IF (L.NE.LJET-1) GO TO 40
      IF (ICHAR.EQ.1) UOLD=U(LMN1)
      U(LMN1)=UD(1)
      V(LMN1)=VD(1)
      P(LMN1)=PD(1)
      RO(LMN1)=ROD(1)
      GO TO 50
 40   IF (L.EQ.LJET) GO TO 50
      U(L1MN1)=UD(2)
      V(L1MN1)=VD(2)
      P(L1MN1)=PD(2)
      RO(L1MN1)=ROD(2)
C
 50   U1=U(LMN1)
      V1=V(LMN1)
      P1=P(LMN1)
      RO1=RO(LMN1)
      U2=U1
      V2=V1
      A1=SQRT(GAMMA*P1/RO1)
      A2=A1
      IF (ICHAR.EQ.2) GO TO 60
      U3=U1
      V3=V1
      P3=P1
      RO3=RO1
      A3=A1
      GO TO 70
 60   U3=U(LMN3)
      V3=V(LMN3)
      P3=P(LMN3)
      RO3=RO(LMN3)
      A3=SQRT(GAMMA*P3/RO3)
C
C     CALCULATE THE PROPERTY INTERPOLATING POLYNOMIAL COEFFICIENTS
C
 70   BU=(U1-U(LM1N1))*DYS
      BV=(V1-V(LM1N1))*DYS
      BP=(P1-P(LM1N1))*DYS
      BRO=(RO1-RO(LM1N1))*DYS
      CU=U1-BU*Y3
      CV=V1-BV*Y3
      CP=P1-BP*Y3
      CRO=RO1-BRO*Y3
C
C     CALCULATE THE CROSS DERIVATIVE INTERPOLIATING POLYNOMIAL
C     COEFFICIENTS
C
      DU=(U1-U(L1MN1))*DXR
      DV=(V1-V(L1MN1))*DXR
```

```
      DP=(P1-P(L1MN1))*DXR
      DRO=(RO1-RO(L1MN1))*DXR
      DU1=(U(LM1N1)-U(L1M1N1))*DXR
      DV1=(V(LM1N1)-V(L1M1N1))*DXR
      DP1=(P(LM1N1)-P(L1M1N1))*DXR
      DRO1=(RO(LM1N1)-RO(L1M1N1))*DXR
      BDU=(DU-DU1)*DYS
      BDV=(DV-DV1)*DYS
      BDP=(DP-DP1)*DYS
      BDRO=(DRO-DRO1)*DYS
      CDU=DU-BDU*Y3
      CDV=DV-BDV*Y3
      CDP=DP-BDP*Y3
      CDRO=DRO-BDRO*Y3
C
C     CALCULATE Y2
C
      CALL MAP (1,L,MDUM,AL,BE,DE,LD1,AL1,BE1,DE1)
      ALS=SQRT(AL*AL+BE*BE)
      UV3=U3*AL+V3*BE+DE
      AL2=AL
      DO 90 ILL=1,3
      UV2=U2*AL2+V2*BE+DE
      Y2=Y3-(UV2+SIGN*AL*ALS*A2+UV3+SIGN*ALS*A3)*DT*0.5
C
C     INTERPOLATE FOR THE PROPERTIES
C
      U2=BU*Y2+CU
      V2=BV*Y2+CV
      P2=BP*Y2+CP
      RO2=BRO*Y2+CRO
      AL2=Y2*AL
      AD=GAMMA*P2/RO2
      IF (AD.GT.0.0) GO TO 80
      PRINT 360, N,L,MDUM
      IERR=1
      RETURN
80    A2=SQRT(AD)
90    CONTINUE
C
C     INTERPOLATE FOR THE CROSS DERIVATIVES
C
      DU1=DU
      DV1=DV
      DP1=DP
      DRO1=DRO
      DU2=BDU*Y2+CDU
      DV2=BDV*Y2+CDV
      DP2=BDP*Y2+CDP
      DRO2=BDRO*Y2+CDRO
C
C     CALCULATE THE PSI TERMS
C
      IF (NDIM.EQ.0) GO TO 110
      IF (IB.EQ.2) GO TO 100
      ATERM2=RO2*V2/(YCB(L)+Y2/BE)
      GO TO 110
100   ATERM2=RO2*V2/(YCB(L)+Y2/BE)
      IF (IAV.EQ.0) GO TO 110
      ATDS=RO2*V(L,2,N1)*DYR*BE
      IF (ABS(ATERM2).GT.ABS(ATDS)) ATERM2=ATDS
C
110   PSI21=-U1*DU1-DP1/RO1
      PSI31=-U1*DV1
      PSI41=-U1*DP1+A1*A1*U1*DRO1
      PSI12=-U2*DRO2-RO2*DU2-ATERM2
      PSI22=-U2*DU2-DP2/RO2
      PSI32=-U2*DV2
      PSI42=-U2*DP2+A2*A2*U2*DRO2
      IF (ICHAR.EQ.1) GO TO 150
C
C     CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
C
      IF (JFLAG.EQ.0) GO TO 120
      IF (IB.EQ.2) GO TO 120
      IF (L.EQ.2) GO TO 120
```

```
      IF (L.NE.LJET-1) GO TO 120
      IF (LJET.EQ.2) GO TO 120
      GO TO 130
  120 DU3=(U(L1MN3)-U3)*DXR
      DV3=(V(L1MN3)-V3)*DXR
      DP3=(P(L1MN3)-P3)*DXR
      DRO3=(RO(L1MN3)-RO3)*DXR
      GO TO 140
  130 DU3=(U3-U(L-1,MDUM,N3))*DXR
      DV3=(V3-V(L-1,MDUM,N3))*DXR
      DP3=(P3-P(L-1,MDUM,N3))*DXR
      DRO3=(RO3-RO(L-1,MDUM,N3))*DXR
C
C     ENTER THE EXHAUST JET ITERATION LOOP
C
  140 IF (JFLAG.EQ.0) GO TO 150
      IF (IB.EQ.2) GO TO 150
      IF (L.LT.LJET) GO TO 150
      YWI(L)=YW(L)
      UDUM=U(LMN3)
      VDUM=V(LMN3)
      PDUM=P(LMN3)
      RODUM=RO(LMN3)
  150 DO 290 NJ=1,10
      IF (ICHAR.EQ.1) GO TO 250
      IF (JFLAG.EQ.0) GO TO 210
      IF (IB.EQ.2) GO TO 210
      IF (L.LT.LJET) GO TO 210
      IF (NJ.EQ.1) GO TO 200
      IF (NJ.GT.2) GO TO 180
  160 YWOLD=YW(L)
      POLD=P(LMN3)
      IF (P(LMN3).LT.PE) GO TO 170
      YW(L)=YW(L)+DELY
      GO TO 190
  170 YW(L)=YW(L)-DELY
      GO TO 190
  180 IF (P(LMN3).EQ.POLD) GO TO 160
      DYDP=(YW(L)-YWOLD)/(P(LMN3)-POLD)
      YWNEW=YW(L)+DYDP*(PE-P(LMN3))
      YWOLD=YW(L)
      POLD=P(LMN3)
      YW(L)=YWNEW
  190 IF (YW(L).LT.(0.98*YWOLD)) YW(L)=0.98*YWOLD
      IF (YW(L).GT.(1.02*YWOLD)) YW(L)=1.02*YWOLD
  200 NXNY(L)=-(YW(L)-YW(L-1))*DXR
      XWI(L)=(YW(L)-YWI(L))/DT
      XWID=XWI(L)
      CALL MAP (1,L,MDUM,AL,BE,DE,LD1,AL1,BE1,DE1)
      ALS=SQRT(AL*AL+BE*BE)
      U(LMN3)=UDUM
      V(LMN3)=VDUM
      P(LMN3)=PDUM
      RO(LMN3)=RODUM
C
C     CALCULATE THE PSI TERMS AT THE SOLUTION POINT
C
  210 IF (NDIM.EQ.0) GO TO 240
      IF (IB.EQ.2) GO TO 220
      ATERM3=RO3*V2/(YCB(L)+1.0/BE)
      GO TO 240
  220 IF (YCB(L).EQ.0.0) GO TO 230
      ATERM3=RO3*V3/YCB(L)
      IF (IAV.EQ.0) GO TO 240
      ATDS=RO3*V(L,2,N3)*DYR*BE
      IF (ABS(ATERM3).GT.ABS(ATDS)) ATERM3=ATDS
      GO TO 240
  230 ATERM3=RO3*V(L,2,N3)*DYR*BE
C
  240 PSI13=-U3*DRO3-RO3*DU3-ATERM3
      PSI23=-U3*DU3-DP3/RO3
      PSI33=-U3*DV3
      PSI43=-U3*DP3+A3*A3*U3*DRO3
C     CALCULATE THE COMPATIBILITY EQUATION COEFFICIENTS
C
  250 ABR=NXNY(L)
```

```
      IF (IB.EQ.2) ABR=NXNYCB(L)
      ALB=0.5*(AL2+AL)/ALS
      BEB=BE/ALS
      A1B=(A1+A3)*0.5
      A2B=(A2+A3)*0.5
      RO1B=(RO1+RO3)*0.5
      RO2B=(RO2+RO3)*0.5
      IF (ICHAR.EQ.1) GO TO 260
      PSI21B=(PSI21+PSI23)*0.5
      PSI31B=(PSI31+PSI33)*0.5
      PSI41B=(PSI41+PSI43)*0.5
      PSI12B=(PSI12+PSI13)*0.5
      PSI22B=(PSI22+PSI23)*0.5
      PSI32B=(PSI32+PSI33)*0.5
      PSI42B=(PSI42+PSI43)*0.5
      GO TO 270
 260  PSI21B=PSI21
      PSI31B=PSI31
      PSI41B=PSI41
      PSI12B=PSI12
      PSI12B=PSI12
      PSI22B=PSI22
      PSI32B=PSI32
      PSI42B=PSI42
C
C     SOLVE THE COMPATIBILITY EQUATIONS FOR U,V,P, AND RO
C
 270  U(LMN3)=(U(LMN1)-ABR*(V(LMN1)-XWID)+(PSI21B-ABR*PSI31B)*DT)/(1.0+A
     1BR*ABR)
      V(LMN3)=-U(LMN3)*ABR+XWID
      P(LMN3)=P2-SIGN*RO2B*A2B*(ALB*(U(LMN3)-U2)+BEB*(V(LMN3)-V2))+(PSI4
     22B+A2B*A2B*PSI12B+SIGN*RO2B*A2B*(ALB*PSI22B+BEB*PSI32B))*DT
      IF (P(LMN3).LE.0.0) P(LMN3)=PLOW*PC
      RO(LMN3)=RO(LMN1)+(P(LMN3)-P(LMN1)-PSI41B*DT)/(A1B*A1B)
      IF (RO(LMN3).LE.0.0) RO(LMN3)=ROLOW/G
      IF (IAV.EQ.0) GO TO 280
C
C     ADD THE ARTIFICAL VISCOSITY FOR SHOCK CALCULATIONS
C
      IF (ICHAR.EQ.1) GO TO 280
      U(LMN3)=U(LMN3)+(QUT(L,MDUM)-ABR*QVT(L,MDUM))/(1.0+ABR*ABR)
      V(LMN3)=-U(LMN3)*ABR
      P(LMN3)=P(LMN3)+QPT(L,MDUM)
 280  IF (JFLAG.EQ.0) GO TO 350
      IF (IB.EQ.2) GO TO 350
      IF (L.LT.LJET-1) GO TO 350
      IF (L.EQ.LJET-1) GO TO 300
      IF (ICHAR.EQ.1) GO TO 350
      DELP=ABS((P(LMN3)-PE)/PE)
      IF (DELP.LE.0.001) GO TO 350
 290  CONTINUE
      GO TO 350
C
C     SOLVE THE COMPATIBILITY EQUATIONS FOR THE DOWNSTREAM SIDE OF THE
C     NOZZLE WALL EXIT POINT
C
 300  UD(3)=U(LMN3)
      VD(3)=V(LMN3)
      PD(3)=P(LMN3)
      ROD(3)=RO(LMN3)
      PD(4)=PE
      XM1=SQRT((UD(3)*UD(3)+VD(3)*VD(3))/(GAMMA*PD(3)/ROD(3)))
      DUMD=1.0+GAM2*XM1*XM1
      TD=PD(3)/ROD(3)/RGAS/G
      TTD=TD*DUMD
      IF (PE.GT.PD(3).AND.XM1.GE.1.0) GO TO 310
      TTD=TD*DUMD
      IF (PE.GT.PD(3).AND.XM1.GE.1.0) GO TO 310
      PTD=PD(3)*DUMD**GAM1
      ROD(4)=ROD(3)*(PE/PD(3))**(1.0/GAMMA)
      GO TO 320
 310  PRD=PE/PD(3)
      GAMD=(GAMMA+1.0)/(GAMMA-1.0)
      ROD(4)=ROD(3)*(GAMD*PRD+1.0)/(PRD+GAMD)
 320  TE=PE/ROD(4)/RGAS/G
      XMACH=SQRT((TTD/TE-1.0)/GAM2)
```

