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:

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

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

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 4: Mach Number Contours and Plug Pressure
10° Conical Plug Nozzle
[Placeholder - Original image required]

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.

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.

```
      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(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
     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,NS
    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 (NOIM,EO,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,EO,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=LC=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,EG,0) GO TO 250
      DO 240 M=1,MMAX
      U(1,M,1)=UI(M)
      V(1,M,I)=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=YCBL(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,XHACH,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)
      PO(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(U(L1,M,N3)-RO(L2,M,N3))
      IF (P(LMAX,M,N1),GT,0.0,AND,RO(LMAX,M,N3),GT,0.0) GO TO 420'MrNl^CT.B.B) 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,1,3)=-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 43
      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=NCP+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 (JFIAG,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 MA8FLO (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,H,N3)**2+V(L,MN3)**2)
      XMACH=VELMAG/SQRT(GAMMA*P(L,M,N3)/RO(L,M,N3))
      PRES=P(L,M,N3)/PC
      RHO=RO(L,H,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=RRHO*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 85, 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*0.4535
      MASSI=MASSI*0.4535
      MASSE=MASSE*0.4535
      THRUST=THRUS*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
      TIM=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,95HTO 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,21HBOUNOARY 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 NO.,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=,14)
        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)

```
     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
      BU=(P(L1MN1)-P(LMN1))*DXR
      BRO=(RO(L1MN1)-(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)-B*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-BD*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 (CHAR,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)-(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)

```
        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 $ NDUM1=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=WXI(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,BEW,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) TO 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 (ILJET.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 (ICAR.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,BEW,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)) ATERMS=ATDS
     GO TO 240
230  ATERMS=RO3*V(L,2,N3)*DYR*BE
C
240  PSI13=-U3*DRO3-RO3*DU3-ATERM3
     PSI23=-U3*DU3-DP3/RP3
     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 (ICAR.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 QUARE 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)

```
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(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,
    1NSTAG
     REAL MN3,NXNY,MASSI,MASST,NXNYCB,MASSE
```

```
      MN3=0.01
      IF (NID,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
```

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:

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

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

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

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

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:

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

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

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

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

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.