```
      SS=SQRT(GAMMA*PE/ROD(4))
      VMAG=XMACH*SS
      UD(4)=VMAG/SQRT(1.0+NXNY(LJET)*NXNY(LJET))
      VD(4)=-UD(4)*NXNY(LJET)
C
C     AVERAGE THE 1-SIDED MACH NOS FOR THE INTERIOR POINT CALCULATIONS
C
      XM2=SQRT((UD(4)*UD(4)+VD(4)*VD(4))/(GAMMA*PD(4)/ROD(4)))
      IF (XM1.GE.1.0) GO TO 350
      XMB=(XM1+XM2)/2.0
      IF (XMB.GE.1.0) GO TO 330
      DPL=1.0
      DPR=1.0
      GO TO 340
330   DPL=XM2-1.0
      DPR=1.0-XM1
      XMB=1.0
340   DPLR=DPR+DPL
      DUM=1.0+GAM2*XMB*XMB
      TEMP=TTD/DUM
      P(LMN3)=PTD/DUM**GAM1
      RO(LMN3)=P(LMN3)/(RGAS*TEMP*G)
      QA=SQRT(2.0*GAM1*(RGAS*TTD*G-P(LMN3)/RO(LMN3)))
      DNXNY=(DPR*NXNY(LJET)+DPL*NXNY(L))/DPLR
      U(LMN3)=QA/SQRT(1.0+DNXNY*DNXNY)
      V(LMN3)=-U(LMN3)*DNXNY
      IF (ICHAR.EQ.1) GO TO 350
      UD(1)=UD(3)
      VD(1)=VD(3)
      PD(1)=PD(3)
      ROD(1)=ROD(3)
      UD(2)=UD(4)
      VD(2)=VD(4)
      PD(2)=PD(4)
      ROD(2)=ROD(4)
350   CONTINUE
      IF (JFLAG.EQ.0) RETURN
      IF (IB.EQ.2) RETURN
      IF (ICHAR.EQ.1) RETURN
      U(LJET-1,MMAX,N1)=UOLD
      YWI(LMAX)=YW(LMAX)
      YW(LMAX)=2.0*YW(L1)-YW(L2)
      NXNY(LMAX)=-(YW(LMAX)-YW(L1))*DXR
      XWI(LMAX)=(YW(LMAX)-YWI(LMAX))/DT
      DELY=ABS(YW(LJET)-YWI(LJET))
      IF (DELY.EQ.0.0) DELY=0.0001
      RETURN
C
360   FORMAT (1H0,61H***** A NEGATIVE SQUARE ROOT OCCURED IN SUBROUTINE
     1WALL AT N=,I4,4H, L=,I2,8H, AND M=,I2,6H *****)
      END
```

B.5 Interior Mesh Calculations (inter.f)

```
      SUBROUTINE INTER
C
C     ****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE NOZZLE RADIUS AND OUTER NORMAL
C
C     ****************************************************************
C
      COMMON /AV/ IAV,CAV,NST,SMP,LSS,CTA,XMU,XLA,RKMU,QUT(81,21),QVT(81
     1,21),QPT(81,21)
      COMMON /ONESID/ UD(4),VD(4),PD(4),ROD(4)
      COMMON /SOLUTN/ U(81,21,2),V(81,21,2),P(81,21,2),RO(81,21,2)
      COMMON /CNTRLC/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,GAMMA,RGAS,GAM1,GA
     1M2,L1,L2,L3,M1,M2,DX,DY,DT,N,N1,N3,NASM,IVEL,ICHAR,N1D,LJET,JFLAG,
     2IERR,IUI,IUO,DXR,DYR,LD,MD,LMD1,LMD3,IB,RSTAR,RSTARS,NPLOT,G,PC,TC
     3,LC,PLOW,ROLOW
      COMMON /GEMTRYC/ NGEOM,XI,RI,XT,RT,XE,RE,RCI,RCT,ANGI,ANGE,XW(81),
     1YW(81),XWI(81),YWI(81),NXNY(81),NWPTS,IINT,IDIF,LT,NDIM
```

```
      COMMON /GCB/ NGCB,XICB,RICB,XTCB,RTCB,XECB,RECB,RCICB,RCTCB,ANGICB
     2,ANGECB,XCB(81),YCB(81),XCBI(81),YCBI(81),NXNYCB(81),NCBPTS,IINTCB
     3,IDIFCB,LECB
      COMMON /BCC/ PT(21),TT(21),THETA(21),PE,MASSE,MASSI,MASST,THRUST,N
     1STAG
      REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE
C
      ATERM=0.0
      IF (ICHAR,EQ,2) GO TO 40
C
C     COMPUTE THE TENTATIVE SOLUTION AT T+DT
C
      MDUM=1
      IF (NGCB.NE.0) MDUM=2
      DO 30 L=2,LMAX
      DO 30 M=MDUM,M1
      CALL MAP (1,L,M,AL,BE,DE,LD1,AL1,BE1,DE1)
      LMD2=LD*(M-1)
      LMN1=L+LMD2+LMD1
      LMN3=L+LMD2+LMD3
      L1MN1=L-1+LMD2+LMD1
      LM1N1=L+LD*(M-2)+LMD1
      UB=U(LMN1)
      VB=V(LMN1)
      PB=P(LMN1)
      ROB=RO(LMN1)
      ASB=GAMMA*PB/ROB
      IF (M.NE.1) GO TO 10
      DUDX=(UB-U(L1MN1))*DXR
      DPDX=(PB-P(L1MN1))*DXR
      DRODX=(ROB-RO(L1MN1))*DXR
      DVDY=(4.0*V(L,2,N1)-V(L,3,N1))*0.5*DYR
      V(LMN3)=0.0
C
      URHS=-UB*DUDX-DPDX/ROB
      RORHS=-UB*DRODX-ROB*DUDX-(1+NDIM)*ROB*BE*DVDY
      PRHS=-UB*DPDX+ASB*(RORHS+UB*DRODX)
      GO TO 20
10    IF (NDIM.EQ.1) ATERM=ROB*VB/((M-1)*DY/BE+YCB(L))
      UVB=UB*AL+VB*BE+DE
      DUDX=(UB-U(L1MN1))*DXR
      DVDX=(VB-V(L1MN1))*DXR
      DPDX=(PB-P(L1MN1))*DXR
      DRODX=(ROB-RO(L1MN1))*DXR
      DUDY=(UB-U(LM1N1))*DYR
      DVDY=(VB-V(LM1N1))*DYR
      DPDY=(PB-P(LM1N1))*DYR
      DRODY=(ROB-RO(LM1N1))*DYR
C
      URHS=-UB*DUDX-UVB*DUDY-(DPDX+AL*DPDY)/ROB
      VRHS=-UB*DVDX-UVB*DVDY-BE*DPDY/ROB
      RORHS=-UB*DRODX-UVB*DRODY-ROB*(DUDX+AL*DUDY+BE*DVDY)-ATERM
      PRHS=-UB*DPDX-UVB*DPDY+ASB*(RORHS+UB*DRODX+UVB*DRODY)
      V(LMN3)=V(LMN1)+VRHS*DT
20    U(LMN3)=U(LMN1)+URHS*DT
      P(LMN3)=P(LMN1)+PRHS*DT
      RO(LMN3)=RO(LMN1)+RORHS*DT
      IF (P(LMN3).LE.0.0) P(LMN3)=PLOW*PC
      IF (RO(LMN3).LE.0.0) RO(LMN3)=ROLOW/G
30    CONTINUE
      RETURN
C
C     COMPUTE THE FINAL SOLUTION AT T+DT
C
40    MDUM=1
      IF (NGCB.NE.0) MDUM=2
      DO 70, L=2,L1
      DO 70 M=MDUM,M1
      CALL MAP (1,L,M,AL,BE,DE,LD1,AL1,BE1,DE1)
      LMD2=LD*(M-1)
      LMN1=L+LMD2+LMD1
      LMN3=L+LMD2+LMD3
      L1MN3=L+1+LMD2+LMD3
      LM1N3=L+LD*M+LMD3
      UB=U(LMN3)
      VB=V(LMN3)
```

```
      PB=P(LMN3)
      ROB=RO(LMN3)
      ASB=GAMMA*PB/ROB
      IF (M.NE.1) GO TO 50
      DUDX=(U(L1MN3)-UB)*DXR
      DPDX=(P(L1MN1)-PB)*DXR
      DRODX=(RO(L1MN1)-ROB)*DXR
      DVDY=(4.0*V(L,2,N3)-V(L,3,N3))*0.5*DYR
      V(LMN3)=0.0
C
      URHS=-UB*DUDX-DPDX/ROB
      RORHS=-UB*DRODX-ROB*DUDX-(1+NDIM)*ROB*BE*DVDY
      PRHS=-UB*DPDX+ASB*(RORHS+UB*DRODX)
      GO TO 60
50    IF (NDIM.EQ.1) ATERM=ROB*VB/((M-1)*DY/BE+YCB(L))
      UVB=UB*AL+VB*BE+DE
      DUDX=(U(L1MN3)-UB)*DXR
      DVDX=(V(L1MN3)-VB)*DXR
      DPDX=(P(L1MN3)-PB)*DXR
      DRODX=(RO(L1MN3)-ROB)*DXR
      DUDY=(U(LM1N3)-UB)*DYR
      DVDY=(V(LM1N3)-VB)*DYR
      DPDY=(P(LM1N3)-PB)*DYR
      DRODY=(RO(LM1N3)-ROB)*DYR
C
      URHS=-UB*DUDX-UVB*DUDY-(DPDX+AL*DPDY)/ROB
      VRHS=-UB*DVDX-UVB*DVDY-BE*DPDY/ROB
      RORHS=-UB*DRODX-UVB*DRODY-ROB*(DUDX+AL*DUDY+BE*DVDY)-ATERM
      PRHS=-UB*DPDX-UVB*DPDY+ASB*(RORHS+UB*DRODX+UVB*DRODY)
      V(LMN3)=(V(LMN1)+V(LMN3)+VRHS*DT)*0.5
60    U(LMN3)=(U(LMN1)+U(LMN3)+URHS*DT)*0.5
      P(LMN3)=(P(LMN1)+P(LMN3)+PRHS*DT)*0.5
      RO(LMN3)=(RO(LMN1)+RO(LMN3)+RORHS*DT)*0.5
      IF (P(LMN3).LE.0.0) P(LMN3)=PLOW*PC
      IF (RO(LMN3).LE.0.0) RO(LMN3)=ROLOW/G
      IF (IAV.EQ.0) GO TO 70
C
C     ADD THE ARTIFICIAL VISCOSITY FOR SHOCK CALCULATIONS
C
      U(LMN3)=U(LMN3)+QUT(L,M)
      V(LMN3)=V(LMN3)+QVT(L,M)
      IF (M.EQ.1) V(LMN3)=0.0
      P(LMN3)=P(LMN3)+QPT(L,M)
70    CONTINUE
      RETURN
      END
```

B.6 Mass Flow Calculations (masflo.f)

```
      SUBROUTINE MASFLO(ISURF)                                          MAS  10
C                                                                       MAS  20
C     ***************************************************************   MAS  30
C                                                                       MAS  40
C     THIS SUBROUTINE CALCULTES THE INITIAL-DATA OR SOLUTION SURFACE    MAS  50
C     MASS FLOW AND THRUST                                              MAS  60
C                                                                       MAS  70
C     ***************************************************************   MAS  80
C                                                                       MAS  90
      COMMON /AV/ IAV,CAV,NST,SMP,LSS,CTA,XMU,XLA,RKMU,QUT(81,21),QVT(81MAS 100
     1,21),QPT(81,21)                                                   MAS 110
      COMMON /ONESID/ UD(4),VD(4),PD(4),ROD(4)                          MAS 120
      COMMON /SOLUTN/ U(81,21,2),V(81,21,2),P(81,21,2),RO(81,21,2)      MAS 130
      COMMON /CNTRLC/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,GAMMA,RGAS,GAM1,GAMAS 140
     1M2,L1,L2,L3,M1,M2,DX,DY,DT,N,N1,N3,NASM,IVEL,ICHAR,N1D,LJET,JFLAG,MAS 150
     2IERR,IUI,IUO,DXR,DYR,LD,MD,LMD1,LMD3,IB,RSTAR,RSTARS,NPLOT,G,PC,TCMAS 160
     3,LC,PLOW,ROLOW                                                    MAS 170
      COMMON /GEMTRYC/ NGEOM,XI,RI,XT,RT,XE,RE,RCI,RCT,ANGI,ANGE,XW(81),MAS 180
     1YW(81),XWI(81),YWI(81),NXNY(81),NWPTS,IINT,IDIF,LT,NDIM           MAS 190
      COMMON /GCB/ NGCB,XICB,RICB,XTCB,RTCB,XECB,RECB,RCICB,RCTCB,ANGICBMAS 200
     1,ANGECB,XCB(81),YCB(81),XCBI(81),YCBI(81),NXNYCB(81),NCBPTS,IINTCBMAS 210
     2,IDIFCB,LECB                                                      MAS 220
      COMMON /BCC/ PT(21),TT(21),THETA(21),PE,MASSE,MASSI,MASST,THRUST,NMAS 230
```

```
     3STAG                                                             MAS  240
      REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE                           MAS  250
C                                                                      MAS  260
      LC2=LC*LC                                                        MAS  270
      LDUM=LMAX-1                                                      MAS  280
      IF (LT,EQ,LMAX) LT=LMAX-1                                        MAS  290
      IF (JFLAG,NE,0) LDUM=LJET-1                                      MAS  300
      IF (ISURF,EQ,1,OR,N1D,EQ,0) GO TO 30                             MAS  310
C                                                                      MAS  320
C     CALCULATE THE MASS FLOW AND THRUST FOR THE 1-D INITIAL-DATA      MAS  330
C     SURFACE                                                          MAS  340
C                                                                      MAS  350
      IF (NDIM,EQ,1) GO TO 10                                          MAS  360
      AREAI=(YW(1)-YCB(1))/LC2                                         MAS  370
      AREAT=(YW(LT)-YCB(LT))/LC2                                       MAS  380
      AREAE=(YW(LDUM)-YCB(LDUM))/LC2                                   MAS  390
      GO TO 20                                                         MAS  400
10    AREAI=3.141593*(YW(1)**2-YCB(1)**2)/LC2                          MAS  410
      AREAT=3.141593*(YW(LT)**2-YCB(LT)**2)/LC2                        MAS  420
      AREAE=3.141593*(YW(LDUM)**2-YCB(LDUM)**2)/LC2                    MAS  430
      GO TO 20                                                         MAS  440
20    VMI=SQRT(U(1,1,1)**2+V(1,1,1)**2)                                MAS  450
      VMT=SQRT(U(LT,1,1)**2+V(LT,1,1)**2)                              MAS  460
      VME=SQRT(U(LDUM,1,1)**2+V(LDUM,1,1)**2)                          MAS  470
      MASSI=RO(1,1,1)*VMI*AREAI*G                                      MAS  480
      MASST=RO(LT,1,1)*VMT*AREAT*G                                     MAS  490
      MASSE=RO(LDUM,1,1)*VME*AREAE*G                                   MAS  500
      THRUST=RO(LDUM,1,1)*U(LDUM,1,1)**2*AREAE                         MAS  510
C                                                                      MAS  520
C     CALCULATE THE MASS FLOW AND THRUST FOR THE 2-D INITIAL-DATA      MAS  530
C     SURFACE                                                          MAS  540
C                                                                      MAS  550
30    MASSI=0.0                                                        MAS  560
      MASST=0.0                                                        MAS  570
      MASSE=0.0                                                        MAS  580
      THRUST=0.0                                                       MAS  590
      DYI=DY*(YW(1)-YCB(1))                                            MAS  600
      DYT=DY*(YW(LT)-YCB(LT))                                          MAS  610
      DYE=DY*(YW(LDUM)-YCB(LDUM))                                      MAS  620
      ND=1                                                             MAS  630
      IF (ISURF,EQ,1) ND=N3                                            MAS  640
      DO 60 M=1,M1                                                     MAS  650
      RADI=(M-1)*DYI+YCB(1)                                            MAS  660
      RADT=(M-1)*DYT+YCB(LT)                                           MAS  670
      RADE=(M-1)*DYE+YCB(LDUM)                                         MAS  680
      IF (NDIM,EQ,1) GO TO 40                                          MAS  690
      AREAI=DYI/LC2                                                    MAS  700
      AREAT=DYT/LC2                                                    MAS  710
      AREAE=DYE/LC2                                                    MAS  720
      GO TO 50                                                         MAS  730
40    AREAI=3.141593*((RADI+DYI)**2-RADI**2)/LC2                       MAS  740
      AREAT=3.141593*((RADT+DYT)**2-RADT**2)/LC2                       MAS  750
      AREAE=3.141593*((RADE+DYE)**2-RADE**2)/LC2                       MAS  760
      ROUI=(RO(1,M,ND)*U(1,M,ND)+RO(1,M+1,ND)*U(1,M+1,ND))*0.5         MAS  770
      ROUT=(RO(LT,M,ND)*U(LT,M,ND)+RO(LT,M+1,ND)*U(LT,M+1,ND))*0.5     MAS  780
      ROUE=(RO(LDUM,M,ND)*U(LDUM,M,ND)+RO(LDUM,M+1,ND)*U(LDUM,M+1,ND))*0MAS  790
     1.5                                                               MAS  800
      ROUE2=(RO(LDUM,M,ND)*U(LDUM,M,ND)**2+RO(LDUM,M+1,ND)*U(LDUM,M+1,NDMAS  810
     1)**2)*0.5                                                        MAS  820
      MASSI=MASSI+ROUI*AREAI*G                                         MAS  830
      MASST=MASST+ROUT*AREAT*G                                         MAS  840
      MASSE=MASSE+ROUE*AREAE*G                                         MAS  850
      THRUST=THRUST+ROU2E*AREAE                                        MAS  860
60    CONTINUE                                                         MAS  870
      RETURN                                                           MAS  880
      END                                                              MAS  890
```

B.7 One-Dimensional Initialization (onedim.f)

OCR note: this listing received a conservative normalization pass for obvious token errors; unresolved ambiguities are tracked in conversion notes.

```
SUBROUTINE ONEDIM
C
C    ************************************************************
C    THIS SUBROUTINE CALCULATES THE 1-D INITIAL-DATA SURFACE
C    ************************************************************
C
      COMMON /AV/ IAV,CAV,NST,SMP,LSS,CTA,XMU,XLA,RKMU,QUT(81,21),QVT(81
     1,21),QPT(81,21)
      COMMON /ONESID/ UD(4),VD(4),PD(4),ROD(4)
      COMMON /SOLUTN/ U(81,21,2),V(81,21,2),P(8l,21,2),RO(81,2l,2)
      COMMON /CNTRLC/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,GAMMA,RGAS,GAM1,GA
     1M2,L1,L2,L3,Ml,M2,DX,DY,DT,N,N1,N3,NASM,IVEL,ICHAR,N1D,LJET,JFLAG,
     2IERR,IUI,IUO,DXR,DYR,LD,MD,LMD1,LMD3,IB,RSTAR,RSTARS,NPLOT,G,PC,TC
     3,LC,PLOW,ROLOW
      COMMON /GEMTRYC/ NGEOM,XI,RI,XT,RT,XE,RE,RCI,RCT,ANGI,ANGE,XW(8l),
     1YW(8l),XWI(8l),YWI(8l),NXNY(8l),NWPTS,IINT,IDIF,LT,NDIM
      COMMON /GCB/ NGCB,XICB,RICB,XTCB,RTCB,XECB,RECB,RCICB,RCTCB,ANGICB
     1,ANGECB,XCB(8l),YCB(8l),XCBI(81),YCBI(8l),NXNYCB(8l),NCBPTS,IINTCB
     2,IDIFCB,LECB
      COMMON /BCC/ PT(21),TT(21),THETA(21),PE,MASSE,MASSI,MASST,THRUST,
     1NSTAG
      REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE

      MN3=0.01
      IF (N1D,EQ,-1,OR N1D,GT,2) MN3=2.0
      GRGAS=1.0/(RGAS*G)
      NXCK=0
      ACOEF=2.0/(GAMMA+1.0)
      BCOEF=(GAMMA-1.0)/(GAMMA+1.0)
      CCOEF=(GAMMA+1.0)/2.0/(GAMMA-1.0)
      IF (N1D,LT,0) GO TO 20
C
C     OVERALL LOOP
C
      IF (NGCB,NE,0) GO TO 10
      RSTAR=RT
      RSTARS=RT*RT
      GO TO 20
10    RSTAR=YW(LT)-YCB(LT)
      RSTARS=YW(LT)**2-YCB(LT)**2
20    DO 130 L=1,LMAX
      IF (L,EQ,1,AND,N1D,EQ,-1) GO TO 130
      IF (L,EQ,1,AND,N1D,GT,2) GO TO 130
      X=XI+DX*(L-1)
      IF (N1D,LT,0) GO TO 50
      IF (NGCB,NE,0) GO TO 30
      IF (X,LT,XT) GO TO 50
      IF (X,GT,XT) GO TO 40
      MN3=1.0
      GO TO 100
30    IF (L,LT,LT) GO TO 50
      IF (L,GT,LT) GO TO 40
      MN3=1.0
      GO TO 100
40    IF (NXCK,EQ,1) GO TO 50
      IF (N1D,EQ,1,OR,N1D,EQ,3) MN3=1.1
      IF (N1D,EQ,2,OR,N1D,EQ,4) MN3=0.9
      NXCK=1
50    IF (NDIM,EQ,1) GO TO 60
      RAD=YW(L)-YCB(L)
      ARATIO=RAD/RSTAR
      GO TO 70
60    RADS=YW(L)**2-YCB(L)**2
      ARATIO=RADS/RSTARS
      C
      C NEWTON-RAPHSON ITERATION LOOP
      C
70    DO 90 ITER=1,20
      ABM = ACOEF + BCOEF * MN3**2
```

```
      ABMC = ABM**CCOEF
      FM = ABMC / MN3 - ARATIO
      FPM = ABMC * (2.0 * BCOEF * CCOEF/ABM-1.0/MN3**2)
      OMN3 = MN3
      MN3 = OMN3 - FM/FPM
      IF (MN3,GT,1.0,AND,OMN3,LT,1.0) MN3=0.99
      IF (MN3,LT,1.0,AND,OMN3,GT,1.0) MN3=1.01
      IF (MN3,GE,000) GO TO 80
      MN3=-MN3
      GO TO 90
80    IF (ABS(MN3-OMN3)/OMN3,LE,0.0005) GO TO 100
90    CONTINUE
      PRINT 140, L
C
C     Fill IN 2-D ARRAYS LOOP
C
100   DEM = 1.0 + GAM2 * MN3 * MN3
      DEMP = DEM**GAM1
      DNXNY = (NXNY(L) - NXNYCB(L)) / M1
      DO 120 M=1,MMAX
      P(L,M,1)=PT(M)/DEMP
      TEMP=TT(M)/DEM
      RO(L,M,1)=P(L,M,1)*GRGAS/TEMP
      Q=MN3*SQRT(GAMMA*P(L,M,1)/RO(L,M,1))
      DN=NXNYCB(L)+DNXNY*(M-1)
      DNS=DN*DN
      IF (DNS,EQ,0.0) GO TO 110
      SIGN=1.0
      IF (DN,GT,0.0) SIGN=-1.0
      U(L,M,1)=Q/SQRT(1.0+DNS)
      V(L,M,1)=SIGN*Q/SQRT(1.0+1.0/DNS)
      GO TO 120
      U(L,M,1)=Q
      V(L,M,1)=0.0
120   CONTINUE
130   CONTINUE
      RETURN
C
140   FORMAT (1H0,10X,93H***** THE 1-D SOLUTION FOR THE INITIAL-DATA SUR
     1FACE FAILED TO CONVERGE IN 20 ITERATIONS AT L=,I2,6H *****)
      END
```

B.8 Centerbody Geometry (geomcb.f)

This subroutine calculates the centerbody radius and slope for cases with a centerbody configuration. It is used in the GEOM routine for geometry preprocessing. *OCR note: this listing was reconstructed from Appendix C PDF text blocks for readability; minor geometric/formatted-text normalization was required where scan quality was poor.*

```
      SUBROUTINE GEOMCB
C
C     **************************************************************
C     THIS SUBROUTINE CALCULATES THE CENTERBODY RADIUS AND SLOPE
C     **************************************************************
C
C     OCR note:
C     Reconstructed from Appendix C listing text blocks. Some FORMAT
C     literal wording was normalized for readability in degraded regions.
C
      DIMENSION YW(81),NXNY(81)
      COMMON /AV/ IAV,CAV,NST,SMP,LSS,CTA,XMU,XLA,RKMU,QUT(81,21),QVT(81,21),QPT(81,21)
      COMMON /ONESID/ UD(4),VD(4),PD(4),ROD(4)
      COMMON /SOLUTN/ U(81,21,2),V(81,21,2),P(81,21,2),RO(81,21,2)
      COMMON /CNTRLC/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,GAMMA,RGAS,GAM1,GAM2,
     1L1,L2,L3,M1,M2,DX,DY,DT,N,N1,N3,NASM,IVEL,ICHAR,N1D,LJET,JFLAG,
     2IERR,IUI,IUO,DXR,DYR,LD,MD,LMD1,LMD3,IB,RSTAR,RSTARS,NPLOT,G,PC,TC,
     3LC,PLOW,ROLOW
      COMMON /GCB/ NGCB,XICB,RICB,XTCB,RTCB,XECB,RECB,RCICB,RCTCB,ANGICB,
     1ANGECB,XCB(81),YCB(81),XCBI(81),YCBI(81),NXNYCB(81),NCBPTS,IINTCB,
```

```
     2IDIFCB,LECB
      COMMON /BCC/ PT(21),TT(21),THETA(21),PE,MASSE,MASSI,MASST,THRUST,
     1NSTAG
      REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE
C
      GO TO (10,30,120,160), NGCB
C
C     CYLINDRICAL CENTERBODY CASE
C
10    IF (IUI.EQ.1) PRINT 230, XICB,RICB,XECB
      IF (IUI.EQ.2) PRINT 240, XICB,RICB,XECB
      LECB=LMAX
      DO 20 L=1,LMAX
      YCB(L)=RICB
      NXNYCB(L)=0.0
20    CONTINUE
      GO TO 200
C
C     CIRCULAR-ARC, CONICAL CENTERBODY CASE
C
30    XI=XICB
      RI=RICB
      RT=RTCB
      XE=XECB
      RCI=RCICB
      RCT=RCTCB
      ANGI=ANGICB
      ANGE=ANGECB
      RI=2.0*RT-RI
      IF (RCI.EQ.0.0.OR.RCT.EQ.0.0) GO TO 190
      ANI=ANGI*3.141593/180.0
      ANE=ANGE*3.141593/180.0
      XTAN=XI+RCI*SIN(ANI)
      RTAN=RI+RCI*(COS(ANI)-1.0)
      RT1=RT-RCT*(COS(ANI)-1.0)
      XT1=XTAN+(RTAN-RT1)/TAN(ANI)
      IF (XT1.GE.XTAN) GO TO 40
      XT1=XTAN
      RT1=RTAN
40    XT=XT1+RCT*SIN(ANI)
      XTCB=XT
      XT2=XT+RCT*SIN(ANE)
      RT2=RT+RCT*(1.0-COS(ANE))
      RE=RT2+(XE-XT2)*TAN(ANE)
      RECB=RE
      RI=2.0*RT-RI
      RE=2.0*RT-RE
      IF (IUI.EQ.1) PRINT 250, XI,RI,RT,XE,RCI,RCT,ANGI,ANGE,XT,RE
      IF (IUI.EQ.2) PRINT 260, XI,RI,RT,XE,RCI,RCT,ANGI,ANGE,XT,RE
      RI=2.0*RT-RI
      RE=2.0*RT-RE
      DO 110 L=1,LMAX
      X=XI+(L-1)*DX
      IF (X.GE.XI.AND.X.LE.XTAN) GO TO 50
      IF (X.GT.XTAN.AND.X.LE.XT1) GO TO 60
      IF (X.GT.XT1.AND.X.LE.XT) GO TO 70
      IF (X.GT.XT.AND.X.LE.XT2) GO TO 80
      IF (X.GT.XT2.AND.X.LE.XE) GO TO 90
C
50    YW(L)=RI+RCI*(COS(ASIN((X-XI)/RCI))-1.0)
      NXNY(L)=(XI-X)/(YW(L)-RI+RCI)
      GO TO 100
C
60    YW(L)=RT1+(XT1-X)*TAN(ANI)
      NXNY(L)=TAN(ANI)
      GO TO 100
C
70    YW(L)=RT+RCT*(1.0-COS(ASIN((XT-X)/RCT)))
      NXNY(L)=(XT-X)/(RCT+RT-YW(L))
      GO TO 100
C
80    YW(L)=RT+RCT*(1.0-COS(ASIN((X-XT)/RCT)))
      NXNY(L)=(XT-X)/(RCT+RT-YW(L))
      GO TO 100
C
90    YW(L)=RT2+(X-XT2)*TAN(ANE)
```

```
      NXNY(L)=-TAN(ANE)
C
  100 YCB(L)=2.0*RTCB-YW(L)
      NXNYCB(L)=-NXNY(L)
      IF (YCB(L).GE.0.0) GO TO 110
      YCB(L)=0.0
      NXNYCB(L)=0.0
  110 CONTINUE
      GO TO 200
C
C     GENERAL CENTERBODY CASE - INPUT CENTERBODY COORDINATES ONLY
C
  120 PRINT 220
      IF (IUI.EQ.1) PRINT 270, IINTCB,IDIFCB
      IF (IUI.EQ.2) PRINT 280, IINTCB,IDIFCB
      LDUM=NCBPTS
      IF (LMAX.GT.NCBPTS) LDUM=LMAX
      IP=1
      DO 130 L=1,LMAX
      XCB(L)=XICB+DX*(L-1)
      CALL MTLUP (XCB(L),YCB(L),IINTCB,NCBPTS,NCBPTS,1,IP,XCBI,YCBI)
  130 CONTINUE
      DO 150 L=1,LDUM
      IF (L.GT.LMAX) GO TO 140
      SLOPE=DIF(L,IDIFCB,LMAX,XCB,YCB)
      NXNYCB(L)=-SLOPE
      IF (YCB(L).GE.0.0) GO TO 135
      YCB(L)=0.0
      NXNYCB(L)=0.0
      SLOPE=-NXNYCB(L)
  135 IF (L.LE.NCBPTS.AND.L.LE.LMAX) PRINT 310, L,XCBI(L),YCBI(L),XCB(L),YCB(L),SLOPE
      IF (L.GT.NCBPTS.AND.L.LE.LMAX) PRINT 320, L,XCB(L),YCB(L),SLOPE
  140 IF (L.LE.NCBPTS.AND.L.GT.LMAX) PRINT 330, L,XCBI(L),YCBI(L)
  150 CONTINUE
      GO TO 200
C
C     GENERAL CENTERBODY CASE - INPUT CENTERBODY COORDINATES AND SLOPES
C
  160 PRINT 220
      IF (IUI.EQ.1) PRINT 290
      IF (IUI.EQ.2) PRINT 300
      DO 180 L=1,LMAX
      XCB(L)=XICB+DX*(L-1)
      IF (YCB(L).GE.0.0) GO TO 170
      YCB(L)=0.0
      NXNYCB(L)=0.0
  170 SLOPE=-NXNYCB(L)
      PRINT 340, L,XCB(L),YCB(L),SLOPE
  180 CONTINUE
      GO TO 200
C
  190 PRINT 350
      IERR=1
      RETURN
C
  200 IF (IUI.EQ.1) RETURN
      DO 210 L=1,LMAX
      YCB(L)=YCB(L)/2.54
  210 CONTINUE
      XICB=XICB/2.54
      XECB=XECB/2.54
      DX=DX/2.54
      RETURN
C
C     FORMAT STATEMENTS
C
  220 FORMAT (1H )
  230 FORMAT (1H0,20X,52HA CYLINDRICAL CENTERBODY HAS BEEN SPECIFIED BY,
     1XICB=,F8.4,12H (IN), RICB=,F8.4,16H (IN), AND XECB=,F8.4,5H (IN))
  240 FORMAT (1H0,20X,52HA CYLINDRICAL CENTERBODY HAS BEEN SPECIFIED BY,
     1XICB=,F8.4,12H (CM), RICB=,F8.4,16H (CM), AND XECB=,F8.4,5H (CM))
  250 FORMAT (1H0,20X,62HA CIRCULAR-ARC, CONICAL CENTERBODY HAS BEEN SPECIFIED BY XICB=,F8.4,5H (IN),
     17H, RICB=,F8.4,6H (IN),/,21X,8HRTCB=,F8.4,7H (IN),,5HXECB=,F8.4,5H (IN),
     28H, RCICB=,F8.4,5H (IN),8H, RCTCB=,F8.4,5H (IN),9H, ANGICB=,F6.2,7H (DEG),
     3/,21X,11HAND ANGECB=,F6.2,8H (DEG),,29HTHE COMPUTED VALUES ARE XTCB=,F8.4,
     45H (IN),10H AND RECB=,F8.4,6H (IN),)
```

```
 260  FORMAT (1H0,20X,62HA CIRCULAR-ARC, CONICAL CENTERBODY HAS BEEN SPECIFIED BY XICB=,F8.4,5H (CM),
     17H, RICB=,F8.4,6H (CM),/,21X,8HRTCB=,F8.4,7H (CM),,5HXECB=,F8.4,5H (CM),
     28H, RCICB=,F8.4,5H (CM),8H, RCTCB=,F8.4,5H (CM),9H, ANGICB=,F6.2,7H (DEG),
     3/,21X,11HAND ANGECB=,F6.2,8H (DEG),,29HTHE COMPUTED VALUES ARE XTCB=,F8.4,
     45H (CM),10H AND RECB=,F8.4,6H (CM),)
 270  FORMAT (1H0,20X,76HA GENERAL CENTERBODY HAS BEEN SPECIFIED BY THE FOLLOWING PARAMETERS,
     1 IINTCB=,I1,9H, IDIFCB=,I1,/,22X,1HL,10X,8HXCBI(IN),10X,8HYCBI(IN),9X,
     27HXCB(IN),10X,7HYCB(IN),11X,5HSLOPE,/)
 280  FORMAT (1H0,20X,76HA GENERAL CENTERBODY HAS BEEN SPECIFIED BY THE FOLLOWING PARAMETERS,
     1 IINTCB=,I1,9H, IDIFCB=,I1,/,22X,1HL,10X,8HXCBI(CM),10X,8HYCBI(CM),9X,
     27HXCB(CM),10X,7HYCB(CM),11X,5HSLOPE,/)
 290  FORMAT (1H0,20X,68HA GENERAL CENTERBODY HAS BEEN SPECIFIED BY THE FOLLOWING PARAMETERS,
     1 //,22X,1HL,11X,7HXCB(IN),10X,7HYCB(IN),11X,5HSLOPE,/)
 300  FORMAT (1H0,20X,68HA GENERAL CENTERBODY HAS BEEN SPECIFIED BY THE FOLLOWING PARAMETERS,
     1 //,22X,1HL,11X,7HXCB(CM),10X,7HYCB(CM),11X,5HSLOPE,/)
 310  FORMAT (1H ,20X,I2,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4,7X,F10.4)
 320  FORMAT (1H ,20X,I2,41X,F10.4,7X,F10.4,7X,F10.4)
 330  FORMAT (1H ,20X,I2,7X,F10.4,7X,F10.4)
 340  FORMAT (1H ,20X,I2,7X,F10.4,7X,F10.4,7X,F10.4)
 350  FORMAT (1H0,48H**** RCICB OR RCTCB WAS SPECIFIED AS ZERO ****)
      END
```

B.9 Table Interpolation (mtlup.f)

This subroutine performs multilinear interpolation on tabular input data. It is called by GEOM to interpolate geometry and flow properties from tables. This routine was adapted from a NASA-Langley program. *OCR note: this listing was reconstructed from Appendix C PDF text blocks for readability; diagnostic/comment text literals were normalized where scan artifacts obscured wording.*

```
      SUBROUTINE MTLUP (X,Y,M,N,MAX,NTAB,I,VARI,VARD)
C
C      THIS SUBROUTINE IS CALLED BY SUBROUTINE GEOM TO INTERPOLATE FOR
C     EQUALLY SPACED NOZZLE WALL COORDINATES FOR THE TABULAR INPUT CASE.
C     SUBROUTINE MTLUP WAS TAKEN FROM THE NASA-LANGLEY PROGRAM LIBRARY.
C
C     MODIFICATION OF LIBRARY INTERPOLATION SUBROUTINE FTLUP
C     MULTIPLE TABLE LOOK-UP ON ONE INDEPENDENT VARIABLE TABLE
C     USES AN EXTERNAL INTERVAL POINTER (I) TO START SEARCH
C      I LESS THAN 0 WILL CHECK MONOTONICITY
C
C     OCR note:
C      Reconstructed from Appendix C listing text blocks. Some original
C      diagnostic text literals were normalized where the scan was unreadable.
C      Diagnostic FORMAT wording is normalized for readability.
C
      DIMENSION VARI(N),VARD(MAX,1),Y(1),V(3),YY(2)
      LOGICAL EX
C
       IF (M.EQ.0) GO TO 170
       IF (N.LE.1) GO TO 170
      EX=.FALSE.
       IF (I.GE.0) GO TO 60
       IF (N.LT.2) GO TO 60
C
C     MONOTONICITY CHECK
C
       IF (VARI(2)-VARI(1)) 20,20,40
C
C     ERROR IN MONOTONICITY
C
 10   PRINT 190, N
      STOP
C
C     MONOTONIC DECREASING
C
 20   DO 30 J=2,N
       IF (VARI(J)-VARI(J-1)) 30,10,10
 30   CONTINUE
      GO TO 60
```

```
C
C     MONOTONIC INCREASING
C
 40   DO 50 J=2,N
      IF (VARI(J)-VARI(J-1)) 10,10,50
 50   CONTINUE
C
C     INTERPOLATION
C
 60   IF (I.LE.0) I=1
      IF (I.GE.N) I=N-1
C
C     LOCATE I INTERVAL (VARI(I) .LE. X .LT. VARI(I+1))
C
      IF ((VARI(I)-X)*(VARI(I+1)-X)) 100,100,70
C
C     IN GIVES DIRECTION FOR SEARCH OF INTERVALS
C
 70   IN=SIGN(1.0,(VARI(N)-VARI(I))*(X-VARI(I)))
C
C     IF X OUTSIDE ENDPOINTS, EXTRAPOLATE FROM END INTERVAL
C
 80   IF ((I+IN).LE.0) GO TO 90
      IF ((I+IN).GE.N) GO TO 90
      I=I+IN
      IF ((VARI(I)-X)*(VARI(I+1)-X)) 100,100,80
C
C     EXTRAPOLATION
C
 90   EX=.TRUE.
 100  IF (M.EQ.2) GO TO 120
C
C     FIRST ORDER
C
      DO 110 NT=1,NTAB
 110  Y(NT)=(VARD(I,NT)*(VARI(I+1)-X)-VARD(I+1,NT)*(VARI(I)-X))/
     1 (VARI(I+1)-VARI(I))
      IF (EX) I=I-IN
      RETURN
C
C     SECOND ORDER
C
 120  IF (N.EQ.2) GO TO 110
      IF (I.EQ.(N-1)) GO TO 140
      IF (I.EQ.1) GO TO 130
C
C     PICK THIRD POINT
C
      SK=VARI(I+1)-VARI(I)
      IF (ABS(X-VARI(I+1)).LT.ABS(VARI(I+2)-X)) GO TO 140
 130  L=I
      GO TO 150
 140  L=I-1
 150  V(1)=VARI(L)-X
      V(2)=VARI(L+1)-X
      V(3)=VARI(L+2)-X
      DO 160 NT=1,NTAB
      YY(1)=(VARD(L,NT)*V(2)-VARD(L+1,NT)*V(1))/(VARI(L+1)-VARI(L))
      YY(2)=(VARD(L+1,NT)*V(3)-VARD(L+2,NT)*V(2))/(VARI(L+2)-VARI(L+1))
 160  Y(NT)=(YY(1)*V(3)-YY(2)*V(1))/(VARI(L+2)-VARI(L))
      IF (EX) I=I-IN
      RETURN
C
C     ZERO ORDER
C
 170  DO 180 NT=1,NTAB
 180  Y(NT)=VARD(I,NT)
      RETURN
C
 190  FORMAT (1H1,49H TABLE BELOW OUT OF ORDER FOR MTLUP, N = ,I5)
      END
```

B.10 Numerical Differentiation (diff.f)

This function calculates numerical derivatives for tabular input data. It is used by GEOM to compute nozzle wall slopes for the tabular input case, providing finite-difference approximations to derivatives. *OCR note: this listing was reconstructed from Appendix C PDF text blocks for readability; comments were normalized while preserving the derivative computation logic.*

```
      FUNCTION DIF (L,M,NP,VARI,VARD)
C
C     THIS FUNCTION IS CALLED BY SUBROUTINE GEOM TO CALCULATE THE
C     NOZZLE WALL SLOPE FOR THE TABULAR INPUT CASE. FUNCTION DIF WAS
C     TAKEN FROM THE NASA-LANGLEY PROGRAM LIBRARY.
C
C     THIS FUNCTION SUBPROGRAM FINDS THE DERIVATIVE AT A GIVEN POINT L
C     FOR THE DESIRED X AND Y IN A GIVEN TABLE. THE N-POINT LAGRANGIAN
C     FORMULA IS USED WHERE N IS ODD.
C
C     L    = INTEGER, POINT OF X AND Y AT WHICH DERIVATIVE IS FOUND
C     M    = INTEGER, 1-5, TO DETERMINE THE POINT FORMULA, N=2*M+1
C     NP   = INTEGER, NUMBER OF POINTS IN TABLE OF VARIABLES
C     VARI = ARRAY OF INDEPENDENT VARIABLE, X, VARI(NP)
C     VARD = ARRAY OF DEPENDENT VARIABLE,   Y, VARD(NP)
C
C     OCR note:
C     Reconstructed from Appendix C listing text blocks; numeric logic
C     preserved and comments normalized for readability.
C
      DIMENSION VARI(NP),VARD(NP),X(11),Y(11)
C
      DIF=1.777E19
      IF (M.LT.1) RETURN
      N=2*M+1
      IF (M.GT.5.OR.N.GT.NP) RETURN
      M1=M+1
      M2=NP-M1
      K=L
      IF (L.LE.M1.OR.N.EQ.NP) GO TO 10
      K=M1
      IF (L.LT.M2) GO TO 10
      K=L-(NP-N)
10    MX=L-K
      DO 20 J=1,N
      MJ=MX+J
      X(J)=VARI(MJ)
20    Y(J)=VARD(MJ)
      A=1.0
      B=0.0
      C=0.0
      DO 40 J=1,N
      IF (J.EQ.K) GO TO 40
      P=1.0
      DO 30 I=1,N
      IF (I.EQ.J) GO TO 30
      P=P*(X(J)-X(I))
30    CONTINUE
      T=X(K)-X(J)
      B=B+Y(J)/(P*T)
      A=A*T
      C=C+1.0/T
40    CONTINUE
      DIF=A*B+Y(K)*C
      RETURN
      END
```

B.11 Mapping Functions (map.f)

This subroutine calculates coordinate mapping functions, converting between physical (x, y) coordinates and transformed (ξ, η, ζ) computational coordinates used in the finite-difference scheme. *OCR note: this listing was replaced with a cleaner in-repo transcription and is currently the most readable available version.*

```
      SUBROUTINE MAP(IP,L,M,AL,BE,DE,LD1,AL1,BE1,DE1)                  MAP  10
C                                                                      MAP  20
C     ************************************************************     MAP  30
C                                                                      MAP  40
C     THIS SUBROUTINE CALCULATES THE MAPPING FUNCTIONS                 MAP  50
C                                                                      MAP  60
C     ************************************************************     MAP  70
C                                                                      MAP  80
      COMMON /AV/ IAV,CAV,NST,SMP,LSS,CTA,XMU,XLA,RKMU,QUT(81,21),QVT(81MAP  90
     1,21),QPT(81,21)                                                  MAP 100
      COMMON /ONESID/ UD(4),VD(4),PD(4),ROD(4)                         MAP 110
      COMMON /SOLUTN/ U(81,21,2),V(81,21,2),P(81,21,2),RO(81,21,2)     MAP 120
      COMMON /CNTRLC/ LMAX,MMAX,NMAX,NPRINT,TCONV,FTD,GAMMA,RGAS,GAM1,GAMAP 130
     1M2,L1,L2,L3,M1,M2,DX,DY,DT,N,N1,N3,NASM,IVEL,ICHAR,N1D,LJET,JFLAG,MAP 140
     2IERR,IUI,IUO,DXR,DYR,LD,MD,LMND,LMD3,IB,RSTAR,RSTARS,NPLOT,G,PC,TCMAP 150
     3,LC,PLOW,ROLOW                                                   MAP 160
      COMMON /GEMTRYC/ NGEOM,XI,RI,XT,RT,XE,RE,RCI,RCT,ANGI,ANGE,XW(81),MAP 170
     1YW(81),XWI(81),YWI(81),NXNY(81),NWPTS,IINT,IDIF,LT,NDIM          MAP 180
      COMMON /GCB/ NGCB,XICB,RICB,XTCB,RTCB,XECB,RECB,RCICB,RCTCB,ANGICBMAP 190
     2,ANGECB,XCB(81),YCB(81),XCBI(81),YCBI(81),NXNYCB(81),NCBPTS,IINTCBMAP 200
     3,IDIFCB,LECB                                                     MAP 210
      COMMON /BCC/ PT(21),TT(21),THETA(21),PE,MASSE,MASSI,MASST,THRUST,NMAP 220
     1STAG                                                             MAP 230
      REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE                           MAP 240
C                                                                      MAP 250
      BE=1.0/(YW(L)-YCB(L))                                            MAP 260
      IF (IP,EQ,0) RETURN                                              MAP 270
      Y=(M-1)*DY                                                       MAP 280
      AL=BE*(NXNYCB(L)+Y*(NXNY(L)-NXNYCB(L)))                          MAP 290
      DE=-BE*Y*XWI(L)                                                  MAP 300
      IF (IP,EQ,1) RETURN                                              MAP 310
      BE1=1.0/(YW(LD1)-YCB(LD1))                                       MAP 320
      AL1=BE1*(NXNYCB(LD1)+Y*(NXNY(LD1)-NXNYCB(LD1)))                  MAP 330
      DE1=-BE1*Y*XWI(LD1)                                              MAP 340
      RETURN                                                           MAP 350
      END                                                              MAP 360
```

B.12 Artificial Viscosity (shock.f)

This subroutine calculates local artificial viscosity terms used to stabilize the solution across shock waves and steep gradients. Following the method of Lax and Wendroff, it provides controlled numerical dissipation to prevent oscillations. *OCR note: this listing was reconstructed from Appendix C PDF text blocks for readability and aligned to Appendix C line blocks in the viscosity-energy term region.*

```
      SUBROUTINE SHOCK(IPASS)
C
C     ************************************************************
C     THIS SUBROUTINE CALCULATES THE LOCAL ARTIFICAL VISCOSITY TERMS
C     FOR SHOCK COMPUTATIONS
C     ************************************************************
C
C     OCR note:
C     Reconstructed from Appendix C PDF text blocks and aligned to
C     Appendix C line blocks for the viscosity-energy term structure.
C
      COMMON /AV/ IAV,CAV,NST,SMP,LSS,CTA,XMU,XLA,RKMU,QUT(81,21),QVT(81,21),QPT(81,21)
```

```
      COMMON /ONESID/ UD(4),VD(4),PD(4),ROD(4)
      COMMON /SOLUTN/ U(81,21,2),V(81,21,2),P(81,21,2),RO(81,21,2)
      COMMON /CNTRLC/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,GAMMA,RGAS,GAM1,GAM2,
     1L1,L2,L3,M1,M2,DX,DY,DT,N,N1,N3,NASM,IVEL,ICHAR,N1D,LJET,JFLAG,
     2IERR,IUI,IUO,DXR,DYR,LD,MD,LMD1,LMD3,IB,RSTAR,RSTARS,NPLOT,G,PC,TC,
     3LC,PLOW,ROLOW
      COMMON /GEMTRYC/ NGEOM,XI,RI,XT,RT,XE,RE,RCI,RCT,ANGI,ANGE,XW(81),
     1YW(81),XWI(81),YWI(81),NXNY(81),NWPTS,IINT,IDIF,LT,NDIM
      COMMON /GCB/ NGCB,XICB,RICB,XTCB,RTCB,XECB,RECB,RCICB,RCTCB,ANGICB,
     1ANGECB,XCB(81),YCB(81),XCBI(81),YCBI(81),NXNYCB(81),NCBPTS,IINTCB,
     2IDIFCB,LECB
      COMMON /BCC/ PT(21),TT(21),THETA(21),PE,MASSE,MASSI,MASST,THRUST,
     1NSTAG
      REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE
C
      GO TO (10,160), IPASS
C
C     CALCULATE LOCAL ARTIFICAL VISCOSITY FOR SHOCK COMPUTATIONS
C
10    IF (N.NE.1) GO TO 30
      NC=0
      RG=RGAS*G
      CTA1=1.0-CTA
      DO 20 L=1,LMAX
      DO 20 M=1,MMAX
      QUT(L,M)=0.0
      QVT(L,M)=0.0
      QPT(L,M)=0.0
20    CONTINUE
30    RDUM=CAV*DT*DX*DY*2.0
      NC=NC+1
      NLINE=0
      IF (NC.NE.NPRINT) GO TO 40
      PRINT 200
      PRINT 190, N, PC
C
40    DO 150 L=LSS,L1
      DO 150 M=1,MMAX
      LMD2=LD*(M-1)+LMD1
      LMMD2=LD*(M-2)+LMD1
      LMPD2=LD*M+LMD1
      LMN1=L+LMD2
      LP=L+1+LMD2
      LM=L-1+LMD2
      MP=L+LMPD2
      MM=L+LMMD2
      LPMP=L+1+LMPD2
      LPMM=L+1+LMMD2
      LMMP=L-1+LMPD2
      LMMM=L-1+LMMD2
      CALL MAP (1,L,M,AL,BE,DE,LD1,AL1,BE1,DE1)
C
C     CHECK TO SEE IF THE DIVERGENCE OF THE VELOCITY IS NEGATIVE
C
      UX=0.5*(U(LP)-U(LM))*DXR
      IF (M.EQ.1) GO TO 50
      IF (M.EQ.MMAX) GO TO 60
      UY=0.5*(U(MP)-U(MM))*DYR
      VY=0.5*(V(MP)-V(MM))*DYR
      GO TO 70
50    UY=(U(MP)-U(LMN1))*DYR
      VY=(V(MP)-V(LMN1))*DYR
      GO TO 70
60    UY=(U(LMN1)-U(MM))*DYR
      VY=(V(LMN1)-V(MM))*DYR
70    DIV=UX+AL*UY+BE*VY
      IF (DIV.LT.0.0) GO TO 80
      QUT(L,M)=0.0
      QVT(L,M)=0.0
      QPT(L,M)=0.0
      GO TO 150
C
80    OQUT=QUT(L,M)
      OQVT=QVT(L,M)
      OQPT=QPT(L,M)
```

```
      UX1=(U(LMN1)-U(LM))*DXR
      UX2=(U(LP)-U(LMN1))*DXR
      VX1=(V(LMN1)-V(LM))*DXR
      VX2=(V(LP)-V(LMN1))*DXR
      TM=P(LM)/(RO(LM)*RG)
      T=P(LMN1)/(RO(LMN1)*RG)
      TP=P(LP)/(RO(LP)*RG)
      TX1=(T-TM)*DXR
      TX2=(TP-T)*DXR

      LDUM=L-1
      CALL MAP (1,LDUM,M,ALM,BEM,DE,LD1,AL1,BE1,DE1)
      LDUM=L+1
      CALL MAP (1,LDUM,M,ALP,BEP,DE,LD1,AL1,BE1,DE1)
      BE1=0.5*(BEM+BE)
      BE2=0.5*(BEP+BE)
      AL1=0.5*(ALM+AL)
      AL2=0.5*(ALP+AL)

      IF (M.EQ.1) GO TO 90
      IF (M.EQ.MMAX) GO TO 100
C
C     CALCULATE THE INTERIOR POINT QUANTITIES
C
      UY1=0.25*(U(MP)+U(LMMP)-U(MM)-U(LMMM))*DYR
      UY2=0.25*(U(MP)+U(LPMP)-U(MM)-U(LPMM))*DYR
      VY1=0.25*(V(MP)+V(LMMP)-V(MM)-V(LMMM))*DYR
      VY2=0.25*(V(MP)+V(LPMP)-V(MM)-V(LPMM))*DYR
      UX3=0.25*(U(LP)+U(LPMM)-U(LM)-U(LMMM))*DXR
      UX4=0.25*(U(LP)+U(LPMP)-U(LM)-U(LMMP))*DXR
      VX3=0.25*(V(LP)+V(LPMM)-V(LM)-V(LMMM))*DXR
      VX4=0.25*(V(LP)+V(LPMP)-V(LM)-V(LMMP))*DXR
      VY3=(V(LMN1)-V(MM))*DYR
      VY4=(V(MP)-V(LMN1))*DYR
      UY3=(U(LMN1)-U(MM))*DYR
      UY4=(U(MP)-U(LMN1))*DYR

      TMY=P(MM)/(RO(MM)*RG)
      TPY=P(MP)/(RO(MP)*RG)
      TMM=P(LMMM)/(RO(LMMM)*RG)
      TMP=P(LMMP)/(RO(LMMP)*RG)
      TPM=P(LPMM)/(RO(LPMM)*RG)
      TPP=P(LPMP)/(RO(LPMP)*RG)
      TY1=0.25*(TPY+TMP-TMY-TMM)*DYR
      TY2=0.25*(TPP+TPY-TPM-TMY)*DYR
      TX3=0.25*(TP+TPM-TM-TMM)*DXR
      TX4=0.25*(TPP+TP-TMP-TM)*DXR
      TY3=(T-TMY)*DYR
      TY4=(TPY-T)*DYR

      MDUM=M-1
      CALL MAP (1,L,MDUM,ALMY,BEMY,DE,LD1,AL1,BE1,DE1)
      MDUM=M+1
      CALL MAP (1,L,MDUM,ALPY,BEPY,DE,LD1,AL1,BE1,DE1)
      BE3=0.5*(BEMY+BE)
      BE4=0.5*(BEPY+BE)
      AL3=0.5*(ALMY+AL)
      AL4=0.5*(ALPY+AL)
      GO TO 110
C
C     CALCULATE THE CENTERLINE POINT QUANTITIES
C
 90   UY1=0.5*(U(MP)+U(LMMP)-U(LMN1)-U(LM))*DYR
      VY1=0.5*(V(MP)+V(LMMP)-V(LMN1)-V(LM))*DYR
      UY2=0.5*(U(LPMP)+U(MP)-U(LP)-U(LMN1))*DYR
      VY2=0.5*(V(LPMP)+V(MP)-V(LP)-V(LMN1))*DYR
      UX3=0.0
      VX3=0.0
      UX4=0.0
      VX4=0.0
      THEW=ATAN(-NXNYCB(L))
      THE=ATAN(V(MP)/U(MP))
      VMAG=SQRT(U(MP)*U(MP)+V(MP)*V(MP))
      RTHE=2.0*THEW-THE
      UR=VMAG*COS(RTHE)
      VR=VMAG*SIN(RTHE)
```

```
      UY3=(U(LMN1)-UR)*DYR
      VY3=(V(LMN1)-VR)*DYR
      UY4=(U(MP)-U(LMN1))*DYR
      VY4=(V(MP)-V(LMN1))*DYR
      TPY=P(MP)/(RO(MP)*RG)
      TMP=P(LMMP)/(RO(LMMP)*RG)
      TPP=P(LPMP)/(RO(LPMP)*RG)
      TY1=0.0
      TY2=0.0
      TX3=0.25*(TPP+TP-TMP-TM)*DXR
      TX4=TX3
      TY4=(TPY-T)*DYR
      TY3=-TY4
      BE3=BE
      AL3=AL
      MDUM=M+1
      CALL MAP (1,L,MDUM,AL4,BE4,DE,LD1,AL1,BE1,DE1)
      GO TO 110
C
C     CALCULATE THE WALL POINT QUANTITIES
C
100   UY1=0.5*(U(LMN1)+U(LM)-U(MM)-U(LMMM))*DYR
      VY1=0.5*(V(LMN1)+V(LM)-V(MM)-V(LMMM))*DYR
      UY2=0.5*(U(LP)+U(LMN1)-U(LPMM)-U(MM))*DYR
      VY2=0.5*(V(LP)+V(LMN1)-V(LPMM)-V(MM))*DYR
      UX3=0.0
      VX3=0.0
      UX4=0.0
      VX4=0.0
      UY3=(U(LMN1)-U(MM))*DYR
      VY3=(V(LMN1)-V(MM))*DYR
      THEW=ATAN(-NXNY(L))
      THE=ATAN(V(MM)/U(MM))
      VMAG=SQRT(U(MM)*U(MM)+V(MM)*V(MM))
      RTHE=2.0*THEW-THE
      UR=VMAG*COS(RTHE)
      VR=VMAG*SIN(RTHE)
      UY4=(UR-U(LMN1))*DYR
      VY4=(VR-V(LMN1))*DYR
      TPM=P(LPMM)/(RO(LPMM)*RG)
      TMY=P(MM)/(RO(MM)*RG)
      TMM=P(LMMM)/(RO(LMMM)*RG)
      TY1=0.0
      TY2=0.0
      TX3=0.25*(TP+TPM-TM-TMM)*DXR
      TX4=TX3
      TY3=(T-TMY)*DYR
      TY4=-TY3
      MDUM=M-1
      CALL MAP (1,L,MDUM,AL3,BE3,DE,LD1,AL1,BE1,DE1)
      BE4=BE
      AL4=AL
C
110   UXY1=UX1+AL1*UY1
      UXY2=UX2+AL2*UY2
      UXY3=UX3+AL3*UY3
      UXY4=UX4+AL4*UY4
      UXY12=0.5*(UX1+UX2+AL3*UY3+AL4*UY4)

      VXY1=VX1+AL1*VY1
      VXY2=VX2+AL2*VY2
      VXY3=VX3+AL3*VY3
      VXY4=VX4+AL4*VY4
      VXY12=0.5*(VX1+VX2+AL3*VY3+AL4*VY4)

      BUY1=BE1*UY1
      BUY2=BE2*UY2
      BUY3=BE3*UY3
      BUY4=BE4*UY4
      BUY34=0.5*(BUY3+BUY4)

      BVY1=BE1*VY1
      BVY2=BE2*VY2
      BVY3=BE3*VY3
      BVY4=BE4*VY4
      BVY34=0.5*(BVY3+BVY4)
```

```
      TXY1=TX1+AL1*TY1
      TXY2=TX2+AL2*TY2
      TXY3=TX3+AL3*TY3
      TXY4=TX4+AL4*TY4
C
C     CALCULATE THE ARTIFICAL VISCOSITY COEFFICIENTS
C
      DIV=UXY12+BVY34
      VID=VXY12-BUY34
      RLA=XLA*RDUM*ABS(DIV)/BE
      RMU=XMU*RDUM*ABS(VID)/BE
      RK=RMU*GAM1*RG/RKMU
      RLP2M=RLA+2.0*RMU
      RLA2=2.0*RLA
      RMU2=2.0*RMU
      RLPM=RLA+RMU
      UVTA=0.0
      VVTA=0.0
      PVTA=0.0
      PCTA=0.0

      IF (NDIM.EQ.0) GO TO 130
C
C     CALCULATE THE AXISYMMETRIC TERMS
C
      IF (M.EQ.1.AND.YCB(L).EQ.0.0) GO TO 120
      Y=FLOAT(M-1)*DY/BE+YCB(L)
      VB=V(LMN1)
      UVTA=(RLPM*VXY12+RMU*BUY34)/Y
      VVTA=RLP2M*(BVY34+VB/Y)/Y
      PVTA=(RLP2M*VB*VB/Y+RLA2*VB*(BVY34+UXY12))/Y
      PCTA=RK*0.5*(BE4*TY4+BE3*TY3)/Y
      DUVTA=RLPM*BE*(VXY4-VXY3)*DYR+RMU*BE*(BUY4-BUY3)*DYR
      DVVTA=RLP2M*0.5*BE*(BVY4-BVY3)*DYR
      DPVTA=(RLP2M+RLA2)*BVY34*BVY34+RLA2*BVY34*UXY12
      DPCTA=RK*BE*(BE4*TY4-BE3*TY3)*DYR
      IF (ABS(UVTA).GT.ABS(DUVTA)) UVTA=DUVTA
      IF (ABS(VVTA).GT.ABS(DVVTA)) VVTA=DVVTA
      IF (ABS(PVTA).GT.ABS(DPVTA)) PVTA=DPVTA
      IF (ABS(PCTA).GT.ABS(DPCTA)) PCTA=DPCTA
      GO TO 130
120   UVTA=RLPM*BE*(VXY4-VXY3)*DYR+RMU*BE*(BUY4-BUY3)*DYR
      VVTA=RLP2M*0.5*BE*(BVY4-BVY3)*DYR
      PVTA=(RLP2M+RLA2)*BVY34*BVY34+RLA2*BVY34*UXY12
      PCTA=RK*BE*(BE4*TY4-BE3*TY3)*DYR
C
C     CALCULATE THE ARTIFICAL VISCOSITY TERMS
C
130   QUT(L,M)=RLP2M*(UXY2-UXY1)*DXR
     1 + AL*(RLP2M*(UXY4-UXY3)+RLA*(BVY4-BVY3))*DYR
     2 + RMU*BE*(VXY4-BUY4-VXY3+BUY3)*DYR + UVTA

      QVT(L,M)=RMU*(VXY2-BUY2-VXY1+BUY1)*DXR
     1 + RMU*AL*(VXY4-BUY4-VXY3+BUY3)*DYR
     2 + BE*(RLA*(UXY4-UXY3)+RLP2M*(BVY4-BVY3))*DYR + VVTA

      QPT(L,M)=RO(LMN1)*(GAMMA-1.0)*
     1 (RLP2M*(UXY12*UXY12+BVY34*BVY34)
     2 + RMU*(VXY12*VXY12+BUY34*BUY34)
     3 + RLA2*UXY12*BVY34 + RMU2*BUY34*VXY12
     4 + RK*((TXY2-TXY1)*DXR + AL*(TXY4-TXY3)*DYR
     5 + BE*(BE4*TY4-BE3*TY3)*DYR) + PVTA + PCTA)

      QUT(L,M)=CTA*QUT(L,M)+CTA1*OQUT
      QVT(L,M)=CTA*QVT(L,M)+CTA1*OQVT
      QPT(L,M)=CTA*QPT(L,M)+CTA1*OQPT

      IF (NC.NE.NPRINT) GO TO 150
      NLINE=NLINE+1
      IF (NLINE.LT.55) GO TO 140
      PRINT 200
      PRINT 190, N, PC
      NLINE=1
140   QPT(L,M)=QPT(L,M)/PC
      PRINT 180, L,M,QUT(L,M),QVT(L,M),QPT(L,M)
```

```
      QPT(L,M)=QPT(L,M)*PC
150   CONTINUE

      IF (NC.EQ.NPRINT) NC=0
      RETURN
C
C     SMOOTH THE FLOW VARIABLES IF REQUESTED
C
160   IF (SMP.LT.0.0.OR.SMP.GE.1.0) RETURN
      SMP4=0.25*(1.0-SMP)
      DO 170 L=2,L1
      U(L,MMAX,N3)=SMP4*(U(L-1,MMAX,N3)+U(L+1,MMAX,N3)+2.0*U(L,M1,N3))
     1 +SMP*U(L,MMAX,N3)
      V(L,MMAX,N3)=-U(L,MMAX,N3)*NXNY(L)
      P(L,MMAX,N3)=SMP4*(P(L-1,MMAX,N3)+P(L+1,MMAX,N3)+2.0*P(L,M1,N3))
     1 +SMP*P(L,MMAX,N3)
      RO(L,MMAX,N3)=SMP4*(RO(L-1,MMAX,N3)+RO(L+1,MMAX,N3)+2.0*RO(L,M1,N3))
     1 +SMP*RO(L,MMAX,N3)

      U(L,1,N3)=SMP4*(U(L-1,1,N3)+U(L+1,1,N3)+2.0*U(L,2,N3))
     1 +SMP*U(L,1,N3)
      V(L,1,N3)=-U(L,1,N3)*NXNYCB(L)
      P(L,1,N3)=SMP4*(P(L-1,1,N3)+P(L+1,1,N3)+2.0*P(L,2,N3))
     1 +SMP*P(L,1,N3)
      RO(L,1,N3)=SMP4*(RO(L-1,1,N3)+RO(L+1,1,N3)+2.0*RO(L,2,N3))
     1 +SMP*RO(L,1,N3)

      DO 170 M=2,M1
      LMD2=LD*(M-1)+LMD3
      LMMD2=LD*(M-2)+LMD3
      LMPD2=LD*M+LMD3
      LMN3=L+LMD2
      LP=L+1+LMD2
      LM=L-1+LMD2
      MP=L+LMPD2
      MM=L+LMMD2
      U(LMN3)=SMP4*(U(LM)+U(LP)+U(MM)+U(MP))+SMP*U(LMN3)
      V(LMN3)=SMP4*(V(LM)+V(LP)+V(MM)+V(MP))+SMP*V(LMN3)
      P(LMN3)=SMP4*(P(LM)+P(LP)+P(MM)+P(MP))+SMP*P(LMN3)
      RO(LMN3)=SMP4*(RO(LM)+RO(LP)+RO(MM)+RO(MP))+SMP*RO(LMN3)
170   CONTINUE

      RETURN
180   FORMAT (1H ,5X,2I5,3F18.8)
190   FORMAT (1H0,63HLOCAL ARTIFICAL VISCOSITY PARAMETERS FOR SHOCK CALC,
     1ULATIONS, N=,I4,5H (PC=,F5.1,1H),/,/,10X,1HL,4X,1HM,10X,3HQUT,
     2 11X,3HQVT,11X,3HQPT,/)
200   FORMAT (1H1)
      END
```

B.13 Exit Boundary Conditions (exitt.f)

This subroutine calculates boundary mesh points at the nozzle exit. It implements the exit boundary condition calculation based on prescribed exit plane conditions (pressure, or characteristic relations for subsonic flow). *OCR note: this listing was reconstructed from Appendix C PDF text blocks for readability and line-aligned to the degraded Appendix C compatibility-equation block.*

```
      SUBROUTINE EXITT
C
C     ***************************************************************
C     THIS SUBROUTINE CALCULATES THE BOUNDARY MESH POINTS AT THE
C     NOZZLE EXIT FOR SUBSONIC FLOW
C     ***************************************************************
C
C     OCR note:
C     This file was reconstructed from Appendix C PDF text blocks.
C     Compatibility-equation updates were line-aligned to Appendix C
C     degraded text blocks and normalized to fixed-form syntax.
C
```

```
      COMMON /AV/ IAV,CAV,NST,SMP,LSS,CTA,XMU,XLA,RKMU,QUT(81,21),QVT(81,21),QPT(81,21)
      COMMON /ONESID/ UD(4),VD(4),PD(4),ROD(4)
      COMMON /SOLUTN/ U(81,21,2),V(81,21,2),P(81,21,2),RO(81,21,2)
      COMMON /CNTRLC/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,GAMMA,RGAS,GAM1,GAM2,
     1L1,L2,L3,M1,M2,DX,DY,DT,N,N1,N3,NASM,IVEL,ICHAR,N1D,LJET,JFLAG,
     2IERR,IUI,IUO,DXR,DYR,LD,MD,LMD1,LMD3,IB,RSTAR,RSTARS,NPLOT,G,PC,TC,
     3LC,PLOW,ROLOW
      COMMON /GEMTRYC/ NGEOM,XI,RI,XT,RT,XE,RE,RCI,RCT,ANGI,ANGE,XW(81),
     1YW(81),XWI(81),YWI(81),NXNY(81),NWPTS,IINT,IDIF,LT,NDIM
      COMMON /GCB/ NGCB,XICB,RICB,XTCB,RTCB,XECB,RECB,RCICB,RCTCB,ANGICB,
     1ANGECB,XCB(81),YCB(81),XCBI(81),YCBI(81),NXNYCB(81),NCBPTS,IINTCB,
     2IDIFCB,LECB
      COMMON /BCC/ PT(21),TT(21),THETA(21),PE,MASSE,MASSI,MASST,THRUST,
     1NSTAG
      REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE
C
      X3=XE
      ATERM1=0.0
      ATERM2=0.0
      ATERM3=0.0
C
      DO 160 ICHAR=1,2
      DO 160 M=1,MMAX
      CALL MAP (2,LMAX,M,AL,BE,DE,L1,AL1,BE1,DE1)
      U1=U(LMAX,M,N1)
      U2=U1
      A1=SQRT(GAMMA*P(LMAX,M,N1)/RO(LMAX,M,N1))
      A2=A1
      IF (ICHAR.EQ.2) GO TO 10
      U(LMAX,M,N3)=U1
      A3=A1
C
C     CALCULATE THE PROPERTY INTERPOLATING POLYNOMIAL COEFFICIENTS
C
10    BU=(U(LMAX,M,N1)-U(L1,M,N1))*DXR
      BV=(V(LMAX,M,N1)-V(L1,M,N1))*DXR
      BP=(P(LMAX,M,N1)-P(L1,M,N1))*DXR
      BRO=(RO(LMAX,M,N1)-RO(L1,M,N1))*DXR
      BYCB=(YCB(LMAX)-YCB(L1))*DXR
      BAL=(AL-AL1)*DXR
      BBE=(BE-BE1)*DXR
      BDE=(DE-DE1)*DXR
      CU=U(LMAX,M,N1)-BU*X3
      CV=V(LMAX,M,N1)-BV*X3
      CP=P(LMAX,M,N1)-BP*X3
      CRO=RO(LMAX,M,N1)-BRO*X3
      CYCB=YCB(LMAX)-BYCB*X3
      CAL=AL-BAL*X3
      CBE=BE-BBE*X3
      CDE=DE-BDE*X3
C
C     CALCULATE THE CROSS DERIVATIVE INTERPOLATING POLYNOMIAL
C     COEFFICIENTS
C
      IF (M.EQ.1) GO TO 20
      DU=(U(LMAX,M,N1)-U(LMAX,M-1,N1))*DYR
      DV=(V(LMAX,M,N1)-V(LMAX,M-1,N1))*DYR
      DP=(P(LMAX,M,N1)-P(LMAX,M-1,N1))*DYR
      DRO=(RO(LMAX,M,N1)-RO(LMAX,M-1,N1))*DYR
      DU1=(U(L1,M,N1)-U(L1,M-1,N1))*DYR
      DV1=(V(L1,M,N1)-V(L1,M-1,N1))*DYR
      DP1=(P(L1,M,N1)-P(L1,M-1,N1))*DYR
      DRO1=(RO(L1,M,N1)-RO(L1,M-1,N1))*DYR
      GO TO 80
20    IF (LECB.EQ.LMAX) GO TO 30
      DU=0.0
      DV=V(LMAX,2,N1)*DYR
      DP=0.0
      DRO=0.0
      DU1=0.0
      DV1=V(L1,2,N1)*DYR
      DP1=0.0
      DRO1=0.0
      GO TO 80
30    DU=(U(LMAX,2,N1)-U(LMAX,1,N1))*DYR
      DV=(V(LMAX,2,N1)-V(LMAX,1,N1))*DYR
```

```
      DP=(P(LMAX,2,N1)-P(LMAX,1,N1))*DYR
      DRO=(RO(LMAX,2,N1)-RO(LMAX,1,N1))*DYR
      DU1=(U(L1,2,N1)-U(L1,1,N1))*DYR
      DV1=(V(L1,2,N1)-V(L1,1,N1))*DYR
      DP1=(P(L1,2,N1)-P(L1,1,N1))*DYR
      DRO1=(RO(L1,2,N1)-RO(L1,1,N1))*DYR
80    BDU=(DU-DU1)*DXR
      BDV=(DV-DV1)*DXR
      BDP=(DP-DP1)*DXR
      BDRO=(DRO-DRO1)*DXR
      CDU=DU-BDU*X3
      CDV=DV-BDV*X3
      CDP=DP-BDP*X3
      CDRO=DRO-BDRO*X3
C
C     CALCULATE X1 AND X2
C
      IF (ICHAR.EQ.2) A3=SQRT(GAMMA*P(LMAX,M,N3)/RO(LMAX,M,N3))
      DO 50 IL=1,2
      X1=X3-(U(LMAX,M,N3)-U1)*0.5*DT
      X2=X3-(U(LMAX,M,N3)-A3+U2-A2)*0.5*DT
C
C     INTERPOLATE FOR THE PROPERTIES
C
      U1=BU*X1+CU
      U2=BU*X2+CU
      P2=BP*X2+CP
      RO2=BRO*X2+CRO
      A2=SQRT(GAMMA*P2/RO2)
50    CONTINUE
      V1=BV*X1+CV
      P1=BP*X1+CP
      RO1=BRO*X1+CRO
      YCB1=BYCB*X1+CYCB
      AL1=BAL*X1+CAL
      BE1=BBE*X1+CBE
      DE1=BDE*X1+CDE
      UV1=U1*AL1+V1*BE1+DE1
      A1=SQRT(GAMMA*P1/RO1)
      V2=BV*X2+CV
      YCB2=BYCB*X2+CYCB
      AL2=BAL*X2+CAL
      BE2=BBE*X2+CBE
      DE2=BDE*X2+CDE
      UV2=U2*AL2+V2*BE2+DE2
C
C     INTERPOLATE FOR THE CROSS DERIVATIVES
C
      DU1=BDU*X1+CDU
      DV1=BDV*X1+CDV
      DP1=BDP*X1+CDP
      DRO1=BDRO*X1+CDRO
      DU2=BDU*X2+CDU
      DV2=BDV*X2+CDV
      DP2=BDP*X2+CDP
      DRO2=BDRO*X2+CDRO
C
C     CALCULATE THE PSI TERMS
C
      IF (NDIM.EQ.0) GO TO 70
      IF (M.EQ.1.AND.LECB.NE.LMAX) GO TO 60
      ATERM1=RO1*V1/((DY*(M-1))/BE1+YCB1)
      ATERM2=RO2*V2/((DY*(M-1))/BE2+YCB2)
      GO TO 70
60    ATERM1=RO1*BE1*DV1
      ATERM2=RO2*BE2*DV2
70    PSI31=-UV1*DV1-BE1*DP1/RO1
      PSI41=-UV1*DP1+A1*A1*UV1*DRO1
      PSI12=-UV2*DRO2-RO2*AL2*DU2-RO2*BE2*DV2-ATERM2
      PSI22=-UV2*DU2-AL2*DP2/RO2
      PSI42=-UV2*DP2+A2*A2*UV2*DRO2
      IF (ICHAR.EQ.1) GO TO 130
C
C     CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
C
      IF (M.EQ.1.AND.LECB.NE.LMAX) GO TO 88
```

```
      IF (M.EQ.MMAX) GO TO 98
      DU3=(U(LMAX,M+1,N3)-U(LMAX,M,N3))*DYR
      DV3=(V(LMAX,M+1,N3)-V(LMAX,M,N3))*DYR
      DP3=(P(LMAX,M+1,N3)-P(LMAX,M,N3))*DYR
      DRO3=(RO(LMAX,M+1,N3)-RO(LMAX,M,N3))*DYR
      GO TO 100
88    DU3=0.0
      DV3=V(LMAX,2,N3)*DYR
      DP3=0.0
      DRO3=0.0
      GO TO 100
98    DU3=(U(LMAX,MMAX,N3)-U(LMAX,M1,N3))*DYR
      DV3=(V(LMAX,MMAX,N3)-V(LMAX,M1,N3))*DYR
      DP3=(P(LMAX,MMAX,N3)-P(LMAX,M1,N3))*DYR
      DRO3=(RO(LMAX,MMAX,N3)-RO(LMAX,M1,N3))*DYR
C
C     CALCULATE THE PSI TERMS AT THE SOLUTION POINT
C
100   IF (NDIM.EQ.0) GO TO 120
      IF (M.EQ.1.AND.LECB.NE.LMAX) GO TO 110
      ATERM3=RO(LMAX,M,N3)*V(LMAX,M,N3)/((DY*(M-1))/BE+YCB(LMAX))
      GO TO 120
110   ATERM3=RO(LMAX,1,N3)*BE*DV3
120   UV3=U(LMAX,M,N3)*AL+V(LMAX,M,N3)*BE+DE
      PSI13=-UV3*DRO3-RO(LMAX,M,N3)*(AL*DU3+BE*DV3)-ATERM3
      PSI23=-UV3*DU3-AL*DP3/RO(LMAX,M,N3)
      PSI33=-UV3*DV3-BE*DP3/RO(LMAX,M,N3)
      PSI43=-UV3*DP3+A3*A3*UV3*DRO3
C
C     CALCULATE THE COMPATIBILITY EQUATION COEFFICIENTS
C
130   IF (ICHAR.EQ.1) GO TO 140
      PSI31B=(PSI31+PSI33)*0.5
      PSI41B=(PSI41+PSI43)*0.5
      PSI12B=(PSI12+PSI13)*0.5
      PSI22B=(PSI22+PSI23)*0.5
      PSI42B=(PSI42+PSI43)*0.5
      GO TO 150
140   PSI31B=PSI31
      PSI41B=PSI41
      PSI12B=PSI12
      PSI22B=PSI22
      PSI42B=PSI42
C
C     SOLVE THE COMPATIBILITY EQUATIONS FOR U, V, P, AND RO
C
150   P(LMAX,M,N3)=PE
      RO(LMAX,M,N3)=RO1+(P(LMAX,M,N3)-P1-DT*PSI41B)/(A3*A3+A1*A1)
      U(LMAX,M,N3)=U2+(((PSI42B+(RO2-RO(LMAX,M,N3))*(A2+A3)*PSI22B/4.0
     1 -(A2*A2+A3*A3)*PSI12B/2.0)*DT-(P(LMAX,M,N3)-P2))
     2 /(RO2+RO(LMAX,M,N3))/(A2+A3))*4.0
      V(LMAX,M,N3)=V1+DT*PSI31B
160   CONTINUE
C
      V(LMAX,MMAX,N3)=-U(LMAX,MMAX,N3)*NXNY(LMAX)
      V(LMAX,1,N3)=-U(LMAX,1,N3)*NXNYCB(LMAX)
      IF (JFLAG.EQ.0) RETURN
      V(LMAX,MMAX,N3)=V(LMAX,MMAX,N3)+XWI(LMAX)
      RETURN
      END
```

B.14 Visualization Output (plot.f)

This subroutine generates visualization output including velocity vector plots and contour plots of dependent variables (pressure, density, entropy). It creates plotter data for external visualization tools. *OCR note: this listing was reconstructed from Appendix C PDF text blocks for readability; contour-cell intersection logic was restored from degraded scan lines with normalized fixed-form token syntax.*

```
      SUBROUTINE PLOT(TITLE,T,NP)
C
C     ************************************************************
C     THIS SUBROUTINE PLOTS THE VELOCITY VECTORS AND DEPENDENT
C     VARIABLE CONTOUR PLOTS
C     ************************************************************
C
C     OCR note:
C     Reconstructed from Appendix C PDF text blocks. Contour-cell
C     intersection logic (labels 240-320) was restored from degraded
C     scan blocks with normalized fixed-form token syntax.
C
      DIMENSION CQ(81,21),CON(9),XCO(4),YCO(4),TITLE(8)
      COMMON /AV/ IAV,CAV,NST,SMP,LSS,CTA,XMU,XLA,RKMU,QUT(81,21),QVT(81,21),QPT(81,21)
      COMMON /ONESID/ UD(4),VD(4),PD(4),ROD(4)
      COMMON /SOLUTN/ U(81,21,2),V(81,21,2),P(81,21,2),RO(81,21,2)
      COMMON /CNTRLC/ LMAX,MMAX,NMAX,NPRINT,TCONV,FDT,GAMMA,RGAS,GAM1,GAM2,
     1L1,L2,L3,M1,M2,DX,DY,DT,N,N1,N3,NASM,IVEL,ICHAR,N1D,LJET,JFLAG,
     2IERR,IUI,IUO,DXR,DYR,LD,MD,LMD1,LMD3,IB,RSTAR,RSTARS,NPLOT,G,PC,TC,
     3LC,PLOW,ROLOW
      COMMON /GEMTRYC/ NGEOM,XI,RI,XT,RT,XE,RE,RCI,RCT,ANGI,ANGE,XW(81),
     1YW(81),XWI(81),YWI(81),NXNY(81),NWPTS,IINT,IDIF,LT,NDIM
      COMMON /GCB/ NGCB,XICB,RICB,XTCB,RTCB,XECB,RECB,RCICB,RCTCB,ANGICB,
     1ANGECB,XCB(81),YCB(81),XCBI(81),YCBI(81),NXNYCB(81),NCBPTS,IINTCB,
     2IDIFCB,LECB
      COMMON /BCC/ PT(21),TT(21),THETA(21),PE,MASSE,MASSI,MASST,THRUST,
     1NSTAG
      REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE
C
C     GENERATE THE VELOCITY VECTOR PLOT
C
      ND=N3
      IF (N.EQ.0) ND=1
      CALL GETG (4HKJBN,JNM)
      XL=XI
      XR=XE
      YT=YW(1)
      YB=YCB(1)
      DO 10 L=2,LMAX
      YT=AMAX1(YT,YW(L))
      YB=AMIN1(YB,YCB(L))
 10   CONTINUE

      VV=0.9*DX
      FIYB=916.0
      XD=(XR-XL)/(YT-YB)
      FIR=(1022.0-1022.0/FLOAT(L1)-1.0)/900.0
      IF (XD.LE.FIR) GO TO 20
      FIXL=0.0
      FIXR=1022.0-1022.0/FLOAT(L1)-1.0
      FIYT=916.0-FIXR/XD
      GO TO 30
 20   FIXL=511.0-450.0*XD
      FIXR=511.0+450.0*XD
      FIYT=16.0
 30   XCONV=(FIXR-FIXL)/(XR-XL)
      YCONV=(FIYT-FIYB)/(YT-YB)

      VMAX=0.0
      DO 40 L=1,LMAX
      DO 40 M=1,MMAX
      VMAX=AMAX1(VMAX,ABS(U(L,M,ND)),ABS(V(L,M,ND)))
 40   CONTINUE
      IF (VMAX.LT.1.0E-10) GO TO 60

      DROU=VV/VMAX
      CALL ADV (1)
      DO 50 L=1,LMAX
      IX1=FIXL+(FLOAT(L-1)*DX)*XCONV
      DYL=(YW(L)-YCB(L))/FLOAT(MMAX-1)
      DO 50 M=1,MMAX
      IY1=FIYB+(YCB(L)+FLOAT(M-1)*DYL-YB)*YCONV
      IX2=FIXL+(FLOAT(L-1)*DX+U(L,M,ND)*DROU)*XCONV
      IY2=FIYB+(YCB(L)+FLOAT(M-1)*DYL-YB+V(L,M,ND)*DROU)*YCONV
      CALL DRV (IX1,IY1,IX2,IY2)
```

```
      CALL PLT (IX1,IY1,16)
50    CONTINUE
      CALL LINCNT (59)
      WRITE (7,410)
      WRITE (7,550) JNM,TITLE,NP,T
C
C     GENERATE THE CONTOUR PLOTS
C
60     I=0
70     I=I+1
      GO TO (80,100,120,140,340), I

80    DO 90 L=1,LMAX
      DO 90 M=1,MMAX
      CQ(L,M)=RO(L,M,ND)*G
90    CONTINUE
      GO TO 160

100   DO 110 L=1,LMAX
      DO 110 M=1,MMAX
      CQ(L,M)=P(L,M,ND)/PC
110   CONTINUE
      GO TO 160

120   DO 130 L=1,LMAX
      DO 130 M=1,MMAX
      CQ(L,M)=P(L,M,ND)/RO(L,M,ND)/RGAS/G-TC
130   CONTINUE
      GO TO 160

140   DO 150 L=1,LMAX
      DO 150 M=1,MMAX
      CQ(L,M)=SQRT(U(L,M,ND)**2+V(L,M,ND)**2)/(GAMMA*P(L,M,ND)/RO(L,M,ND))
150   CONTINUE

160   QMN=1.0E06
      QMX=-QMN
      DO 170 L=1,LMAX
      DO 170 M=1,MMAX
      QMN=AMIN1(CQ(L,M),QMN)
      QMX=AMAX1(CQ(L,M),QMX)
170   CONTINUE
      XX=QMX-QMN
      DQ=0.1*XX
      DO 180 K=1,9
      CON(K)=QMN+FLOAT(K)*DQ
180   CONTINUE
      K=9

      CALL ADV (1)
      CALL LINCNT (59)
      GO TO (190,200,210,220), I
190   WRITE (7,560)
      GO TO 230
200   WRITE (7,570)
      GO TO 230
210   WRITE (7,580)
      GO TO 230
220   WRITE (7,590)
230   WRITE (7,400) QMN,QMX,CON(1),CON(K),DQ
      WRITE (7,550) JNM,TITLE,NP,T
C
      DO 520 L=2,LMAX
      DY=(YW(L-1)-YCB(L-1))/FLOAT(MMAX-1)
      DY1=(YW(L)-YCB(L))/FLOAT(MMAX-1)
      DO 520 M=2,MMAX
      NN=0
      DO 520 KK=1,K
       K1=0
       K2=0
       K3=0
       K4=0
       IF (CQ(L-1,M-1).LE.CON(KK)) K1=1
       IF (CQ(L,M-1).LE.CON(KK)) K2=1
       IF (CQ(L-1,M).LE.CON(KK)) K3=1
       IF (CQ(L,M).LE.CON(KK)) K4=1
```

```
      IF (K1*K2*K3*K4.NE.0) GO TO 520
      IF (K1+K2+K3+K4.EQ.0) GO TO 520
      IF (NN.NE.0) GO TO 240
      NN=1
      XCO(1)=XI+FLOAT(L-2)*DX
      XCO(2)=XCO(1)+DX
      XCO(3)=XCO(1)
      XCO(4)=XCO(2)
      YCO(1)=YCB(L-1)+FLOAT(M-2)*DY
      YCO(2)=YCB(L)+FLOAT(M-2)*DY1
      YCO(3)=YCB(L-1)+FLOAT(M-1)*DY
      YCO(4)=YCB(L)+FLOAT(M-1)*DY1
240   LL=0
      IF (K1+K3.NE.1) GO TO 250
      IC1=1
      IC2=3
      LP1=L-1
      MP1=M-1
      LP2=L-1
      MP2=M
      ASSIGN 250 TO KR1
      GO TO 280
250   IF (K1+K2.NE.1) GO TO 260
      IC1=1
      IC2=2
      LP1=L-1
      MP1=M-1
      LP2=L
      MP2=M-1
      ASSIGN 260 TO KR1
      GO TO 280
260   IF (K2+K4.NE.1) GO TO 270
      IC1=2
      IC2=4
      LP1=L
      MP1=M-1
      LP2=L
      MP2=M
      ASSIGN 270 TO KR1
      GO TO 280
270   IF (K3+K4.NE.1) GO TO 520
      IC1=3
      IC2=4
      LP1=L-1
      MP1=M
      LP2=L
      MP2=M
      ASSIGN 520 TO KR1
280   LL=LL+1
      XX=(CON(KK)-CQ(LP1,MP1))/(CQ(LP2,MP2)-CQ(LP1,MP1))
      IF (LL.EQ.2) GO TO 290
      IX1=FIXL+(XCO(IC1)+XX*(XCO(IC2)-XCO(IC1))-XL)*XCONV
      IY1=FIYB+(YCO(IC1)+XX*(YCO(IC2)-YCO(IC1))-YB)*YCONV
      GO TO KR1,(250,260,270,320)
290   IX2=FIXL+(XCO(IC1)+XX*(XCO(IC2)-XCO(IC1))-XL)*XCONV
      IY2=FIYB+(YCO(IC1)+XX*(YCO(IC2)-YCO(IC1))-YB)*YCONV
      CALL DRV (IX1,IY1,IX2,IY2)
      IF (KK.NE.1) GO TO 300
      CALL PLT (IX1,IY1,35)
300   IF (KK.NE.K) GO TO 310
      CALL PLT (IX1,IY1,24)
310   LL=0
      IF (LP2.NE.L) GO TO 520
      IF (MP2.NE.M) GO TO 320
      GO TO 260
320   CONTINUE
      CONTINUE
520   CONTINUE

      DO 330 L=2,LMAX
      IX1=FIXL+(FLOAT(L-2)*DX)*XCONV
      IX2=FIXL+(FLOAT(L-1)*DX)*XCONV
      IY1=FIYB+(YCB(L-1)-YB)*YCONV
      IY2=FIYB+(YCB(L)-YB)*YCONV
      IY3=FIYB+(YW(L-1)-YB)*YCONV
      IY4=FIYB+(YW(L)-YB)*YCONV
```

```
      CALL DRV (IX1,IY1,IX2,IY2)
      CALL DRV (IX1,IY3,IX2,IY4)
330   CONTINUE
      GO TO 70

340   DY=1.0/FLOAT(MMAX-1)
      CALL ADV (1)
      RETURN

400   FORMAT (1H ,10HLOW VALUE=,1PE11.4,2X,11HHIGH VALUE=,E11.4,2X,
     1 12HLOW CONTOUR=,E11.4,2X,13HHIGH CONTOUR=,E11.4,2X,
     2 14HDELTA CONTOUR=,E11.4)
410   FORMAT (1H ,16HVELOCITY VECTORS)
550   FORMAT (1H ,A10,4X,8A10,2X,2HN=,I4,2X,2HT=,1PE10.4,4H SEC)
560   FORMAT (1H ,7HDENSITY)
570   FORMAT (1H ,8HPRESSURE)
580   FORMAT (1H ,11HTEMPERATURE)
590   FORMAT (1H ,11HMACH NUMBER)
      END
```

Appendix A

Characteristic Relations: η Constant Plane

Introduction

This appendix derives the characteristic relations for the η = constant reference plane. These relations are used to implement inlet boundary conditions in the NAP solver.

I. Equations of Motion

The equations of motion can be written in the form:

$$\frac{\partial P}{\partial t} + u\frac{\partial P}{\partial \xi} + v\frac{\partial P}{\partial \eta} = -vP_\eta - pau_\xi - pBv_\xi - \frac{epvB}{n} \tag{A.1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial \xi} + \frac{P_\xi}{p} = -vu_\eta - ap_\xi/p \tag{A.2}$$

$$\frac{\partial v}{\partial t} + uv_\xi + \frac{BP_\eta}{p} = -vv_\eta - \frac{eP_\eta}{p} \tag{A.3}$$

$$\frac{\partial P}{\partial t} + up_\xi - a^2\left(\frac{\partial P}{\partial t} + u\frac{\partial P}{\partial \xi}\right) = -vP_\eta + a^2vp_\eta \tag{A.4}$$

II. Characteristic Curves

The characteristic curves are derived from analysis of the system's hyperbolicity:

$$\frac{d\eta}{dT} = v \tag{A.5}$$

$$\frac{d\eta}{dx} = \frac{\partial \eta}{\partial x} \tag{A.6}$$

III. Characteristic Variables

Define the characteristic variables:

$$\psi_1 = vP_\xi - pau_\xi - pBv_\xi - \frac{epvB}{n} \tag{A.7}$$

$$\psi_2 = -vu_\xi - aP_\xi/p \tag{A.8}$$

$$\psi_3 = -vv_\xi - \frac{eP_\xi}{p} \tag{A.9}$$

$$\psi_4 = -vP_\xi + a^2vP_\xi \tag{A.10}$$

IV. Compatibility Equations

Substituting the characteristic equations yields compatibility relations. For the characteristic curve with slope $d\eta = (u - a)dT$:

$$dp - \rho adu = (\psi_2 + 2\psi_1 - \rho a\psi_2)dT \tag{A.11}$$

For the characteristic curve with slope $d\eta = (u + a)dT$:

$$dp + \rho adu = (\psi_4 + 2\psi_1 + \rho a\psi_2)dT \tag{A.12}$$

These relations provide the basis for implementing inlet boundary conditions through the method of characteristics.

Note: Technical content in this appendix was extracted via OCR from the original NAP document. Equation symbols, indices, and coordinate transformations have been verified against the method description in Chapter I but should be confirmed against the original source for critical applications.

Appendix B

Characteristic Relations: ζ Constant Plane

Introduction

This appendix derives the characteristic relations for the ζ = constant reference plane. These relations are used to implement wall and centerbody boundary conditions in the NAP solver.

I. Equations of Motion

The equations of motion for the ζ-plane are:

$$\frac{\partial P}{\partial t} + v\frac{\partial P}{\partial \eta} + \rho a u_\eta + \rho B v_\eta = -uP_\xi - \rho u_\xi - \frac{epvB}{n} \tag{B.1}$$

$$\frac{\partial u}{\partial t} + vu_\eta + \frac{aP_\eta}{\rho} = -uu_\xi - \frac{P_\xi}{\rho} \tag{B.2}$$

$$\frac{\partial v}{\partial t} + vv_\eta + \frac{BP_\eta}{\rho} = -uv_\xi \tag{B.3}$$

$$\frac{\partial P}{\partial t} + vP_\eta - a^2\left(\frac{\partial P}{\partial t} + v\frac{\partial P}{\partial \eta}\right) = -uP_\xi + a^2 uP_\xi \tag{B.4}$$

II. Characteristic Curves

Following the development of Appendix A, the characteristic curves for the ζ-plane are:

$$\frac{d\zeta}{dT} = v \tag{B.5}$$

$$\frac{d\zeta}{dx} = v \pm a^* a \tag{B.6}$$

where $a^* = (a^2 + B^2)^{1/2}$ represents the effective sound speed in the transformed coordinate system.

III. Compatibility Equations

The compatibility equations for the ζ-constant plane are:

$$a du - \rho dv = (\psi_0 - a\psi_1) dT \tag{B.7}$$

$$dp - a^2 d\rho = \psi_4 d\xi \tag{B.8}$$

$$dp - \rho a^2 \frac{du}{a^*} - \rho B a \frac{dv}{a^*} = \left(\psi_2 + a\psi_1 - \frac{\rho a a \psi_0}{a^*} - \frac{\rho B a \psi_1}{a^*} \right) dT \tag{B.9}$$

$$dp + \rho a^2 \frac{du}{a^*} + \rho B a \frac{dv}{a^*} = \left(\psi_3 + a\psi_1 + \frac{\rho a a \psi_0}{a^*} + \frac{\rho B a \psi_1}{a^*} \right) dT \tag{B.10}$$

These compatibility equations apply along the characteristic curves and provide the boundary condition implementation for wall and centerbody surfaces.

Note: Technical content in this appendix was extracted via OCR from the original NAP document and reconstructed using the methods described in Chapter I.E. Coordinate transformations and all equations should be verified against reference material before use in alternative implementations.